Mobile Cinema

by
Pengkai Pan

B.E. Industrial Design, Zhejiang University, 1994
M.S. Media Science in Media Technology
Massachusetts Institute of Technology
Cambridge, MA, 1999

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

at the
Massachusetts Institute of Technology
July 2004

Signature of Author
____

Program in Media Arts and Sciences
July 2004

Certified By
____

Glorianna Davenport
Principal Research Associate
Director of Interactive Cinema
Program in Media Arts and Sciences
Thesis Supervisor

Accepted By
____

Andrew Lippman
Chairman
Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Mobile Cinema

by
Pengkai Pan

MIT Program in Media Arts and Sciences
July 2004

Abstract

This thesis develops techniques and methods that extend the art and craft of storytelling, and in particular enable the creation of mobile cinema.

Stories are always constrained by the medium in which they are told and the mode by which they are delivered to an audience. This dissertation addresses the design of content, systems, and tools that facilitate the emerging type of computational audio-visual narrative that we call mobile cinema. Storytelling in this medium requires temporally and spatially encoded narrative segments that are delivered over a wireless channel to mobile devices such as PDAs and mobile phones. These devices belong to "the audience," individuals who are navigating physical space and interact with local circumstances in the environment.

This thesis examines the underlying requirements for coherent mobile narrative and explores two particular challenges which must be solved in order to make a reliable and scalable stream of content for mobile cinema: technology uncertainty (the fact that what the mobile cinema system presents may not be what the creator intends) and participation uncertainty (the fact that what the audience does may not be what the creator expects).

The exploration and analysis of these problems involved prototyping two versions of the M-Views system for mobile cinema and three prototype cinematic narratives. Small user studies accompanied each production. The iterative process enabled the author to explore both aspects of uncertainty and to introduce innovations in four key areas to help address these uncertainties: practical location detection, authoring tools designed for mobile channels, responsive story presentation mechanisms, and creative story production strategies.

Thesis supervisor: Glorianna Davenport

Mobile Cinema

by
Pengkai Pan

The following people served as readers for this thesis:

Thesis reader: ____

William J. Mitchell
Academic Head
Program in Media Arts and Sciences

Thesis reader: ____

Pattie Maes
Associate Professor
Program in Media Arts and Sciences

Thesis reader: ____

Ted Selker
Associate Professor
Program in Media Arts and Sciences

Acknowledgments

First and foremost I would like to thank my supervisor, Glorianna Davenport, who since 1997 has been giving me a fabulous research environment, providing various resources, and supporting me to develop novel ideas. I would also like to thank my remarkable committee members, William J. Mitchell, Pattie Maes, and Ted Selker for all of their encouragement, help, and understanding as this work has progressed over the past several years. The support and assistance of the following MIT faculty and academic research staff are gratefully acknowledged: Walter Bender, Dan Ariely, Henry Lieberman, Bruce Blumberg, V. Michael Bove, Jr., Chris Csikszentmihályi, Joe Jacobson, Andrew Lippman, Joe Paradiso, Mitchel Resnick, John Maeda, Hiroshi Ishii, Nicholas Negroponte, Henry Jenkins III, Janet Sonenberg, Alan Brody, Robert Irwin, and Tom Malone.

I would like to acknowledge the contributions of all M-Views project team members, David Crow, Carly Kastner, Lilly Kam, Debora Lui, Chris Toepel, Meng Mao, William T. Stoltzman, and Dan Bersak. Without their contributions to both system development and story productions, this research journey could not have been finished.

Special thanks to our Interactive Cinema colleagues and alumni: Barbara Barry, Paul Nemirovsky, Aisling Kelliher, Ali Mazalek, Kevin Michael Brooks, James Jung-Hoon Seo, Roger (Arnan) Sipitakiat, Flavia Sparacino, and A. Arjan Schutte. They have been invaluable sources of information and advice on everything narrative-related.

Beyond the Interactive Cinema Group, I would especially like to thank friends and colleagues at Media Lab: Erik Blankenship, Cameron Marlow, Sunil Vemuri, Karrie Karahalios, Fernanda Viegas, Kimiko Ryokai, Brygg Ullmer, Angela Chang, Fulu Li, Gili Weinberg, James Dai, Hugo Liu, Michael Rosenblatt, Josh Lifton, Ryan Chin, Nathan Eagle, Tanzeem Choudhury, Brian Clarkson, Karen Liu, Yuan Qi, Peter Gorniak, Natalia Marmasse, Stefan Marti, Ernesto Arroyo, Stefan Agamanolis, Surj Patel, Win Burleson, Andrea Lockerd, Nyssim Lefford, Wei Chai, Push Singh, Vadim Gerasimov, Xu Sun, Deborah J. Cohen, Alexandra Kahn, Deborah Widener, Polly Guggenheim, Jack Driscoll, and Tracy Daniels.

I gratefully acknowledge David Boor from IBM, Thomas Gardos from Intel, Taka Sueyoshi from Sony, Steve Whittaker from BT, Riku Suomela and Franklin Reynolds from Nokia, Wen Jean Hsue from ITRI, and Seung Hun Jeon from Samsung for their kindness and support. This work is supported in part by grants from the Media Lab's Digital Life Consortia and the Council for the Arts at MIT.

My greatest debt of gratitude is owed to my family. My parents have sacrificed so much to give me tremendous opportunities for realizing my dreams. They are the most important source of inspiration in my life. In the past several years, my wife has always been accompanying me with remarkable patience and love, even when I had very difficult times during this long research journey. I also want to thank my sister and brother-in-law. Eight years ago, my brave sister did a great interview on my behalf at the MIT Media Lab. Without her, I would not have had this once-in-a-lifetime opportunity to come to MIT. This dissertation is dedicated to my family.

Contents

List of Figures

List of Tables

List of Charts

Equation

1.0 INTRODUCTION

In the past five years, I have been developing an original type of computational visual narrative, *mobile cinema*. It is experienced in temporally and spatially encoded narrative segments that can be delivered on wireless devices, such as PDAs and mobile phones, as the audience navigates physical locations and interacts with the environment. The mobile cinema story creator pre-arranges story webs, where the participant experiences stories as an investigator or a spectator. As the participant navigates physical space, s/he triggers distinct media elements; these cinematic story elements can be considered as embedded in the locale, often depicting events that were recorded at the location where they appear.

In mobile cinema, the story journey is defined as discontinuous, participatory, and cinematic experience. In the case of this genre of media, being discontinuous means that the audience perceives the story in chunks at different locations and at different times. Being participatory means that the audience influences how the story is unfolded by walking through physical spaces, by trading story messages with other audience members, or by creating additional narrative elements. Being cinematic means that these narratives are rendered in motion pictures and sound. The narrative itself includes characters, settings, and actions, which may be built into plots. Mobile cinema can communicate a narrative that includes characters, situations, and directions to a spectator who is moving through a physical environment in which the story scenes are “virtually” embedded. The participants may not see all of the experience, but they should still experience what they think is a coherent narrative.

Mobile cinema is a new genre of media and is fundamentally different from conventional storytelling media, such as film, TV, radio, CD-ROM, and the Internet. The differences lie in three areas:

- Unlike traditional cinema that doesn’t connect viewing place to narrative setting, mobile cinema should be evocative of place in ways such that the environment augments story experience. As we have known it in the past, the cinematic story is presented more or less continuously to a viewer confined in a theater or a living room. As cinema migrated to interactive narratives experienced on CD-ROM at a desktop, the viewer/participant remained intensely focused on the screen-based experience, ignoring the surrounding place. In each of these venues, the dramatic continuum is experienced with only small real life interruptions. This focus does not occur with the same narrowness in mobile cinema, where the narrative presentation is aggressively combined with ambient environments – physical buildings, objects, people, or even smells in these physical places can affect the reading of the cinematic elements. This mixed reality story presentation is able to offer a sense of authenticity that comes from ambient awareness, the movement of the audience's body, and depicted story events that were often recorded at the location where they appear. In designing mobile cinema, we are looking to design a cinematic story experience that plays out in small fragments over time in response to how the audience walks around a location. The cinematic experience should take advantage of this half real and half virtual (cinematic) experience, allowing different members of the audience to move through the space along different paths and engage in different aspects of the physical environment (meet a friend or take a phone call, etc.).

- Mobile cinema enables the audience to layer the dramatic story experience on their memory and understanding of the physical world. After participating in a mobile cinema story, the audience has not only impressions of story elements, such as characters, settings, and plots, but also additional impressions of existing physical environments, objects, or co-players in the same story locations. For example, if the audience participates in a horror mobile cinema at a hospital, she or he would associate this shocking experience with the hospital. In contrast, conventional storytelling media present separated story worlds, which could be perceived in any theater or living room. In addition, if a mobile cinema story is being experienced with more than one participant at a time, the overall story experience may also be associated with the social encounters. Co-participating in the same mobile cinema story requires communication and collaboration and these activities would be parts of overall story impressions on memory.
- Mobile cinema communicates a narrative that includes characters, plots, situations, and directions to a spectator who is moving through a physical environment in which the story scenes are “virtually” embedded. The participants may not see all of the experience, but they should still perceive what they think is a coherent narrative. Unlike most location-based tour stories in museums, which entreat the audience to follow a curated path through an exhibition, the mobile cinema experience might have very few constraints on movement through physical space (maximum control over navigation is left to the audience). Mobile cinema stories may have chronological, spatial, or causal structures, but must progress through an arrangement of scenes that include a beginning and an end to story. In mobile cinema, event progression is critical to deriving meaning and experience with many possible paths and endings. In contrast, most museum-based tour systems are not plot-driven, but rather information-driven narratives.

Because mobile cinema is so different from conventional storytelling media, new techniques and methods must be developed and qualified for it.

1.1 Research Journey

The vision of developing mobile cinema is rooted in various forms of location-based stories[1] or performances[2] [3]. For instance, Janet Cardiff has demonstrated a location-based visual story in San Francisco’s Museum of Modern Art, where the audience picks up a video camera and actively follows a cinematic scenario as it was produced throughout the same space.[4] Her creative art works confirmed that the explicit and continuous spatial connection between the real and the virtual is able to effectively engage the audience. However, these innovative art projects have not resulted in building computational narrative platforms to support classes of multithreaded narrative through mobile channels.

The vision of building mobile cinema is also rooted in context-aware computing, inspired by a visionary paper by Mark Weiser (1991). In the past decade, many computer scientists have

[1] http://www.abbeymedia.com/Janweb/
[2] *Tamara*, created by John Krizanc and Richard Rose, is one of the best-known interactive dramas.
[3] http://www.tonylovestina.com/
[4] http://www.wired.com/news/culture/0,1284,42152,00.html?tw=wn_story_related

proposed, built, and evaluated numerous context-aware applications for museum and city tourists (Abowd *et al.* 1997; Petrelli *et al.* 1999; Reinhard *et al.* 1999; and Cheverst *et al.* 2000). However, these location-aware systems primarily deal with self-contained information. This means that the physical context, such as location, is the main triggering force, no matter what the audience has seen previously in those applications. In these systems, there is no need for story creators to consider the various order in which segments can be experienced by the audience because there is no "plot" per se. Each segment is a self contained story. In this sense the guide book approach does not require the creator to consider more sophisticated dependancies between segments which are fundamental to the presentation of story based narrative and which are more difficult to orchestrate in the mobile environment where the audience can move about at will.

My research focuses on building structured narratives via mobile channels. *Structured narratives* means that discontinuous story pieces that are embedded in space and time have story plots. Both the physical context of the participant and what the audience has previously seen are main triggering forces in presenting stories. For example, a location-based murder mystery story needs to be displayed in certain sequences in order to keep the story coherent. The story structure may involve (1) introduction of setting and characters; (2) explanation of the state of affairs; (3) initiating event; (4) emotional response; (5) complicating emotions; (6) outcome; (7) reactions to outcome (Chatman 1978; Brooks 1996).

To make structured content, story makers need to consider many possibilities of how the audience can receive mobile stories, and the mobile cinema computational system must be designed to (1) keep the coherence of story structures, and (2) cope with indeterminate input, such as walking around a physical location, from the story participant. Coherence means that mobile story plots need to be orderly, logical, and aesthetically consistent. The story plots need to be communicated as part of every viewer's narrative experience; plots tend to have a temporal development; and plots are revealed in a succession of scenes being narrated either by direct action in a scene—the audience "sees" it happen—or by a character "telling" the audience what happened off screen.

Although human beings' minds are very good at deciphering discontinuous pieces of information, given some hints or clues together in space and time, the story designer must have some levels of control to present stories with maximum impact. Story coherence as envisioned by a designer/author can be disrupted by many factors. In mobile cinema, where the author may "direct" the audience, the author needs to have a realistic understanding of what the audience in aggregate knows. For instance, if the author directs the participant by having a character name a location to which the author expects the participant to go and the participant is unfamiliar with the location named, the pathway the audience chooses may be different from what the story previously envisaged. How mobile cinema systems can be made to adjust story presentation on the fly to maintain coherence is a serious challenge.

To address this challenge, two sets of developments have been undertaken. The first set is technological development, which includes the investigations of location detection technology, authoring software, story presentation engine, story script scheme, and system architecture. The results of these developments are two versions of a computational system, M-Views, for supporting both mobile cinema production and presentation. The second set of developments involves story creation, which includes three mobile cinema story productions, two mobile cinema workshops, and several collaborative mobile productions with students from other departments. The three mobile productions are: *Another Alice, MIT in Pocket,* and *15 Minutes*;

and both mobile cinema workshops were held during MIT's Independent Activities Periods. The reason for taking the parallel development approach to exploring new forms of storytelling in the mobile domain is very simple: solving a chicken-and-egg problem. This means that without producing mobile cinema stories, I cannot know what kinds of technological problems need to be solved and what kinds of tools need to be created; without building appropriate computational tools, it is impossible to create mobile cinema stories.

The M-Views system 0.1 is the first prototype, which is based on a Compaq iPAQ equipped with a Pocket PC GPS receiver, connecting to a laptop-based server through standard 802.11 wireless networks. The client runs GPS-based location detection software and a media player; the server runs the story script, which determines the story presentation coherently, given the walking paths by the audience. The M-Views system 0.2 is the main technological platform for authoring, presenting, and evaluating mobile cinema stories in this thesis. This version mainly consists of three modules: M-Studio, M-Views Server, and M-Views Presenter. M-Studio is a desktop-based authoring tool for supporting the design of episodic cinematic stories at distributed locations; the server is in charge of detecting contextual information, delivering messages, parsing mobile cinema story scripts, handling video streaming, and administering users' profiles and log files; and the client runs MapAgent and M-Views Messenger. The basic functions of MapAgent are (1) to detect the audience's physical location by triangulating 802.11 wireless signals and (2) to offer the author an easy-to-use tool for defining and managing story spots. The comparison between the M-Views system 0.1 and 0.2 is shown in Table 1.

	M-Views System 0.1	**M-Views System 0.2**
Location-detection	GPS	802.11 triangulation
Authoring tool	N/A	M-Studio
Client	A location monitor software PocketTV Media Player	MapAgent M-Views messaging framework PocketTV Media Player
Server	Location data comparison component Media server Apache Tomcat	MapAgent Messaging framework Flagging system XML database Apache Tomcat Web administration

Table 1 Comparison between the M-Views system 0.1 and 0.2

On the story production side, *Another Alice* is the first story production, created on the M-Views system 0.1. This is a murder-mystery story in which the viewer is the investigator, who must literally go to the location where the next clip takes place in order to trigger playback. Since each character is telling the story from his/her perspective, each narration is different. *MIT in Pocket* uses the M-Views system 0.2 technology and features over fifty story events that take place across the MIT campus. The contextual cues include location and time of day (e.g., if someone is in the main lobby at 10 a.m., they will be sent a clip of a student rushing to their morning class in the nearby lecture hall). This story web is based on both fictional and non-fictional components. Unlike *Another Alice, MIT in Pocket* employs a web-structure, which allows the participant to start a story from any story location at MIT and which may end at any story location. *15 Minutes* is another murder-mystery-like story, in which all scenes happen in only three places in a corporation building. To participate in this story, the audience has to investigate wild rumors about a robbery threat by walking or running among the three locations. Unlike *Another Alice* and *MIT in Pocket, 15 Minutes* has much more participation tolerance. This means that the participant can start the story from any of three locations and the story structure is always coherent, no matter how she walks through this story. The comparisons among *Another Alice, MIT in Pocket,* and *15 Minutes* are shown in Table 2.

	Another Alice	***MIT in Pocket***	***15 Minutes***
Goal	Create a fictional mobile cinema story	Reflect the spirit of MIT	Build a reliable and scaleable mobile story
Characters	4 characters	5 leading characters and 17 supporting characters	4 characters
Locations	9 locations	15 locations	3 locations
Starting points	One story location	Any story location	Any story location
Ending points	3 story locations	Any story location	3 story locations
Approximate length	50 minutes	3~4 hours	15~20 minutes
Sense of place	Matched production location	Matched production location	Either matched or unmatched production location
Audience motivation	Characters tell the audience	Tour experience and characters tell the audience	Either characters tell the audience or she makes her own decision.
Production process and effort	One summer and crew of 5	A half year and crew of 6	A month and crew of 3

Table 2 Comparisons among *Another Alice, MIT in Pocket,* and *15 Minutes*

The approach of doing both technology and story development in parallel is more like learning walking with two legs. Tangible characters, actions, plots, and settings have required specific demands for developing technology; on the other side, working technology makes mobile cinema alive and testable. Table 3 illustrates the brief time line of these two types of development.

	2000	**2001**	**2002**	**2003**	**2004**
System	M-Views 0.1	M-Views 0.1	M-Views 0.2	M-Views 0.2	Evaluation
Production		*Another Alice*	*MIT in Pocket*	*15 Minutes*	Evaluation

Table 3 Road map to developing the M-Views system and mobile cinema content

1.2 Contribution

This thesis examines the underlying requirements for coherent mobile narrative and explores two particular challenges which must be solved in order to make a reliable and scalable stream of content for mobile cinema: technology uncertainty (the fact that what the mobile cinema system presents may not be what the creator intends) and participation uncertainty (the fact that what the audience does may not be what the creator expects). The identification of these two types of uncertainty provides not only a measurement for exploring, developing, and evaluating mobile cinema, but also a language with which engineers, designers, and participants can communicate efficiently.

The exploration and analysis of these challenges involved prototyping two versions of the M-Views system for mobile cinema and three prototype cinematic narratives. Small user studies accompanied each production. The iterative process enabled the author to explore both aspects of uncertainty and to introduce innovations in four key areas to help address these uncertainties: practical location detection, authoring tools designed for mobile channels, responsive story presentation mechanisms, and creative story production strategies.

A major contribution of this dissertation is three design principles for mobile cinema development: design for audience situation; leverage mobile experience; and choose story location wisely. These principles discuss audience motivation, story location, production approaches and process, mobile behavior, and content distribution. These three design principles have been discovered through the five-year investigation into both system development and story production and provide practical groundwork for the future evolution of mobile cinema.

In this thesis, I also discuss a parallel development framework for inventing a novel storytelling medium, mobile cinema. This framework is effective for both creating original storytelling forms based on the current wireless communication revolution and developing media systems based on a variety of design requirements for creating, presenting, and distributing mobile cinema content.

1.3 Thesis Roadmap

This thesis is organized as follows:

In the next chapter, I discuss the main motivation for pursuing this five-year research, possible practical applications for these research projects, and the potential engineering benefits from mobile cinema system design.

In chapter three, I first define two research problems: technology uncertainty and participation uncertainty. These two problems guide both system designs and story productions. Then I propose hypotheses for addressing these problems.

In chapter four, I review related work in three domains: oral storytelling and participatory performance, interactive storytelling, and uncertainty research. Through these reviews, I am able to recognize the advantages and constraints of different approaches, methods, and designs that could benefit the creation of mobile cinema.

Based on the related research in chapter four, I present the M-Views system in chapter five. This chapter starts with the introduction of an early version of the M-Views system 0.1, which has provided me many valuable lessons for designing computational mobile cinema systems and stories. Based on these lessons, I suggest three computational design approaches for developing reliable, cost-effective, and scalable mobile cinema systems. The focus of chapter five is the design and implementation of the M-Views 0.2 system.

Using M-Views 0.2, two mobile cinema stories have been created. In chapter six, I first describe the details of the two productions and corresponding story evaluations; then I discuss three design principles and reflect on the key lessons that I have learned through the whole course of this research from both the production and technical points of view.

In chapter seven, I make concluding remarks with discussion of possible future research directions, and reiterate the major contributions of this work.

2.0 MOTIVATION

Why should mobile cinema be investigated? What are the motivations behind this research project? Would this research have any practical applications? What could have been learned through the development of the M-Views system? In this chapter, the main research motivation for building mobile cinema is discussed; then I describe possible future mobile applications that could benefit from my investigation into mobile cinema; finally, I talk about how this research could contribute to mobile computing engineering and mobile human-computer interaction design.

2.1 Mobile Narrative

In this thesis, mobile storytelling means storytelling through a mobile channel. Mobile cinema is indeed an original form of storytelling designed for mobile channels. So, why does storytelling on mobile devices matter? What are the cultural and technological contexts that are shaping the forms of storytelling? What kinds of unique opportunities do mobile devices provide us? How could people create, perceive, and engage in new forms of mobile cinema stories? What user behaviors would there be, which have not been witnessed in previous digital media-based storytelling forms? The initial research motivation was to explore alternative visual forms of storytelling through the mobile channel. The exploration has three aspects: taking both cultural and technological contexts into account, leveraging new opportunities, and understanding the audience's mobile behavior.

2.1.1 The Changing Cultural and Technological Contexts

I address contextual backgrounds from three perspectives:

- Storytelling matters.
- Mobile channels are becoming more important than ever.
- Technology-mediated storytelling could change the course of our society.

First, storytelling is considered a valuable framework for arranging and organizing discrete data into meaningful patterns to think about our life, society, and the world. Bran Ferren[5], former President of Research & Development and Creative Technology for Walt Disney Imagineering, argues that "most people function in a storytelling mode. It's the way we communicate ideas, richly, as well as how we structure our thoughts." Storytelling's methods have been employed by filmmakers, journalists, psychologists, educators, business people, and others in their daily professional activities. Edward Branigan claims that "one of the important ways we perceive our environment is by anticipating and telling ourselves mini-stories about that environment, based on stories already told. Making narratives is a strategy for making our world of experiences and desires intelligible. It is a fundamental way of organizing data" (Branigan

[5] http://www.anderson.ucla.edu/research/marschak/1999-2000/22oct99.htm

1992.) For community members, storytelling is not merely a means for sharing stories, experiences, and emotions, but a carrier of relationship. Smith *et al.* (2000) say that "when people express their personal recollections, their stories become relevant to other members of their community who share a common history. Our own stories and those of our neighbors put a human face on these otherwise impersonal events." Because mobile cinema presents location-based stories, it has a great potential for helping both individuals and communities of people to communicate ideas, express feelings, and reflect lives.

Second, the mobile channel has become one of the most important information channels for human beings. The importance reflects at least two aspects: the number of users and the variety of usage. The Yankee Group forecasts the number of global cell phone users to have exceeded 1.75 billion by 2007[6]. In several mobile societies, such as Japan and Finland, cell phones are must-have daily devices for sharing messages, playing games, checking news, buying goods, dating, and so on. For example, Finland, which has only about five million people, generated more than 1.6 billion SMS messages in 2003[7]. In China, more than 150 billion SMS messages were sent in 2003, including more than seven billion sent in the 10-day spring holiday period alone[8]. With the rapidly increasing number of users, new modes of social communication have already emerged and these modes have been shaping our personal life, community connections, and social structures. In Manila, people used cell phone SMS to organize demonstrations for overthrowing the presidency of President Estrada in 2001[9]; in Japan, cell phone users combine location-based technology and matchmaking software to seek potential dating partners[10]; in New York City, celebrity fans coordinated their community activities through both wired and wireless Internet access (Rheingold 2002.)

Third, Bran Ferren[11] argues that "Every time a technology has been introduced that allows one or more people to do better or more compelling storytelling, like language or writing…, it has changed the course of our society. It has become a permanent part of our lives, and it has a startling impact in establishing the kind of step functions that are characteristic of how our society runs." People have been using wired phones to share verbal stories since they were invented over a hundred years ago. Mobile technology is a main driver to make verbal storytelling ubiquitous, inexpensive, and convenient. Recently, camera phones and 2.5G wireless networks have provided a platform for people to create alternative forms of image and text-based personal stories, Moblogging[12]. Hundreds of thousands of stories have been created and shared through camera phones. This new democratization of storytelling media has proved that mobile phones are also suitable for sharing image-based stories, in addition to verbal stories. This emerging usage trend of mobile devices will help mobile cinema be adopted in the foreseeable future.

In summary, mobile channels have become important parts of our daily life and been used for creating and sharing a variety of forms of stories, such as voice-based, message-based, and

---

[6] http://www.infosyncworld.com/news/n/4121.html
[7] http://e.finland.fi/netcomm/news/showarticle.asp?intNWSAID=23490
[8] http://www.cctv.com/lm/133/42/82430.html
[9] http://www.thefeature.com/article?articleid=39711
[10] www.rolandberger.com/expertise/en/media/publications/RB_Dial_M_For_Mobile_2000.pdf
[11] http://www.siggraph.org/s97/conference/keynote/
[12] http://www.typepad.com/news/2003/07/moblogging.html

image-based. So, what are emerging opportunities that support novel cinematic storytelling via mobile channels? The following section addresses this question.

2.1.2 Leveraging New Opportunities

The first opportunity is that mobile devices offer not only connectivity and mobility, but also emerging features, such as location detection technology, with which people can explore new cinematic storytelling options. Earlier versions of location detection technology were mainly stand-alone GPS devices. Due to the limits of form factors, people had to carry both GPS receivers and cell phones in order to build location-based mobile applications. In 2000, NTT launched the first location-based service, DoCo-Navi[13]. In 2002, this service generated more than a million maps every day in Japan. In the same year, Qualcomm also announced more than five million subscribers using its gpsOne services[14]. In the past few years, numerous emerging location detection technologies and services have been introduced[15]. With the rapidly increasing number of mobile devices with location-detection features, a new computational platform is emerging. It offers people a storytelling medium that is different from any previous wired computational media, such as the CD-ROM or the Internet.

The fundamental difference between mobile cinema and conventional cinema is that the former stories are constructed and shared in time and space; and the latter stories are often viewed in related fixed locations, such as movie theaters, living rooms, or on desktops. To participate in mobile cinema, the audience needs to walk around, pay attention to landmarks, or may buy a cup of coffee while watching stories on mobile devices; to watch a film, the audience is often fully devoted to viewing stories. To create mobile cinema, the author has to take many uncertain situations into account first. For example, how long will the audience stay at one place? Will the audience do something else, in addition to watching stories? Will an audience member talk to other audience members while she walks through a place? The processes of creating and consuming cinematic stories are changed.

If various sources of uncertainty are coped with, the story author can take advantage of the location information to present stories that are designed for this particular space. For example, the author could create a love story that happens in Starbucks, where the lighting is romantic and the coffee smell is sophisticated. The love story would be presented to the audience member while she walks into a Starbucks. In addition to the audio and visual information from the mobile device, the audience could also be engaged with the love story by sensing the ambient atmosphere and coffee scent. This is a simple example illustrating how location technology could support emerging forms of storytelling. In fact, human beings have been telling all kinds of oral stories at old buildings, historical rooms, famous landmarks, and interesting architectural sites without any technology for thousands of years. New technologies yet to be invented could also enhance location-based cinematic storytelling.

The second opportunity relates to the increasing potential for more and more people becoming digital content creators with emerging technologies.

[13] http://www.aic.or.jp/related/Brief/brief06.htm
[14] http://www.qualcomm.com/press/releases/2002/press1117.html
[15] http://www.directionsmag.com/article.php?article_id=548

For example, ten years ago, a high-end digital camera cost more than $50,000 and had to be operated by a crew. Nowadays, the price of a portable digital camera is less than $500 and can be operated by most teenagers. Millions of people have become active content contributors as well as consumers with new tools, such as weblog. A study showed that 44% of Americans—more than 53 million adults—create online content.[16] "The Generation C phenomenon captures the tsunami of consumer generated 'content' that is building on the Web, adding tera-peta bytes of new text, images, audio, and video on an ongoing basis." William Munch argues that two main drivers are fueling this trend: 1) everyone wants to be a creator; and 2) new tools support consumers "to create, to produce, and to participate."[17] For example, Apple sells a set of comprehensive creative tools: GarageBand, iPhoto, iMovie, and iDVD at affordable prices[18]. Microsoft and Sony both offer similar tools for people to create cinematic stories.

The momentum for changing consumers into creators has been accelerating in the mobile domain. Nokia Lifeblog[19] automatically keeps track of photos, videos, and messages, so that the mobile phone user can easily create, organize, and share digital content via mobile phones. Compared to wired weblog, the moblog is more accessible, convenient, and affordable. "Phones that make it easy to send digital video directly to the Web make it possible for 'peer-to-peer journalism' networks to emerge" (Rheingold 2000). For example, Steve Mann[20] and his students have used wearable devices to capture newsworthy events and activities and broadcast them to the Internet. Justin Hall predicts "it is only a matter of time before an important amateur news video is directly distributed to the web, or to ten friends with video-mail in a news chain letter. When that happens, this new form of news distribution will become the news, and then ultimately, it will be no big deal" (Rheingold 2000). The connectivity and mobility of moblog will increasingly turn ordinary consumers into proactive creators in the foreseeable future.

How can people make cinematic stories via mobile channels, and what kinds of story forms will emerge in five years? Mobile cinema is certainly worthy to invest effort in given this trend of more people becoming creators. The development approach is to find self-motivated young students at MIT, identify their creative processes, and work with them to understand their needs for building emerging forms of stories through mobile channels. This approach has resulted in three mobile cinema story productions, two workshops, several collaborative production projects, and a direction for some of the creative work in the Story Networks Group at Media Lab Europe.

2.1.3 Understanding the Audience's Mobile Behavior

The third motivation for building mobile cinema is to understand emerging mobile behaviors. By mobile behaviors, I mean that people have different perceptions, norms, communication patterns, and actions while they use mobile phones. The degree of understanding of mobile

[16] http://www.trendwatching.com/trends/GENERATION_C.htm
[17] http://winksite.com/site/help_bl_view.cfm?blog_id=1092
[18] http://www.apple.com/software/
[19] http://www.nokia.com/nokia/0,1522,,00.html?orig=/lifeblog
[20] http://genesis.eecg.toronto.edu/

behavior will determine the course of mobile cinema development. Why do people behave differently in mobile environments? How could we take new behaviors into account during the system and story development? I use the following example to articulate my argument. Some people may get offended while another person talks via a mobile phone in a public space. Why does this kind of situation happen? Goffman's theories of different public "faces" we present to different audiences offer useful perspectives to analyze it[21]. When a person uses a mobile phone in public, they are simultaneously in two spaces: the physical space and the virtual conversation space. The person often employs a different attitude to talk to other people who are on the other end of the phone. This kind of attitude switch could conflict with the behavioral requirements of the physical space.

For developing mobile cinema, on one hand, we need to know potential user behaviors while they participate in mobile cinema. Would mobile cinema be part of their daily life? Would they need to fully commit two hours to participate in mobile stories? In a mobile cinema experience, how would people decide where to go next? How could the author know people's behaviors, which may be various, depending on each individual? How could personalized experience be created? Does the participant want to watch mobile cinema more than once? These questions are very critical for developing both systems and stories.

On the other hand, building mobile cinema systems and stories has provided me tangible computational systems and stories for observing, analyzing, and studying user behaviors. Many details of mobile cinema experiences were beyond our imaginations during production and concrete lessons needed to be learned through testing. For example, by observing how people participated in *Another Alice,* we learned that telling the audience to go to a building and meet the next character is often not efficient, because a building often has many rooms and the audience has problems finding a specific location. If the author was not aware of this kind of design detail, the audience could not have a smooth story experience. Several story designers also wanted mobile cinema to be a part of their life. During the evaluation of *MIT in Pocket,* we witnessed that a participant was doing laundry while he walked back and forth between his dorm room and an MIT dining room.

Critical research questions, such as how to present coherent mobile stories, how to create a flexible creation framework, and how to communicate with the participant, cannot be addressed until we understand the participant's mobile behaviors. Meanwhile, without building mobile cinema, we lack any concrete means to study these issues.

In summary, the main motivation for building mobile cinema is to explore alternative cinematic storytelling forms based on the analysis of the technological and social trends, new opportunities, and the audience's mobile behaviors. In the following section, I discuss the potential practical applications that could spin off from this investigation into mobile cinema.

2.2 Mobile Applications

There are a number of potential practical applications: mobile touring, mobile learning, and mobile entertainment and mobile games.

[21] http://ess.ntu.ac.uk/miller/cyberpsych/goffman.htm

2.2.1 Mobile Touring

"International tourism is the world's largest export earner and an important factor in the balance of payments of most nations."[22] In 2002, the international tourism revenue was about $474 billion and more than 702 million tourists traveled internationally. The Internet and wireless technology have revolutionized the tourism business and related services in the past decade, but most technology improvements have happened in the e-business and online marketing sectors. Recently, increasing numbers of researchers and companies have started to investigate mobile technologies for enhancing tourism experiences[23] [24]. Mobile cinema research could help other people design better mobile touring systems in at least two domains: technology development and content creation. Many technology and creation problems that have been addressed in mobile cinema, such as story presentation, user behavior study, and content creation, are also critical in building mobile touring systems. For example, mobile touring systems need practical tools for people to create location-based content. What would be sustainable creation models? Would mobile stories be created by any tourists or media companies? Should tourists use their own mobile phones or rent special devices? Would there be any central servers? These issues have been raised during our brainstorming sessions. The making and evaluation of our mobile cinema productions have provided frameworks for conducting further studies on these issues.

2.2.2 Mobile Learning

Wireless broadband and mobile devices have the potential for supporting field-based learning in addition to the current classroom-based learning paradigm. Most current technologies are able to offer quite powerful tools to realize similar experiences. For example, in 2003, the European Commission's Technology Enhanced Learning Center launched two mobile learning projects: m-learning and MOBLlearn. Several initial ideas, innovations, and technologies are being investigated [25].

- *Multimedia guides to galleries and museums incorporating movies, images, music, internet access, and email facilities on hand-held devices (UK and Italy).*
- *Mobile collaboration and ad-hoc networks to support organizational learning.*
- *Incorporating video clips into handheld devices as a way of training people in hospitals (Sweden).*
- *Learning a foreign language through mobile phones (Finland and UK).*
- *Context-aware and location-based services to provide learning content anywhere, anytime, to anyone.*

Patricia Mason, Head of TEL Unit, European Commission Information Society Directorate-General, said: "Those working in mobile learning share a vision of citizens enabled to take part

[22] www.world-tourism.org
[23] http://www.eyefortravel.com/europeconference2004/prog_wireless.shtml
[24] http://europa.eu.int/comm/enterprise/services/tourism/policy-areas/newtechnologies.htm
[25] http://www.mobilemms.com/show_mmsnews.asp?id=1972

in learning activities in any location at any time and to engage in that learning singly or collaboratively at a pace that suits their particular circumstances and needs." I also believe that mobile learning will challenge the current classroom-based learning paradigm for two reasons:

- Young people have been spending much time on mobile devices every day. Mobile communication is familiar and fun for them. If learning content were appealing, young people would love to learn through mobile channels. Mobile learning could become another normal way of learning.
- Mobile learning not only could happen anytime and anywhere, but is also a practical means to connect people and make learning become a social event. Rich Borovoy and Mitchel Resnick's mobile learning project, Folk Computing, has explored this potential (Borovoy *et al.* 2001).

Most of my research work addresses problems such as story construction, narration, human-computer interaction, and user mobile behaviors. These questions are very critical for future learning through mobile channels.

2.2.3 Mobile Entertainment and Games

Mobile gaming is getting popular. New models of cell phones and PDAs offer bigger, brighter color screens, and faster connections. There are all sorts of fun, intuitive, and entertaining games available: action/adventure, puzzles, casino, sports, etc.[26] [27] In Finland, Japan, and Korea, cellular carriers provide various downloadable services for mobile games—ring tones, graphics, and animation—at an affordable cost. Wireless World Forum predicts that the mobile games market will reach $1.9 billon in 2006, rising from US $520 million in 2003[28]. New forms of mobile games will be introduced in the coming three years, while newer generations of mobile infrastructure are expected to be deployed. In addition to the booming mobile game business, numerous mobile entertainments have been launched. For example, in 2003, Samsung released a mobile phone with color TV function, which can receive broadcasts from public access channels and record up to 50 frames of video clips[29]. In 2004, Sony released a new service, Personal Media Assistant, which can stream personalized radio content, such as news, artist information, and community features, to consumers' smart phones[30].

The demands and developments in both the mobile game and entertainment domains accelerate the evolution of mobile communication infrastructures, which could eventually provide reliable, affordable, and convenient platforms for presenting mobile cinema. More and more researchers have started to invent a next generation of mobile games that require very similar infrastructures to mobile cinema. For example, the Mixed Reality Laboratory at the University of Nottingham and Blast Theory, an artists' group, launched a location-based mobile game, *Can You See Me*

[26] http://ewirelessgames.com/
[27] http://cellmedia.com/
[28] http://www.unstrung.com/document.asp?doc_id=36687
[29] http://www.samsung.com/PressCenter/PressRelease/TelecommunicationNews/TelecommunicationNews_20030609_0000006235.htm
[30] http://www.infosyncworld.com/news/n/4727.html

Now? (CYSMN), in 2001[31]. On one hand, three professional performers are equipped with portable computers, wireless networks, and GPS receivers, running through city streets; on the other hand, fifteen online players compete with the three runners through a virtual model of a city.

CYSMN is a mobile game, in which the players know their goals very clearly before they start to play. In the game, there are no story characters, plots, or story structures. In contrast, the author of mobile cinema has to consider all story elements and combine them carefully to emotionally engage the audience. I will compare mobile games to mobile cinema in my later reflection chapter. Nevertheless, the infrastructure that *CYSMN* requires is very similar to the M-Views system. Our development and evaluation could also provide great lessons for researchers like them to use to develop the next generation of mobile games.

2.3 Mobile Computing: Engineering and Design

The development of the M-Views system is a tangible case study that helps us understand mobile computing in two domains: mobile computing engineering and mobile computer and human interaction.

2.3.1 Mobile Computing Engineering

To build a mobile cinema system, we need to understand the current wireless network infrastructure—its capabilities, its constraints, and its trends. Much of the work in my mobile cinema research has focused on building working systems that support mobile cinema production and presentation without much additional investment or major change of the current infrastructure. The reason for doing so is to provide technological test beds, in which other people can have technical access and make mobile cinema for themselves or their communities. Here are several mobile engineering questions addressed in this thesis. How could a location-detection technique be created without adding any additional beacons? How could video content be streamed to mobile devices with or without local caching? How could a location-based mobile story be deployed in another location without recompiling story scripts and location profiles? Should mobile cinema infrastructure have a centralized or decentralized architecture? These engineering questions and experiments have helped me understand how to build scalable, reliable, and low-cost mobile computing systems.

Another type of engineering problem is related to tools for story production. For example, how to represent mobile cinema stories? The representation framework of mobile cinema has to be easy for story creators and be understandable (have interpretable code) by machine. In current M-Studio and the M-Views server, a flag mechanism has been employed. The main advantage of flagging is its flexibility. That means that flagging can represent almost all kinds of clip properties and story structures. The main disadvantage of flagging is its high learning curve. For a beginner, flagging is not always intuitive. Furthermore, we only built three mobile cinema

[31] http://www.equator.ac.uk/Projects/CitywidePerformance/Canyouseeme.htm

stories in-house and all of the three stories were fiction-oriented. Could the XML-based flagging system be flexible enough so that new forms of mobile cinema can be created without major system changes? Building stories certainly helps us to engineer systems that are suitable for developing mobile applications.

2.3.2 Mobile Computer and Human Interaction

The paradigm of computer-human interaction through mobile channels is quite different from desktop-based computer-human interaction. The fundamental difference is that the audience's body becomes a "mouse point" to navigate through space. To participate in mobile cinema, the audience has to pay attention to two information resources: the physical environment and the information on a mobile device. The content creator also needs to take both information resources into account. What kinds of location information are essential and must be presented explicitly? How could the mobile story help the audience navigate through a campus, without giving away too much story information? Mobile devices often have small screens and four or five buttons. How to design a simple interface for interacting with mobile cinematic content is also a big challenge. The development, evaluation, and comparison of two versions of the M-Views system provide test beds for us to explore the new paradigm of mobile computer-human interaction.

2.4 Summary

In this chapter, I have discussed several main questions that have motivated and guided me in investigating mobile cinema: Why is storytelling highly important for human beings? How could mobile storytelling affect the course of our society? What are new opportunities that technological and social trends have provided us? What would the audience's mobile behaviors be in participating in mobile cinema? In addition, I have also described a number of practical applications that could benefit from my research on mobile cinema: mobile touring, mobile learning, and mobile games and entertainment. Finally, I have discussed my research motivation from the perspective of mobile computing engineering and human-computer interaction design. In the next chapter, I want to articulate the hard research problems for creating mobile cinema and propose my research hypotheses.

3.0 Problem and Hypotheses

In this chapter, I first define two major research problems that are addressed in this thesis. These two problems have been identified and refined through the five-year technological development and story creations. Second, I propose my research hypotheses, which are positioned in four fields: location detection technology, authoring tools, responsive story presentation, and creative production strategies.

3.1 Problem Definition

Ideal mobile cinema systems should support constructions of story experience that are reliable, cost-effective, and scalable. Being reliable means that mobile cinema experience is smooth, appealing, and able to deal with various audience responses; being cost-effective means that the producing of multi-threaded mobile cinema stories can be done by ordinary individuals or small groups of people; being scalable means that location-based stories can also be set up at other places that have similar physical environments. To build such a system, what are the hard problems that must be dealt with? The five-year experience of both creating mobile cinema stories and developing different versions of the M-Views system indicates that the major problem is that mobile cinema involves many sources of uncertainty related to network infrastructure, content production, user behaviors, human-computer interaction, technology deployment, and maintenance. These uncertainties lead directly to two possible negative consequences: (1) the audience experience becomes cumbersome, confusing, and unengaging; or (2) the cost of producing coherent mobile cinema content becomes extremely high. A successful system must help create and present reliable and coherent cinematic stories at a relatively reasonable cost-benefit ratio. In this thesis, I focus primarily on two types of uncertainty that must be addressed in order to build a successful mobile cinema system.

Technology uncertainty: *What the mobile cinema system presents may not be what the creator intends.*
Participation uncertainty: *What the audience does may not be what the creator expects.*

3.1.1 Technology Uncertainty

The creator is uncertain about whether or not a particular piece of mobile cinema story can be delivered to the right participant at the expected place and time. This is because the current wireless network is not built for rich multimedia applications. The emerging cellular wireless network is not fast enough (3G is about 200~300kb/s) to deliver video content at a desired size and frame ratio. The 802.11 wireless networks that rely on different subnets have not become seamlessly integrated for mobile applications and context-awareness infrastructures have not been fully established. These imperfect infrastructures inevitably have sources of uncertainty. In order to create mobile cinema systems, I first need to identify the different sources of technology uncertainty. Several sources are as follows:

- The quality of streaming video content on mobile devices is not guaranteed. Low quality of video images can be caused by various factors: wireless bandwidth, limited decoding capability, or the audience walking too fast. Unlike a fixed Internet connection, where its quality of video content is relatively constant, the current wireless connection cannot always provide reliable video quality for mobile cinema.
- There is as yet no perfect location detection solution that works for both indoors and outdoors without additional setups. GPS receivers do not work indoors; beacons require additional setup and expensive maintenance; software-based triangulation technology also needs additional setup and maintenance when a new story is deployed. Furthermore, physical objects within a location may change over time, which makes previous location calibrations invalid. Subnets could also cause tedious problems for the audience walking from one subnet to another. If this issue cannot be solved, the performance of software-based triangulation could remain very slow.
- Mobile devices often have limited power and a complicated user interface. Our earlier experiments indicated that battery power was a critical technical issue, if the story was complicated and required more than one hour's participation by the audience. In addition, our prototype was built based on commercial iPAQs, which were designed for general information purposes, such as *To Do List, Calendar, Contact,* and *Email.* The user interface is another source of uncertainty for mobile cinema users, because misusing mobile cinema devices would cause both technical issues and user interaction problems.

These sources of technology uncertainty need to be identified precisely; otherwise, the story author cannot present the story experience as she imagines it. Some of these sources of uncertainty will be eliminated as better technologies are developed, but others are inevitable for the foreseeable future. In order to identify these sources of uncertainty, we developed both mobile cinema content and computational prototypes as working test beds for observing the participant's use experience, conducted case studies, and did comparison studies. We found that these sources of technology uncertainty are often mixed together from the participant's perspective. This means that when the participant encounters a technical problem, she does not know exactly what is causing the problem. For example, in *Another Alice,* an expected video clip did not display at a certain story location. The participant was not sure if the location-detection was broken, or the streaming video was too slow. This kind of ambiguity could lead to a very frustrating story experience. For the story creator, a practical way to identify technology uncertainty is to build a story and run tests. Based on the test data, he can either modify story production elements, such as story locations, or improve the technology setup, such as re-building story maps. However, this build-test approach could take much time and effort.

3.1.2 Participation Uncertainty

Mobile story creators are often uncertain about the participatory activities by the audience in mobile computing environments, during both design and production stages. In other words, what the audience does may not be what the creator expects. This happens often. As with technology uncertainty, there are numerous sources of participation uncertainty. For example:

- The story participant may not be familiar with particular story locations. In *Another Alice,* most story locations use numbers as building names, such Building E15 or 66. If the participant does not know the MIT campus very well, she often has some problems finding a particular story location. Since location-detection technology only senses the established story locations, missing a story location means that the participant may not be able to continue the story.

- The information given by characters through mobile devices may not be clear enough. For example, when a character tells the participant to go to Building E15, the participant may not know the specific room in E15. While the story is being created, the authors always have some assumptions about the participant's ability to find a location. We have found out that many of these assumptions only apply for a certain number of audience members, not all of them.
- The audience may do something else while participating in mobile cinema. A participant might buy a cup of coffee, or chat with friends while watching mobile cinema. These activities may conflict with story setups. In *Another Alice,* a user was supposed to go from E15 to Kendall Square within ten minutes. However, the user met a friend on the way to Kendall and chatted for about five minutes. The ending changed because of this short conversation. While the story was being designed, the author could not imagine all the kinds of things that the participant might do.
- Even when the audience members are familiar with the story locations, different people have their own ways to describe locations and directions. Kevin Lynch (1960), former Professor of City Planning at MIT, compared numerous mental maps that were materialized on paper through an interview process and combined maps from many individuals. His discoveries were surprising. Location representation and description vary. For example, Boston people give directions by using landmarks almost exclusively; in contrast, people in Jersey City, with extremely uniform architecture, give directions by street number and points of the compass. In mobile cinema, the author may not be aware of this issue.

The reasons that cause the participation uncertainty vary, but I believe that one of the fundamental issues is related to information asymmetry in any interaction. For example, the story creator may know all story locations very well, but the audience may not know all location names. In mobile cinema, story experience is interwoven with both information from PDAs and physical ambience. Although the story creator can try to craft the mobile story on mobile devices as thoughtfully as possible, she is often not sure how much location information could affect the decisions made by the audience. Furthermore, many audience members who visit MIT may already have their own schedules that do not fit a particular mobile cinema. They may only spend two hours in participating in *MIT in Pocket,* which usually requires at least three hours walking through the MIT campus. The author was rarely aware of this kind of user information during story creation.

The second reason has to do with control. Film storytelling is a process fully controlled by the director. In mobile cinema, the story cannot be unfolded without the participation of the audience. Therefore, communication and negotiation can trigger many uncertainties while the audience is participating in mobile cinema. This communication problem occurs often in almost all interactive storytelling systems. Many previous computer-based interactive storytelling systems (hyperlink fiction, interactive TV, and responsive storytelling systems) have similar problems. To participate in interactive stories, the audience often needs to make explicit decisions, such as to click buttons to answer questions, to vote in surveys, or to move their bodies. Unlike computer games, which usually have clear goals and rules, many early computer-mediated storytelling systems encounter extreme difficulty in inviting the audience to participate in storytelling (Jenkins 2002; Pinhanez *et al*. 2000). Asking the audience to make decisions without context is another main source of participation uncertainty.

As with technology uncertainty, participation uncertainty could cause cumbersome participation experience or cost the author great efforts to build reliable story experiences. Furthermore, different people may behave differently when participating in mobile cinema. How could a mobile cinema system be able to sense the users' behaviors? How could the system understand these behaviors and make appropriate strategies to respond to them quickly? How could the system help story designers to prepare different story interactions? How could story designers personalize the participant's experiences? What kinds of communication between the mobile devices and the participant are effective, so that the participant is able to 1) know critical story content and context and 2) still have enough curiosity and motivation to continue a mobile cinema story? These questions need to be addressed for coping with participation uncertainty and making compelling mobile cinema.

In summary, there are two types of uncertainty: technology and participation uncertainty. If no effective solutions had been proposed for coping with them, two negative consequences would have occurred: 1) The participation experience would have been frustrating, confusing, and unengaged; or 2) the mobile cinema author would have had to expend tremendous efforts to design, test, and modify mobile cinema stories. In the next section, I am proposing research hypotheses that lead my design approaches to cope with these two types of uncertainty.

3.2 Research Hypotheses

I hypothesize that, if it is to become widely adopted, mobile cinema needs to cope with various sources of uncertainty. A mobile cinema system supporting better location detection technologies, specially designed authoring tools for mobile channels, responsive story presentation mechanisms, and creative story production strategies can help cope with these uncertainties. An author often has four tasks in building a mobile cinema story: 1) to present a clip at the right time and right place, 2) to make sure a story sequence that is constructed in different places and times is coherent, 3) to provide alternative content or messages to keep story coherence even if the participant interacts in unexpected ways, and 4) to reduce the overall complications of mobile cinema production. The following discussion addresses these four tasks in order to cope with the various sources of uncertainty.

If what the mobile cinema system presents is often not what the creator intends, no one will be willing to create any mobile cinema content. Hence, to deal with technology uncertainty is the first step in building a reliable storytelling medium. However, many current technology imperfections that are sources of uncertainty cannot be eliminated in the foreseeable future. What I can do is to take these imperfections into account and design a mobile cinema test bed based on the current technology infrastructure. For example, I can design a map tool for indicating an overall signal strength of a wireless network, so that the story creator can avoid placing mobile cinema content in a location with weak wireless signals. By doing so, the video streaming quality can be improved, even though the problem of an unreliable network has not been solved directly. To help the audience be aware of specific technology uncertainties, a wireless signal monitoring software can be built to indicate if a failed video delivery is caused by a location-detection problem or by streaming video problems. By checking the monitoring software, the audience can decide to wait at a place for a longer time, or just walk away. The increasing awareness can help the audience make better decisions to participate in mobile cinema based on transparent contextual information.

Participation uncertainty must also be coped with. I hypothesize that new mobile cinema systems need to offer at least two features: an authoring tool for mobile channels and responsive story presentation mechanisms. A story creator can use the authoring tool to design mobile cinema, such as editing mobile cinema content, visualizing story structures, and simulating story interactions with the participant. With the authoring tool, the creator is able to ask simple production questions, such as how a story unfolds differently if the participant walks from point A to point C, instead of walking from point A to point B. The authoring tool should be designed specifically for mobile cinema, which is defined as discontinuous, participatory, and cinematic experience. This means that it should allow the author to design mobile cinema stories not only based on temporal structures, but also spatial structures; it should be able to help the author understand different audience members' reactions to a piece of mobile cinema story; and it should support basic design features that a conventional filmmaker needs.

Furthermore, responsive story presentation mechanisms are needed to monitor the audience's contextual information and feedbacks, and to process, classify, and react to this information, in order to maintain story coherence. Just like a human storyteller, a mobile cinema system should be a good pattern-builder—and also a responsive one. Ideal responsive story mechanisms should be transparent. This means that monitoring processes should not bother the participant, who, therefore, is able to immerse herself into stories. The mechanism should be established by the author and have certain degrees of autonomy. In this sense, it acts like a software agent, which is adaptive, proactive, and autonomous on behalf of the author. In order to design a responsive mechanism, other issues have to be considered as well, such as user mobile behaviors, technology constraints, and production and authoring processes. Given a responsive and reliable presentation platform, how could the paradigm of content creation be changed?

The final point is about story production methods. New methods should support creating compelling story content with reasonable production efforts. A mobile cinema story often consists of multiple story threads for dealing with different types of participation. However, this production approach often conflicts with conventional filmmaking processes. For example, many cinematic story creators assume, consciously or unconsciously, that "[a] story is a controlled experience; the author consciously crafts it, choosing precisely these events, in this order, to create a story with maximum impact" (Costikyan, 2001). This belief restrains the creator from creating new forms of content for the mobile cinema medium. How can mobile cinema storytellers deal with participatory story creation? Production strategies should help the designer create cinematic stories for the open, mobile, and physically distributed media environment. Good strategies should support the designer not only to create appealing content for general audience members, but also to provide personalized experience. For example, the creator may design a narrator character to keep the audience always in the right story loops, even if the narrator is not the main character. If a participant really enjoys all romantic content in one mobile cinema story, the creator may produce additional romantic video clips to meet this desire. The creator may also limit story locations to a certain number, so that the participant would not get confused by too many locations in one story. These are good methods that could be useful for producing and deploying mobile cinema.

3.3 Summary

In this chapter, I have defined a major problem: creating mobile cinema often involves many sources of uncertainty related to network infrastructure, content production, user behaviors, human-computer interaction, technology deployment, and maintenance. These uncertainties could lead to cumbersome story experiences or a high cost of story production. To solve this problem, I have hypothesized four key features for coping with these sources of uncertainty: practical location detection technologies, authoring tools for mobile channels, responsive story presentation mechanisms, and creative story production strategies. In the next chapter, I draw lessons from related research domains and identify my research role in the interactive storytelling field.

4.0 Related Research

This chapter is informed by four main research domains: 1) oral storytelling and performance, 2) interactive storytelling, 3) human-computer interaction, and 4) context-aware computing.

The first set of lessons has been learned from the research domains of traditional oral storytelling and participatory performance. What kinds of skills does a good storyteller usually employ to interact with the audience? How could a theater performer deal with uncertainty caused by the audience's participation? Although I have no intention to comprehensively analyze the theory of participatory storytelling, these lessons indeed inspire me to study some basic storytelling skills that human storytellers have been mastering for thousands of years. Some models of oral storytelling and participatory performance have been used as analogies during the development of the M-Views system and the production of mobile cinema stories.

I then look at the domain of interactive narrative systems and how they monitor, support, and coordinate participation by the audience. I focus on two types of interactive narrative systems: heuristic narrative and co-constructive narrative. The former type of computational narrative systems focus on inviting the audience to experience stories actively; the latter type make the audience to become a creator. The difference between these two types of narrative systems is the level of shared authorship.

Finally, I reflect on two specific research domains: human-computer interaction and context-aware mobile computing. By studying human-computer interaction, I am able to recognize the advantages and constraints of different approaches to coping with uncertainties in human-computer interaction; and by incorporating approaches to context-aware mobile computing, I am able to design mobile cinema in a way that minimizes the uncertainty in current mobile infrastructure.

4.1 Oral Storytelling and Participatory Performance

4.1.1 Oral Storytelling

Oral storytelling is often continuous, verbal, and controlled by the storyteller; and this is different from mobile cinema experience defined as episodic, participatory, and cinematic. However, many oral storytellers present stories interactively in order to engage the audience. This means that the storyteller often pays attention to the audience's reaction and modifies the styles of narration during storytelling. Kroeber (1992) and Brooks (2003) classify participation by the audience into two categories: mental and physical. In light of this thesis, I primarily study the physical participation that "influences the tone and possibly the outcome of the performance." The key question addressed in this section is how the storyteller deals with participation by the audience. How does the author sense the audience? What styles of narration are most appropriate? And how is meaning and coherence created during oral storytelling and theater performance?

4.1.1.1 Reading the Audience

Reading the audience means that the storyteller proactively pays attention to the audience, senses its reactions, and interprets them appropriately. Through reading the audience, the author tries to understand if they are engaged or bored, enlightened or confused, sad or amused. Reading the audience means that the author understands the audience's responses without explicitly asking the audience any questions; therefore the audience can be immersed in the story. Furthermore, good storytellers are able to interpret the subtle contextual responses of the audience. Is the lighting too dim? Is that noise too distracting? (Martin 1996)

Many previous interactive storytelling systems, such as hyperlink fiction and vote-based multimedia systems, have failed for many reasons. For example, these systems had no means to "read" the audience. Asking a reader to choose whatever hyperlinked story paths she wants prevents her from immersing in the story (Costikyan 2001). Many hyperlink story readers do not have a clear goal to achieve; they do not know how the payoff mechanism works in navigating in story webs; and they are often distracted by multiple highlighted links. Furthermore, these hyperlink storytelling systems only have static structures. It means that all linkages among story elements are fixed and cannot be changed no matter how the audience interacts with the stories.

In order to build mobile cinema systems, what kinds of contextual information from the audience need to be read? How do we design a system that could "read" these kinds of information? How could the system understand the audience's response without asking explicit questions that could damage story engagement? How could a system know the audience's contextual responses? These questions need to be addressed in the development of the M-Views system.

4.1.1.2 Multiple Ways of Narration

Marvin Minsky argues that, "You can't understand anything unless you understand it in several different ways." I would say that a storyteller couldn't be a great storyteller unless she can tell a story in various ways. A good storyteller is a "good pattern-builder—and also a flexible one" (Martin 1996). Good storytellers often have multiple patterns of stories in their minds and can present the right patterns to the right people at the right time. Multiple ways of telling stories mean that the main story elements, such as characters, actions, and plots, can be re-organized differently. Oral storytellers have mastered this skill for thousands of years. The differences among various versions of presentation may be as small as timing or wording, or as big as changing narrative structures. Changing timing or wording happens all the time. Regarding the changing of narrative plots, characters, or structures, an obvious example is the oral fairy tale. For example, no one knows exactly how many different versions of the tale of Snow White exist in the world. In some early versions, it is Snow White's mother, rather than the stepmother, who wants to kill Snow White; in other versions, the seven lusty knights save Snow White instead of the seven dwarfs[32].

The advantage of having various presentations is that the author has means to engage the audience or deal with unexpected interjections by the audience. For example, if the audience is confused by a certain part of the story, the author may slow down the presentation or explain a related context. If most audience members are interested in a particular character, the storyteller may twist the story structure to emphasize this character. If the story room is too dark, making people sleepy, the storyteller may speak more loudly her voice.

The ability to tell stories according to the audience responses or environmental context usually requires long-term training and practice. Experienced oral storytellers know their audience members, such as their backgrounds, interests, and expectations. This know-how usually grows through many performances, conversations, and interactions, and is often culture-specific.

Compared to oral storytelling, cinematic stories are often pre-made, meaning that the story structures cannot change during presentation. What lessons could we learn from great oral storytellers in order to build cinematic databases that support multiple narrations? What kinds of challenges do mobile cinema content developers need to overcome to build multiple story structures? Would it cost too much to do? These questions will be addressed in conjunction with a discussion of the development of M-Views and the three productions.

4.1.1.3 Coherent Presentation

The goal of storytelling is to bring a group of characters alive. Bruner (1990) argues that, "A narrative is composed of a unique sequence of events, mental states, happenings involving human beings [usually] as characters or actors. These are its constituents. But these constituents do not, as it were, have a life or meaning of their own. Their meaning is given by their place in the overall configuration of the sequence as a whole, its plot or fabula." Coherent presentation is a key to bringing characters alive. In the context of fictional storytelling, it means that stories are composed of a certain sequence of locations, characters, and actions. Characters' actions follow story curves: they are faced with a situation to which they must respond; their actions change the situation, which leads to a big challenge; the new situation calls for the character to take another action, until a steady state is reached at the end.

Coherence also means continuity, which is a fundamental principle of cinematic storytelling. "In a sequential presentation of content, meaning is derived from the way in which individual shots and sounds are connected to previous shots and sounds. The important function of story progression is that the viewer can construct a meaningful sense of the whole" (Davenport and Murtaugh 1995). From the audience's perspective, most stories are sequential presentation. In mobile cinema, story sequences could be discontinuous and distributed in space, but overall the story is presented on one display sequentially. Keeping continuity is a basic required skill for cinematic story makers.

[32] http://www.scils.rutgers.edu/~kvander/

Inviting the audience to participate in mobile cinema creates the challenge of how to build a computational system that supports the author in keeping the story coherent. Current computer technology neither supports narrative thinking (Schank 1990; Bruner 1990) nor has common sense (Minsky 1988). In other words, we cannot expect computers to make autonomous or proactive decisions for adjusting story structures to cope with the audience's participation and keeping the story coherent. What better technologies could help mobile cinema story authors model, monitor, and keep the coherence of pre-made stories? To answer these questions, I will draw lessons from both participatory theater performances and the evaluations of *Another Alice* to design computational frameworks that help the mobile cinema author to present coherent stories.

In summary, oral storytelling provides basic analytical frameworks for studying how to sense the audience, how to build multiple ways of telling stories, and how to present stories coherently. Although these lessons are not gained through comprehensive analyses of oral storytelling, they help us to brainstorm and build the basic computational models for mobile cinema systems.

4.1.2 Participatory Performance

In addition to oral storytelling, in this section, I briefly describe certain experiments in theatrical performance that inform how we might structure a mobile story. The first example is *Tamara*, created by John Krizanc and Richard Rose,[33] one of the best-known interactive dramas. The second example is a set of experimental theater performances that were mainly contributed by Schechner (1994), who has been devoted for decades to participatory experiments.

4.1.2.1 Interactive Drama

Tamara is one example of a major interactive drama inspiring us to create mobile cinema. In an Italian-style mansion, there are ten settings, ten characters, and ten plays within *Tamara*. No matter whether a play happens in a ballroom, a bedroom, a kitchen, or even the garden, the audience is always able to follow whatever character and story she chooses. While multiple plays happen simultaneously, the audience follows one or more characters from the stairway to the drawing room, spying on a group of nobles and their servants as they show off their loves and infidelities. Halfway through the show, the audience members gather together to gossip about what they have observed, to share personal impressions of different characters, and to decide whom they should follow for the next half of the show.

Krizanc's vision is to free the public. "The author—Krizanc himself—would become only the provider of material. The builder of situations, he who chooses a point of view, would be the public. As it turned out, each member of the public was allowed to choose his own character and actively follow him or her throughout the house in which *Tamara* was performed" (Krizanc 1997). Audience members could also switch, and leave a room with a different character than they had entered with. The two-decade performance of *Tamara* teaches us at least the following lessons: 1) Because it is a multitude of plays (each character has their own story), *Tamara* allows repeated viewings. Because the audience makes different decisions on where to go next

[33] http://www.canadiantheatre.com/dict.pl?term=Tamara

and whom to follow, she has different impressions of the same character each time; 2) Because each audience member makes her own choices to view stories, *Tamara* stimulates additional conversations among audience members. The audience sits down to dinner half-way through and gossips about their theories of who did what to whom. This show provides strong evidence that participatory performance not only offers personalized experience, but also enhances sociability; 3) The performers don't need to take the audience's participation into account within one performance (they could take the audience's feedback and make the next performance better). All plays happen simultaneously and that leaves the audience a great deal of freedom to stay in or leave a room. The qualities of all performances are controlled by the performers; 4) For each show, *Tamara* only allows 100 people to participate (scalability is the main limit for this participatory performance).

4.1.2.2 Environmental Theater

Richard Schechner[34] has been a professor and theater director at the Department of Performance Studies at New York University's Tisch School of the Arts since 1967. As a founder of the Experimental Performance Group, he pioneered numerous participatory theater performances, such as *Dionysius in 69, Commune,* and *Mother Courage and Her Children*, in which he was challenging the boundaries that separated audience and actor as well as redefining what performance means. He wrote that, "Audience participation expands the field of what a performance is, because audience participation takes place precisely at the point where the performance breaks down and becomes a social event" (Schechner 1994). Based on many years of experimental performance in environmental theater, he summarized a list of objections and obstacles to audience participation:

- *The rhythm of the performance is thrown off, maybe destroyed.*
- *All participation is manipulative because the performers know things the audience does not.*
- *Once the question 'who is boss?' is raised between performers and audience, nothing but hostility follows.*
- *Neither the actor nor the spectator is trained to deal with participation.*

He said that those obstacles are caused by "an aesthetics and a social system that are not built to accommodate participatory arts" (Schechner 1994).

To encourage participation by the audience, he recommended changing the social order in participatory performances. For instance, both the performers and the audiences need to accept improvisational as well as prepared rhythms. Participatory performance needs to adopt "a latticelike structure in which highly organized actions exist side by side with more open structures." If a participatory performance cannot continue, let it stop and find a new way to resume. The performers need to be trained as "guide" and "host"; audiences need to learn how to move, speak, and act in the theater.

Schechner's experiences and suggestions indicate that participation is not only a design issue, but also a social setting issue, in which the performer and the audience need to establish a common understanding and agreement. These experimental performances are different from

[34] http://www.news.cornell.edu/Chronicle/02/1.31.02/Schechner.html

Tamara in terms of the role of the audience. In *Tamara,* the audience is a spectator and can choose which play to watch but cannot change the structure of the performance; in Schechner's theater, the audience is often a part of performances, which changes the outcome of the whole play.

What we learn from Schechner's experiments is three-fold:

- Participatory performance doesn't always work out. One of the reasons is that collaborative work requires communication and grounding information. The audience invited to participate often lacks the knowledge of the basic meaning of a play.
- In addition to performing a play, the performers need to guide or coordinate the audience members to participate collectively. Experienced guiders know how to deal with improvisational as well as prepared rhythms.
- Many participatory performances are evolving processes. Both performers and audience members learn new things from communication, collaboration, and the feedback from other audience members. A participatory process may not be as smooth as a truly controlled performance process, but it stimulates communication and learning.

In summary, both *Tamara* and Schechner's Environmental Theater provide us rich examples of how to invite the audience into collective performances, how to communicate and collaborate with the audience, and how to deal with unplanned and random activities. These lessons help me investigate a central question in designing a mobile cinema system: how to create mobile cinema stories that motivate the audience to walk around in physical space. I will address this question later in the story production section.

In the next section, I describe the lessons from interactive storytelling, which has been one of the major inspiration sources for developing mobile cinema.

4.2 Interactive Storytelling

Interactive storytelling implies computational mediation. It adds a feedback loop between tale and audience that is technically mediated. In order to build mobile cinema, I first need to understand computer-mediated storytelling, analyze what are participatory activities in related interactive narrative projects, and know how technology and methods are developed to invite the participant to be an active part of narrative processes.

Before the invention of any written material, stories were mainly created and shared in verbal formats. Each presentation was different, based on different settings, audience, and other types of performance variables, such as time, location, and tellers' styles. After printing was invented, books were able to present stories in a single fixed linear form, in which this happened, then that happened, and then this happened. The difference between oral stories and books was that books are permanence. In a book, the sequences of a story were linear and fixed. The invention of film and TV adopted the same linear forms for presenting stories. The biggest advantage of the linear story form is that the author has the maximal control of story creation and presentation; meanwhile, the biggest disadvantage is that each audience member has to read, listen, or watch the same story sequentially, even if many people have different behaviors, backgrounds, or interests and even if someone has watched or listened to or read the story many times. Also, while the story mimics real life, a linear story almost always simplifies the complexity of real life. The invention of computers, sensors, and networks has provided computational storytelling platforms which give the author reasonable control over story presentation, while at the same time, they can offer the audience personalized experience. One of the early interactive storytelling examples was Perlmutter Martin's *Murder, Anyone?* published by Vidmax Inc in 1982. This laserdisc-based murder mystery story was designed as a sort of parlor game that is reminiscent of the social interaction in reality TV today. An opening scene sets up the situation; from this point on the group watching in the living room lobbies for where to go next. The decision is made by inputting a scene number on a keypad. There is also a game element: the first one to guess the murderer, the method, and the motive wins. The story is carefully orchestrated so that 16 different solutions exist depending on which path is taken. With the widely adopted CD-ROM and Internet technology, various storytelling systems have been explored, designed, and built.

In this section, I classify interactive narrative into two types, heuristic narrative (Davenport 2002) and co-constructed narrative, according to the roles of the participant in computer-mediated storytelling. The first type of interactive story provides me lessons about how the participant has been invited to experience stories actively; the second type of story helps me examine how the participant becomes a creator with the facilitation of the computer. This classification does not pretend to be complete; however, it will help identify the position of my research.

4.2.1 Heuristic Narrative

Davenport (2002) has defined ***heuristic narrative*** as narrative that "fuses a collection of communications channels, content fragments, metadata, audience sensors, story models and computational action-selection algorithms into an interactive storytelling system." In heuristic narrative, computer systems offer the participant various levels of influence on how a story is presented. In other words, a story is presented to the audience by computer, based on partial authorship from the audience. These partial authorships can be explicit or implicit, individual or collective, intelligent or intuitive.

For example, choice-based narrative systems, such as hyperlink fiction (Delany and Landow 1991), question-based narrative system (Domike *et al.* 2003), and voting-based cinema[35] allow the participant to explicitly choose among multiple story paths that are interlinked. The participant's choices are often based on what the participant has read and making a best guess. If it is a voting-based interactive story, the final outcome is decided by a group of people. However, these full-control approaches to building storytelling systems have received unfavorable responses. The participant dislikes controlling story presentations without appropriate context for making decisions (Costikyan 2001).

Today's interactive storytelling is still different from computer games in terms of the authorship and control. In interactive storytelling, the story designers have pre-created several story paths and want to engage the audience through the development of characters and plots; the audience come to enjoy a teller telling a story and rarely have a clear goal regarding their own actions in

[35] Mr. Payback, Interfilm Technologies (1995).

their minds. By contrast, computer games often provide the players with clear goals, such as finding the treasures by killing monsters or solving puzzles. In addition, players often know game rules, which help them to make decisions to participate in games. However, as new technologies and skills are developed, both forms have learned tremendous lessons from each other. On one hand, more and more designers have employed story elements, such as multiple points of view, character development, and plot constructions, in game design. On the other hand, filmmakers have also used many game design tactics in filmmaking [36] [37]. The distinction between the two types of media has been blurred.

Other heuristic narrative systems offer the participant only partial control, but provide more contextual information for the audience to decide about participating in storytelling. For instance, scripted narrative systems [38] have sets of pre-defined and centralized story scripts coordinating the presentation of text, picture, audio, and video story content. Unlike hyperlink fiction, scripted narrative systems often have certain time or sequence constraints. For example, if the audience has clicked on one hyperlink, then another related hyperlink becomes invalid. The advantage of scripted presentation is that the creator has satisfactory control of the presentation processes for maximum effectiveness; the disadvantage is that the authoring process can be extremely complex, time-consuming, and cumbersome if the systems have large story databases (Sparacino *et al*. 2000). To keep story coherence is a critical issue in any interactive storytelling systems, because the audience often does not see the big picture of the story presentation and could react to the story randomly.

Many researchers have extended the scripted authoring methods into responsive storytelling systems, in which the participant can communicate with the computer via more channels. For example, Sashay/sleep Depraved (Baird 1998), Magic Windows (Agamanolis 2001), and Jeffrey Shaw's VR systems [39] can read and interpret the participant's gestures, postures, motion, and voice, if they match the expected pre-programmed ones. Many of these research projects were inspired by conventional theater performances and tried to reconstruct performance-based storytelling through sensors, computers, and new display technologies. However, few scripted responsive storytelling systems are able to provide complex, coherent, and immersive story experiences. There are several key obstacles. For example, each participant's response may be different, and that could create major uncertainty in building constant and complex story experiences. There is no universal gesture set, so the story creators have to teach people: people don't perform gestures in the same way. Multiple sources of inputs were designed for interacting with the participant in intuitive ways, but these multiple inputs often resulted in confusing the participant. Many experimental projects became interactive art installations or games with little narrative character.

In order to deal with idiosyncratic interaction during narration and to eliminate cumbersome authoring processes, several approaches, such as filter-based (Houbart 1994; Nack and Parkes 1997; Agamanolis 2001) and spreading-activation network (Davenport and Murtaugh 1997; Murtaugh 1996) have been proposed. Among the latter, for example, Contour makes presentation decisions based on both the participant's inputs and internal states of the system.

[36] http://www.popmatters.com/film/reviews/t/time-code.html
[37] http://whatisthematrix.warnerbros.com/
[38] http://www.macromedia.com/software/director/
[39] http://www.jeffrey-shaw.net/

The internal states are a set of weights that continuously judge the relationship among all possible candidate story elements in order to select the most appropriate presentation sequence on the fly. Mazalek (2001) implemented spreading-activation network in *Tangible Viewpoints.* The spreading-activation approach has several advantages: 1) it offers an open structure that allows the author to add new content into an existing story database without changing the database's structures; 2) it is suitable for documentary stories, which are often organized by the framework of *Who, Where, When, What, and How*, and 3) its interface displays visually the relationships among different story elements, so that the audience has a better understanding of story context.

4.2.2 Co-constructed Narrative Systems

The preceding projects provide experience that invites participation by the audience, but this participation is still rather passive. Under what conditions could the participant make proactive contributions to a narrative environment? Many ***co-constructed narrative systems*** have advocated that the audience should become more proactive to either (1) communicate with other audiences to extend story context, enrich imagination, and reinforce story engagement, or (2) contribute new story elements into existing story content to enlarge databases. The key motivation for making a proactive audience is to create an engaging story experience. In proactivly participating in story content construction, the audience needs to pay attention to story details, think about story conversations and actions, anticipate coming plots, communicate with other audience members, and decide how to interact with story systems.

Communication among audience members reflects ideal story sharing scenarios where the audiences emotionally engage with stories together, joyfully interact with each other, and possibly learn from each other. In these scenarios, story experiences could be reinforced by the interaction among the audience. "The Wheel of Life," directed by Glorianna Davenport and Larry Friedlander in 1991, was an interactive narrative installation at the MIT Media Lab (Davenport and Friedlander 1995). For each show, audience came in continuously: as one group got through the first environment, the next started. The audience was paired off—one person would be an explorer and the other would be a guide—to immerse in three mythical spaces (Water, Earth, and Air). "Perhaps because it involved partnership and exchange between audience members," Davenport speculates, "it generated an unusual amount of interpretive reflection by the public during and after the experience."

Enhancing the communication among the audience is another approach to enriching story experiences. For example, *Lurker* was created by Lee Morgenroth (1995) to experiment with participatory online fictional narrative. In Lurker, six participants were asked to help hackers through exchanging emails and posting video clips over 5 days. Lurker is more like an online game with rich fictional content, because all participants were given a role (lurker) and a goal (help find a missing person). Enhancing audience communication has become a trend for almost all conventional films, TV programs, games, and interactive storytelling systems. Millions of online chat rooms, bulletin boards, weblogs, and moblogs have been created by either the author or the fan communities. They post the newest gossip about their favorite actors, discuss unseen scenes, and argue about possible story plots in coming episodes. In my research, to increase the participant's sociability is always a critical goal. During the development of the M-Views systems, several approaches to connecting participants have been proposed, such as the messaging approach, competition approach, and puzzle approach.

As noted earlier, in addition to connecting to the participants, many storytelling systems allow the participants to contribute new story elements into existing story content for enlarging story databases. In this kind of situation, the participant is the creator. For example, MUDs' multiple participants remotely log into a computer server and create story spaces collectively.[40] The participant could either be assigned a particular role or self-invent a character in order to build story fantasy with other participants. Many MUDs offer not only a rich collection of commands, objects, and scenes, but also a variety of avatars. These means could make it more convenient for the audience to participate in virtual fantasy and to represent imagination collectively. In addition to role-based play, Brondmo and Davenport (1989) created laser disc-based media tools that allowed audience members to create new links/pathways through the content that others could follow. Multiple authors, through a workshop process, made content. In 1994, Davenport originally created a particular type of co-constructive narrative she called "*Evolving Documentary"* which enables story creators to develop complex stories that can reflect evolving reality. New material can be added into the existing story database by carefully assigning meta-descriptors to related video material (Davenport and Murtaugh 1997). Other co-constructed narrative systems, such as CINEMAT (Davenport *et al.* 2000) provide the participant more spaces to co-create stories. CINEMAT in Mexico City invited two people from the audience to interact with the system. One person ran on a sensor-equipped carpet; the other told a stream-of-consciousness story based on the images appearing on the projection. The crowd laughed all the time, either because the narrator was stuck or because the runner was so funny.

Mobile cinema systems should also provide tools and methods for audience members to contribute their own story materials into the existing story database. These story materials could be as simple as messages, or as sophisticated as video content. The current version of the M-Views system provides several creation tools for proactive participants. For example, everyone can compose new messages and share them with other audience members; the system also provides a simple web-based administration interface, by which new stories can be added through a subscription model; and M-Studio offers a storyboard for supporting designers who lack programming experience.

In summary, what I have learned from these related interactive narrative projects is three-fold.

First, asking the participant to make decisions without giving enough contextual information is unsatisfying. Many interactive systems fail because the participant is uncertain about what to do in order to drive the story forward. Most participation experiences are either boring, such as hypertext fiction, or short, such as *Sashay/sleep Depraved*. Interactive storytelling systems should focus on telling different narrations according to the audience responses. The interaction processes should be transparent without the cost of distracting the audience from the story experience.

Second, each participant often has different reactions to unfolding stories. Not all of these reactions are predictable. Narrative systems should have practical strategies to deal with

[40] An MUD is a computer program, usually running over the Internet, that allows multiple users to participate in virtual-reality role-playing games. (*The American Heritage® Dictionary of the English Language, Fourth Edition)*

participation uncertainties. Winkler (2000) and Davenport (2002) suggest encouraging social interactions among the participants in order to help them learn from each other and keep participation processes smooth.

Third, most narrative systems focus on post-production—editing, authoring, and presentation—rather than pre-production and production. The lack of methods and technologies for interactive narrative pre-production and production increases the complexity of building true participatory storytelling systems, because variable input and feedback requires variable content. New design tools seem necessary to allow mobile cinema authors to deal with both technology and participation uncertainty. How can the author pre-visualize the audience's behaviors? How can new production models be designed to reduce the effort of making multiple story threads? These questions have guided our design of mobile cinema systems and contents.

In the next section, I shift the discussion from storytelling to general uncertainty research. What is uncertainty? What kinds of methods have been proposed by people who are in related research domains?

4.3 Human-Computer Interaction

Designing a mobile cinema system has roots in many research theories and projects in human-computer interaction (HCI), because HCI communities have developed numerous technologies and methods to cope with uncertainty on many sides, such as the human side, computer side, communication side, and collaboration side. In this section, I discuss several points in conjunction with the design of the M-Views system.

One of the early technologies to solve uncertainty on the computer side was WYSIWYG, short for *what you see is what you get*, originally developed by Butler Lampson and Charles Simonyi at Xerox PARC[41]. WYSIWYG has become a standard for desktop publishing, because it eliminates most of the uncertainty between what the user designs on the desktop and what appears when the document is printed. Inspired by the idea of WYSIWYG, an XML-based story script language has been developed in the M-Views system, in order for the story designer to see on the Storyboard of M-Studio (Pan *et al*. 2002) exactly what will display on M-Views Presenter (Crow *et al*. 2003) if both wireless network and location detection work appropriately. By observing people using these tools, we found that a production and presentation of mobile cinema involves multiple processes, which are often operated by different people. Furthermore, participating in mobile cinema depends on multiple sources of interaction: human-to-computer, human-to-human, computer-to-physical environment, and human-to-physical environment. These multiple sources of interaction rapidly increase the complexity of human-computer interaction.

Multimedia applications, such as Authorware[42], Director[43], and HyperCard[44], provide interaction controllers, story script programming languages, and simulation tools to support the creator in

[41] http://www.edge.org/digerati/simonyi/simonyi_p3.html
[42] http://www.macromedia.com/software/authorware/
[43] http://www.macromedia.com/software/director/
[44] Apple Computer. HyperCard: Reference Manual, 1993

design processes and reduce uncertainty from interaction by the audience. For example, Director enables the behavior of graphical elements to be arranged with a score or a timeline. The newest version is integrated with streaming media, a 3D authoring tool, and hundreds of Lingo keywords to control interactive elements. Authorware offers built-in logic, data tracking, knowledge objects, and templates. Although these tools facilitate the implementation of complex multimedia applications for desktop environments, they offer little help for dealing with interaction in context-aware mobile multimedia. In desktop environments, interactions are limited to typing words, clicking the mouse, and playing a piece of media. In the mobile environment, context-aware devices deal with both explicit interactions, such as clicking a button, and implicit interactions, such as walking into a building. None of these tools account for the dynamic interactions of mobile environments. Nevertheless, what I have learned is that an authoring and simulation tool is essential for the designer to create mobile cinema content. The tool should be able to visually display story structure, simulate possible human interaction, and allow the creator to evaluate alternative content development processes.

Although HCI researchers have proposed and implemented various design-based solutions to create better computational products and services (Robertson *et al.* 1977; Shneiderman 1983; Norman 1986; Myers 1988; and Baecker *et al.* 1995), Maes[45] argues that the computer environment is getting more and more complex, and the user is becoming more and more naïve. Horvitz (2002) further points out that "Uncertainty about a user's knowledge, intentions, and attention is inescapable in human-computer interaction." As a result, a different approach to reduce uncertainty in HCI, named "Intelligent HCI," has been proposed and studied by a group of researchers (Maes 1995; Lieberman 1997; and Horvitz 1999). The basic principle of this approach is that the computer makes proactive decisions about human-computer interaction instead of passively following the user's manipulation. Hence, human-computer interaction could become more adaptive, personalized, and flexible. In particular, a range of technologies, such as collaborative filtering (Shardananda and Maes 1995; Konstan *et al*. 1997; Balabanovi'c and Shoham 1997; and Joachims *et al.* 1997) and Bayesian reasoning (Pearl 1988; Horvitz *et al.* 1998; and Maglio *et al.* 2000), have been studied and implemented to assist humans in dealing with massive and unstructured data sets, such as the Internet.

Nevertheless, simplicity, versatility, and pleasurability are key design principles for supporting human-computer interaction because "we are analog beings" (Norman 1999). In order to design the M-Views system, aesthetic basics, task and work flow analyses, human factors studies, case studies, and empirical analyses of design have been deployed. The reason that I mainly focus on design approaches to reducing uncertainties is two-fold.

First, mobile cinema is still in its infancy. To teach a computer about how to deal with uncertain human-computer interaction, the designers themselves first need to understand the basic interaction scenarios. The highest priority, therefore, when I began this research was to build a reliable platform and to create mobile cinema stories. With a prototype in hand, design analysis of formal, experiential, and cultural aspects can help me to identify at least the most critical potential human-computer interaction problems and solve them through iteration and new prototypes. For example, what is the main motivation for the participant to walk from one place to another? What is the main information resource—the stories, the physical environment, or the messages? How does new human-computer interaction enable the participant to communicate with other members? What is an ideal user interface supporting the basic

[45] http://www.acm.org/sigchi/chi97/proceedings/panel/jrm.htm

operational processes for participating in mobile cinema with intuitive steps? These basic needs could be satisfied through good design.

Second, most machine-learning techniques require a critical mass of data, so that use patterns may be identified, recognized, and analyzed. The deployment of mobile cinema is still limited at the MIT campus, and we don't have enough data to start with. Nevertheless, the current M-Views system provides two features for future employment of machine-learning techniques: a comprehensive log system and a heuristic mechanism. The former records all participants' interaction data, such as the participants' location and time information, messages, and operational processes on the M-Views Presenter. These interaction data have been used to analyze mobile cinema experiences in three productions. The heuristic mechanism clusters all story content and classifies it into different groups for providing personalized mobile cinema presentation. This mechanism could be activated for future research as soon as there is sufficient mobile cinema content.

4.4 Context-aware Computing

The context-aware computing community describes itself as people who use "'context knowledge' such as where we are, how we feel and what we have done to drive machines to use our intentions to work with us."[46] Because media content can be delivered discretely in space and time, context-awareness can play an important role in mobile cinema. Context-aware computing research will allow better understanding of how to make mobile cinema experiences more appealing, smooth, and less explicit in the negotiation between the participant and the media system. However, "the user interfaces to mobile applications must be designed to cope with the level of uncertainty that is inevitably introduced into any system that uses wireless communications" (Rodden *et al.* 1998). In order to build context-aware mobile cinema systems as well as reduce uncertainties in mobile channels, in this section, I examine several related technologies and methods studied by context-aware computing communities. First, I investigate both tour guide systems and context-aware reminder applications in order to understand the advantages and constraints of these systems. Second, I outline different approaches to coping with uncertainties in context-aware mobile applications. In particular, I describe three projects that employed different approaches to dealing with uncertainty: a mobile game, wearable guide systems for museum visitors, and uncertainty in intelligent rooms. Although these four research projects are not mobile cinema projects, they provide multiple solutions for coping with uncertainty issues in context-aware computing.

4.4.1 Tour Guide Systems

The earliest computer-mediated surrogate travel multimedia production is Aspen Moviemap[47]. This project, created by the MIT Architecture Machine group, provided a surrogate travel experience of touring around Aspen with recorded time and space. Although this project is not based on mobile devices, it allowed visitors to this virtual rendition of the city to navigate the streets of Aspen, see Aspen at different times of year, and in different centuries, and used

[46] http://cac.media.mit.edu:8080/contextweb/jsp/index.htm
[47] http://www.naimark.net/projects/aspen.html

embedded movies to give visitors a "sense of" the people who inhabit and work in the city. In addition to virtual navigation tour systems, numerous audio guide systems have been created and used for decades[48] [49]. These systems have a very simple content presentation schema: the audience walks to a place, finds the number representing this place, and clicks on the number to receive the corresponding audio clip. This schema is static, requires additional set-up, and cannot provide personalized content.

Early context-aware applications are mobile tour guide systems, which include Cyberguide system (Abowd *et al.* 1997), the Guide system (Cheverst *et al.* 2000), the Hippie project (Reinhard *et al.* 1999), and the HyperAudio project (Petrelli *et al.* 1999). These projects are all aimed at providing location-aware experiences for museum or city tourists. Many of these systems use GPS receivers to detect tourists' locations, parse this information on a server, and send a piece of information back to the audience via mobile devices. All these systems adopt client-server architectures similar to the mobile cinema project, but differ in that they do not focus on the structured narrative aspect. In these systems, each piece of location-based information could be stand-alone without considering the participant's context history or activity history (what the participant has viewed and is viewing). No matter where and how the audience navigates through locations, the story structures are the same. In these systems, the main sources of uncertainty are technological. Since participation uncertainty is not the focus in these projects, little effort has been made to support story coherence. These research projects have recently inspired many location-based services for mobile phones and PDAs. People make use of GPS location-detection technology to generate maps, find nearby businesses, and develop location-based games.

Several location-based tour guide applications focus on promoting social activities around tour experiences. For example, Sotto Voce (Aoki *et al.* 2002*)* is a mobile museum guide system without using any comprehensive location-detection technology. The user of this system has to manually point to corresponding rooms or objects according to the digital images on mobile devices. What it offers instead is a feature of "*eavesdropping,*" which allows audience members to hear other people's audio guide stories. By doing so, this system encourages museum visitors to talk with each other and thus promotes sociability.

4.4.2 Context-aware Reminder Systems

Context-aware reminder systems (Marmasse and Schmandt 2000; Salber *et al.*1999*;* and Rhodes 2000) deal with multiple types of contextual information, such as location, time, activity, and nearby people. Like tour guide systems, these systems aim to trigger reminders at the right time and location for appropriate conditions. The main difference between context-aware reminder systems and tour guide systems is that most reminder systems are always proactive to observe, sense, and analyze the contextual information of the user. In other words, reminder systems are "partners" of the user for daily use; in contrast, tour guide systems may only be used one or two days by people who visit a museum or a city. Since the design purposes are different, these two types of systems may take different approaches to modeling user behaviors, calibrating contextual information, and presenting mobile content. Although context-aware reminder

[48] http://www.espro.com/
[49] http://www.tourmate.com

systems have provided valuable lessons in terms of modeling context, building profiles of the user, and delivering messages appropriately, they rarely consider the aspects of adaptive presentation and content authoring, which are critical in mobile cinema.

4.4.3 Coping with Uncertainty

What are the lessons that I have learned from related context-aware computing to cope with uncertainty? Although, to the best of my knowledge, there is no similar research project that also focuses on mobile cinema, different research projects have taught me particular lessons about coping with uncertainty from various standpoints. In the following section, I describe three projects: Benford *et al.*'s CYSMN project, Sparacino's Sto(ry)chastics project, and Byun and Cheverst's ubiquitous computing environments (Byun and Cheverst 2003). Through the discussions and comparisons, I outline the state-of-the-art research results on coping with uncertainty in context-aware computing domains.

Mobile Game (CYSMN): Benford *et al.* (2003) built a mobile mixed-reality game: Can You See Me Now? (CYSMN). In this game, three runners who are professional performers, equipped with mobile computers, wireless networks, and GPS receivers, run through city streets to compete with fifteen online players who virtually "run" on the Internet. According to their experiments, they described three sources of uncertainty in CYSMN: GPS, 802.11b networking, and other technical failures. Similar to what we have encountered in our earlier version of the M-Views system, location-detection is the most significant source of uncertainty. These three sources of uncertainty not only affected the three runners' performances, but also disturbed online experiences. To deal with them, the Benford team employed two strategies: hiding uncertainty and revealing uncertainty. For instance, they implemented a position validation scheme to filter out locations where GPS receivers do not work appropriately; this approach avoids, and thus hides, areas of uncertainty. They also deliberately reveal uncertainty to mobile game participants by providing dynamic color-coded maps of good and bad network areas. Both strategies are practical for coping with uncertainty. In the M-Views system, the MapAgent also employs a similar technique to visualize wireless signal strengths, so that the story designers can know the physical story environments first, and then can choose story locations wisely.

Wearable Guide System (Sto(ry)chastics): Sparacino (2001) took a probabilistic approach (Bayesian networks) to developing a personalized interactive museum guide system, Sto(ry)chastics. This system collects the museum visitor's information, such as where and how long the visitor stays in the museum, classifies this information according to a Bayesian network, and presents the visitor with personalized guide stories. In other words, this system employs a learning mechanism that proactively monitors the museum audience's behaviors and makes story presentation strategies on the fly in order to cope with uncertain activities quickly. Sparacino says that this Bayesian network-based learning approach has the following prerequisites. "It needs to have the depth of content of a scripted system, the flexibility of a responsive system, and the autonomous decentralized architecture of a behavioral system. It also needs to go beyond the behavioral scheme and respond not just by weighing stimuli and internal goals of its characters, but also by understanding the user's intentions in context, and by learning from the user" (Sparacino 2003).

In Sto(ry)chastics, story presentation is driven by sensor data, so that this system is able to cope with participation uncertainty according to a learning process, instead of a fixed modality of user-story interaction and story authoring. However, this system is only the first step in the direction of having learning-based adaptive techniques. It only senses three types of visitors: busy, greedy, and selective. Its classification model is a static mode rather than a dynamic mode. This means that even if a visitor starts with a busy behavior and later becomes a selective visitor, the system cannot reconsider its classification during one visit. This implementation may affect overall story personalization. For example, in the evaluation of our mobile cinema production, *15 Minutes,* 20 participants generated 18 different walking paths. Classification based on pre-determined types may not work well for fictional mobile cinema stories. Furthermore, the internal logic of mobile cinema stories is very critical for maintaining story coherence. The author needs to have practical ways to craft story structures. Sensor-data-driven story presentation may only be suitable for a particular type of guide systems.

Uncertainty in Intelligent Rooms: Similarly to Sparacino's premise, Byun and Cheverst (2003) argued that managing uncertainty is a key in building adaptive context-aware systems. In their intelligent room project, they classified various sources of uncertainty into three categories: sensors (margins of error in a given sensor), discretization of continuous-valued attributes (deriving higher-level contexts from a group of lower-level or raw contexts), and rule extraction from context history (ambiguity of learned data size and training processes). To cope with these sources of uncertainty, they summarized the following solutions:

- *Through initial analysis, it appears that applying fuzzy representation of context and fuzzy decision trees is one possible solution for overcoming the limitations of discretisation of continuous-valued contexts.*
- *The membership value resulted from the classification using a fuzzy decision tree can be used as a way for representing uncertainty of suggestions to the user.*
- *We have also determined that utilising appropriately large size of context history can provide more certain rules, more coverage of rules, and more resilience to incomplete data.*
- *By providing explicit explanations regarding the level of uncertainty for proactive adaptations, the user can override the adaptation and even amend the rules and/or context history.*

Although their experiments and findings were published at the *Workshop on User Modelling for Ubiquitous Computing 2003,* and the M-Views system was built in 2002, there are many similar findings in terms of managing various sources of uncertainty. For example, wireless sensors inevitably produce erroneous or inaccurate data under various conditions: a long distance between the M-Views Presenter and an access point, interference among various access points, or the participant walking too fast. Furthermore, the interpretation of lower-level data is often associated with uncertainty. When a participant cannot receive a video clip at an expected location, she may not know the exact reasons. Location-detection software could be measuring wrong data; the streaming video could be slow; or there may even be some bugs in the story script. The uncertainty of interpretation could make the participant very frustrated. As with Byun and Cheverst's strategies for coping with uncertainty, we have provided explicit explanations to the participant, such as sending her a message, so that the participant may amend her strategy.

In addition to these three examples, I have also drawn lessons from some theoretical works. For example, Dix (1995) has proposed a theoretical framework to cope with uncertainty. He suggests 1) matching pace to task or 2) matching task to pace. James and Sayrs (1999) have proposed to develop the Route Planning Uncertainty Manager (RPLUM) tool kit, which applies uncertainty management to terrain analysis and route planning for future battles. Three approaches have been proposed: "(1) creation and use of spatial-temporal-probabilistic tuples, (2) support for diverse probabilistic dependencies, and (3) visualizations of battlefield uncertainties." In particular, they focus on visualizing a shared perception of battlespace uncertainties for the purpose of route planning.

4.5 Summary

In this chapter, I have taken a closer look at four related research domains: oral storytelling and participatory performance, interactive storytelling, human-computer interaction, and context-aware computing. These reviews have helped me understand several critical research issues: how human storytellers deal with various audiences' responses; how theater performers invite the audience to participate in performances; how related interactive storytelling systems tried to change the relationship between the author and the audience; and how a variety of sources of uncertainty have been coped with in both human-computer interaction and context-aware computing. In the next chapter, I articulate the development of the M-Views system.

5.0 M-Views

In this chapter, I discuss the M-Views system, starting from the prototype of M-Views system 0.1 and a mobile cinema production, *Another Alice.* M-Views system 0.1 was a prototype for producing and presenting mobile cinema content based on standard hand-held GPS technology. *Another Alice* is a murder-mystery presented on the M-Views system 0.1. This story is an experiment to help me understand important questions, such as how the audience interacts with a mobile cinema story; what kinds of location knowledge the audience needs in order to play the story; and what possible production approaches there are to making mobile cinema content. Both the development of M-Views system 0.1 and the production of *Another Alice* not only reflect major technology uncertainty and participation uncertainty in system developments, but also encourage us to create better systems and stories.

Based on these discussions from both M-Views 0.1 and *Another Alice,* I propose computational approaches to designing the M-Views system version 0.2: enhancing context-awareness capability, providing authoring and simulation tools, and offering flexible presentation schemes. These three computational approaches lead the design and implementation of the M-Views system, which consists of M-Studio, M-Views Server, and M-Views Presenter. I will articulate the evaluation of M-Views in conjunction with two mobile cinema productions, *MIT in Pocket* and *15 Minutes,* in the next chapter.

5.1 M-Views 0.1 and *Another Alice*

M-Views version 0.1 was developed in the year 2000. The project goal was to develop a prototype for exploring some basic ideas of mobile cinema production and presentation. What would a basic mobile cinema system be? What would be a simple location-based mobile fiction story? What were the main obstacles to overcome during story production? How could the audience interact with the system?

5.1.1 Design Approaches and Implementation

The design approach to building version 0.1 was three-fold:

- Fast prototyping: a system should be built within one month. The system should be suitable to current wireless network and location detection infrastructures.
- Easy interacting: a system should be easy-to-use by any participant. The participant should only need to know simple instructions in order to interact with stories.
- Supporting a fictional mobile story: a system should support structured mobile stories, such as murder-mystery stories. Therefore, critical mobile narrative elements, such as characters, settings, actions, and plots, should be evaluated.

The architecture of version 0.1 (Chart 1) consists of two modules: a wireless client and a laptop-based server. The client is based on an iPAQ and equipped with a GPS receiver, which detects the location of the user. It also has an 802.11b wireless card and runs client software, which is responsible for the following tasks: 1) reading GPS data from the receiver, 2) sending the GPS data to the laptop-based server via wireless connection, 3) playing back a video clip if the physical location is matched to the story location information on the server, and 4) sending a feedback to the server after a clip is played. The server manages several tasks: 1) processing a set of story scripts that describe how pieces of story clips should be played out appropriately, given different walking paths by the audience; 2) comparing the GPS data from the client to the location data in the story scripts; 3) sending streaming video clips to the client via wireless networks; and 4) monitoring when a video clip has been delivered and updating the parameters in the story scripts.

The M-Views client 0.1 was written in eMbedded Visual C++ and run on the Windows Pocket PC operating system; the server 0.1 was written in Java, running and contained in Apache Tomcat. The Servlet used standard HTTP POST to communicate with the client and HTTP GET for administration and maintenance. A PocketTV media player was used to stream video content on the client.

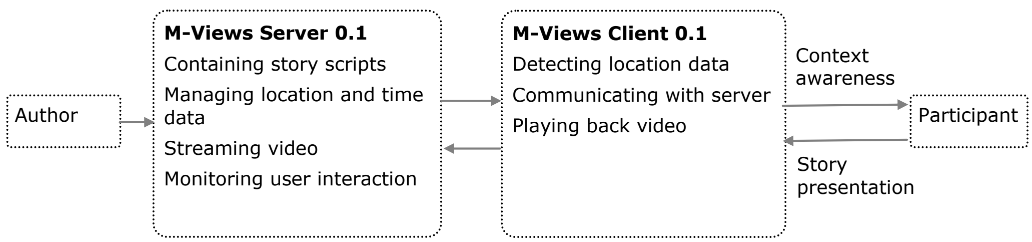


Chart 1 The architecture of the M-Views System version 0.1

5.1.2 Story Production

Built upon the M-Views version 0.1 system, *Another Alice* is a story in which the participant is the investigator. The participant starts the story by receiving an email telling her to go to the office of Professor Eugenie. There, the participant meets one of three characters: a UROP student, the professor’s secretary, or a pre-med, depending on the time of day. All the characters tell the participant that Professor Eugenie is dead and ask the participant to go somewhere else if she wants to find out more information about the professor. The participant has to make a decision about whether or not she should trust the character, where she should go next, and how fast she should run. Since each character is telling the story from his or her perspective, each narrative is different. The participant has to walk from one location to another within a certain time to watch one of the endings of the story.

Locations**:** All *Another Alice* stories happen at MIT. There are nine locations: Student Center, Kendall T Station, Building N14, Walker Memorial Building, Media Lab, Kresge Oval, Building 68, Medical Center, and Au Bon Pain. Seven of the nine places are located on the east campus of MIT, so that it is relatively convenient for the audience to walk through the story.

Character	**Timeline** →													
Student	O					68	O	A	14	O				
Secretary	O	W			K	68	L			M	O			
Pre-med	O	W	A	68	L	W					O		M	K
Alice											68	M	K	
Location: O=office, M=Medical Center, K=Kendall Square, W=Walker, 68=Building 68, and so on														

Table 4 The story structures of *Another Alice*

Table 5 Five possible story paths

Characters: There are four main characters: the premed student, Issac, David, and Alice. Both the premed and Issac are the students working for Professor Eugenie; David is her secretary and Alice is her daughter.

Plots: All stories start from the same location, Professor Eugenie's office. Based on the time of day, the audience is given one of three characters, premed, Issac, or David to "meet." Depending on walking paths, the audience watches different pieces of the stories. For example, at Building N14, the premed student tells the audience her research role on the professor's research team. She claims that she deserves credit for building a revolutionary nano-medical chip for delivering medicine based on real-time diagnosis information. At Au Bon Pain, David tells his audience stories about the professor, her research, her daughter, and the whole research team. At Walker Memorial, Issac recalls time when he worked for Professor Eugenie, who had a great vision about the future medical delivery technology. Depending on the audiences' walking paths and the timing of participation, there are three endings: 1) the premed student is able to run fast enough to skip away and leave the audience with a mysterious ending; 2) the audience collects all necessary clues and runs fast enough to catch the premed student, who confesses what she did; 3) the audience witnesses a meeting in David's office. Issac confesses that he forgot to fill a medical prescription for the professor and feels very guilty. He believes that his mistake caused the death of the professor. Table 4 illustrates the overall story structures of *Another Alice*; and Table 5 shows five possible story paths generated by the audience.

5.1.3 Lessons and Discussion

About a dozen preliminary evaluations were conducted at the MIT campus. The evaluation procedure consisted of: 1) finding an MIT student who knows all the story locations; 2) showing her how to use the M-Views client 0.1; 3) sending her an email and telling her the first story location; 4) escorting her to walk through the story; and 5) collecting all log files from a laptop server. These preliminary evaluations provided the following lessons:

- The biggest source of technology uncertainty was the GPS receiver, which was not working appropriately among tall buildings. The MIT campus is extremely crowded and consists of nearly one hundred buildings in the 154-acre campus. The participants had to walk around a particular building several times to trigger a piece of video clip. In some testing situations, a nearby story piece was mis-triggered because of inaccurate location data. For example, one user watched the conversation clip, and decided to go to Building N14. However, the story spot was only effective on one side of the building. So, the user was confused and was unable to decide where to go next. If this kind of problem occurred, the escort had to tell the user the missed information. Given this location-detection uncertainty, the cost of deploying *Another Alice* could be extremely high.
- The second technology issue is related to the limitations of iPAQ design and form factors. For example, the combination of a GPS receiver and an 802.11b wireless card consumes incredible power. Walking through the MIT campus three times may take the audience about one hour. At some point, the audience had to walk around a building to find the right place for watching a clip. Therefore, *Another Alice* could not always present one of the three endings due to this power issue. Furthermore, to display a video clip during the daytime, the LCD screen of an iPAQ must be turned to the brightest level, which also rapidly consumes power.

- The main participation uncertainty is caused by the content design of *Another Alice* and story interactions. Because the dialogues on the video clips were the only information that the participant had access to, every word became important. If the audience missed an important hint or misunderstood some dialogue, she would have problems following the story. Furthermore, the participant needed to know the exact story locations well to find the place where the character asked her to go next. Because the whole story happened over nine places and some places had large areas, it was not always easy for the audience to identify the right spot. Another issue is that while a participant waited at a place for a while, she didn't know if there was a technology problem or if the story was designed to show nothing. Therefore, the story design had to be extremely thoughtful in terms of giving the audience clear information in appropriate ways.
- *Another Alice* had a typical tree-like structure, so that the participant understood his or her role and mission. In several scenes, the characters gave the audience a short time slot to arrive at the next place, which created tension because it required a fast decision and quick movement so he could catch the next piece of story on time. The tradeoff from the tree-like structure was that some participants didn't like to follow what the characters explicitly told them to do and they would become bored.

In summary, the development of M-Views system 0.1 and the production of *Another Alice* provide tangible experiences and lessons for how a simple mobile cinema system could be built up in a short time frame and how the system could support a mobile cinema story production. The preliminary testing also revealed both technology and participation problems. In the next section, I will propose the computational approaches to designing the M-Views system version 0.2.

5.2 Design Approaches of M-Views 0.2

Several key questions are addressed in this section. How could technology uncertainty and participation uncertainty be coped with, based on current infrastructures? What kinds of tools does the author need to keep story coherence, given various participation by the audience? Could mobile cinema stories reflect the spirit of the production locations? Could one mobile cinema story be deployed in different locations with similar settings?

The version 0.2 was developed in 2002 and 2003 and has been used for several mobile cinema stories. This version involves the development of (1) an 802.11-based location-detection solution for both indoor and outdoor situations, (2) an XML-based M-Views client-server for presenting mobile cinema stories, (3) an authoring tool for mobile cinema creation, and (4) two mobile cinema stories for exploring fictional mobile cinema production and experience. The main design approaches to building the system are three-fold.

- Enhancing context-awareness capability: according to the previous lessons, two kinds of context-awareness capabilities need to be improved: physical context and story context. The former indicates where the participant is and when; the latter indicates which clips have been watched by the participant and how.

- Providing authoring and simulation tools: the system should provide the author basic creation tools for supporting the design of episodic cinematic stories at distributed locations.
- Offering flexible presentation schemes: the system should be able to present mobile cinema stories based on both a prepared audience's interaction and some random user situations.

5.2.1 Enhancing Context-awareness Capability

The system should offer a location-detection solution that 1) can sense both indoor and outdoor locations; 2) can be set up and maintained at reasonable cost, and 3) should not make the form factor of the client cumbersome. Three solutions have been tested: GPS, IrDA adapter, and a wireless triangulation software. The advantages and disadvantages of the three solutions are described in Table 6.

	GPS	IrDA	Triangulation
Advantages	Existing infrastructure	High accuracy (1 foot) No additional components on the client side Little cost of power	Decent accuracy (3~9 feet) Working both indoor and outdoor No additional components Little cost of power
Disadvantages	Middle-low accuracy (20~30 feet) Only working outdoors Additional components Consuming additional power	Requiring additional setup and high cost of maintenance Only working indoors	Requiring additional setup

Table 6 Comparisons among GPS, IrDA, and Triangulation

The comparison indicates that a software-based triangulation solution is the most appropriate one, given the current wireless infrastructure and the constraints of mobile devices. 802.11 signals usually work both indoors and outdoors on the MIT campus. A software solution doesn't require additional hardware components, consumes little power, and is easy to maintain. In order to make this solution serve the needs of mobile cinema production, two additional issues must be taken into account: 1) most mobile cinema creators are not technology-savvy; therefore, asking them to set up the profiles of wireless signals creates too many technical obstacles. The M-Views system must provide ordinary creators with easy-to-use tools for setting up story locations. 2) The software-based solutions must be integrated seamlessly into the rest of M-Views system, so that the deployment and the maintenance of the system are manageable. In version 0.2, a MapAgent has been created for detecting, triangulating, and profiling 802.11 wireless signals on a single iPAQ device. The details of the theory, design, and implementation of MapAgent will be discussed in the next chapter.

In mobile cinema, maintaining story coherence is critical, because mobile content is presented discontinuously and distributed in physical space. The system should also be able to detect the audience's interaction, such as which clips the audience has watched, how long the audience stays at a location, how fast the audience moves from Point A to B, and how the audience communicates with other people. All interactive data needs to be transferred back to the story server in real time, where a new story scripting language is needed for representing mobile cinema. The story scripting language also needs to be comparable to both the current wireless and wired network infrastructure. In addition, there are various needs for creating different types of story scripts. For example, one creator wants to associate weather information with a piece of video clip; another creator may need to link pieces of mobile cinema stories together according to a common theme; or there are many future requirements that cannot be imagined right now. In M-Views, an XML-based story scripts scheme has been created. A story script defines both the story content, such as video, image, or sound clip, and the story context, such as the location of the clip. This scheme is designed for defining story context, and flexible enough for meeting these story design requirements. In the next section, the method for creating and managing story scripts will be discussed.

5.2.2 Providing Authoring and Simulation Tools

The process of producing *Another Alice* indicated clearly that story creators need new authoring tools for mobile cinema for various reasons. First, the current non-linear authoring tools, such as Apple's *Final Cut Pro* [50] and Adobe's *Premiere* [51], don't support location-based story construction because mobile cinema's presentation depends not only on time and location, but also on the internal structure of a story. To create a mobile cinema story, the authors are often uncertain about many issues, such as the distance between two buildings, the appropriate pathway given starting and ending points, and the story density at a particular location. Second, mobile cinema is not an experience fully controlled by the author. The audience's participation also affects the outcome of a story. Third, different authors have their own preferences for creating stories. For example, the author of *Another Alice* experimented with two ways of constructing the story: character-based and plot-based. Furthermore, cinematic production for mobile devices is a visual creation process, in which the author needs to manipulate, play with, and monitor visual and audio content through intuitive interfaces.

In addition to authoring tools, the author of *Another Alice* and people who participated in two mobile cinema workshops (2002 and 2003) also expressed the strong need for simulating the mobile cinema experience on desktop. For example, a story creator was not sure what the exact story sequence would be if the audience didn't go to a specific location. Another creator wanted to know what a story experience would be if the audience paused a mobile cinema story for a half hour due to an important phone call. The desire to have a simulator was mainly due to the author's need to keep the coherence of structured stories and engage the audience in episodic, physically distributed story experiences.

To build the authoring and simulation tools for the M-Views system, I found that the following components are useful.

- A clip editor: each mobile cinema clip needs to be associated with contextual information, such as location and time. A clip editor helps the author to define the basic information that determines how a sequence would be constructed and played.

[50] http://www.apple.com/finalcutpro/
[51] http://www.adobe.com/products/premiere/main.html

- A storyboard: it allows the author to create story sequences in time and space. The author can construct her story by placing clip thumbnails into linear grid, which represents time thread, character thread, or even plot thread. This storyboard is a basic working place and new authoring features are built upon it.
- A location editor: geographical information is critical to the author because most mobile cinema stories are strongly associated with physical settings or reflect the spirit of locations. A location editor could help the author to see the layout of a particular place and design stories specifically for the location. Important design information, such as the density of a story web, also needs to be viewed by the editor.
- Story scripts: in *Another Alice,* the state of a clip is mainly determined by three factors: the location of the audience, the clips that have been viewed, and other contextual information. In order to design a practical story representation mechanism, there are two challenges: 1) how to encourage the author to build coherent stories; and 2) how to minimize the authoring processes. The first challenge means that the author needs to design multiple-thread stories or various presentation schemes for one story. Given the audience's participation, the story structures would be changed in both planned and unplanned ways. How could story coherence be kept according to the story scripts? The second challenge reflects the balance between the effort of authoring and the quality of the story (robustness of the presentation). In other words, if the author is able to predict all possible walking paths by the audience and eliminate all technology uncertainties, he would have full control of the story presentation. However, to gain full control in mobile cinema is almost impossible. Therefore, there is a tradeoff between the effort and the quality. To overcome these two challenges, a story flag mechanism has been designed. This mechanism works in a straightforward way: each story is associated with a set of flags that must be true in order for the clip or message to display. The value of the flag is maintained in a centralized story server and is changed as the story continues. The author can use the flag to define and manage the state of each clip and the overall structure of a mobile cinema story without writing a line of code. More details about how story flags work and the advantage and disadvantage of this mechanism will be discussed in the chapter on M-Studio.
- Simulation tools: Although the idea of building emulators for mobile devices has been used for years[52] [53] [54], my approach takes it one step forward, by taking the participant's activities, story creation, and context-awareness into account. For example, a map simulator visualizes possible walking paths by a participant according to the assigned starting and ending points. A story line generator gives an author the capability of creating possible story lines, subject to constraints of location, time, and flags. Simulation tools could put the author in the position of the audience and help her to simulate the final presentation experience before the story is deployed in mobile environments.

In summary, mobile story creators need tools to help them think about various possibilities for participant interaction with mobile stories. With these tools, the creator would be able to design mobile stories in space and time, visualize story structures, geographical distributions of story elements, and time factors, and simulate different ways for the audience to participate. Just as WYSIWYG did for desktop publishing, the new authoring and simulation tools would help the creator convert uncertain participation into certain actions.

[52] http://www.emulator-zone.com/doc.php/mobile/
[53] http://msdn.microsoft.com/mobility/downloads/sdks/
[54] http://www.forum.nokia.com/main/0,6566,033,00.html

5.2.3 Offering Flexible Presentation Schemes

The lessons from oral storytelling, participatory theater performance, the development of M-Views system 0.1, and the preliminary evaluation of *Another Alice,* indicate that a flexible presentation scheme is essential for building and presenting mobile cinema. The fundamental reason is that the mobile cinema experience cannot always be fully controlled by its creators, and the production of visual content may take much effort and time. Being flexible means that the mobile cinema presentation system is able to deal with unplanned situations. For example, if the participant gets lost, the mobile cinema system would send him or her a message to help the participant go back to the story loop. Also, messages are always useful for giving additional information to a participant unfamiliar with the location.

To offer flexible presentation schemes, two approaches are employed: messaging and a heuristic function. Messaging is a convenient way for the creator to communicate with the participant or to simulate the story experience before any video production. Several story creators also imagined that the participants could use the messenger to communicate with each other to enhance story experience. In particular, people who wanted to use the M-Views system to develop mobile games were interested in using messages to organize group-based games. Several mobile cinema developers also suggested sending messages from the characters in order to engage the audience. The heuristic function could be used to present personalized content. It works as an automatic rating mechanism, keeping track of the user's log files, and creating the user's profile. For example, if the participant liked to watch romantic content, the mobile cinema system would collect this kind of information. As long as a threshold is reached, more romantic mobile stories could be delivered to the participant.

In this section, I have proposed three computational approaches to designing M-Views 0.2: creating better context-awareness technology, offering both authoring and simulation tools, and designing flexible story presentation schemes. In the following section, I will articulate the implementation of the M-Views system version 0.2.

5.3 Implementation of M-Views 0.2

The M-Views version 0.2 system design[55] consists of three main modules: M-Studio, M-Views Server, and M-Views Presenter. Chart 2 shows the architecture of version 0.2. M-Studio is an authoring tool, helping the mobile cinema story author to design, simulate, and adjust mobile

[55] The design, development, and evaluation of the M-Views system has been teamwork under the supervision of Glorianna Davenport. I initially proposed the idea of Mobile Cinema and the overall architecture of the first and second versions of the M-Views system. Steven Chan and David Crow designed the first version of M-Views Presenter; David Crow also designed the second version of M-Views; Carly Kastner implemented M-Studio; Christina Chen, David Crow, Daniel Mcanulty, and Isaac Rosmarin created *Another Alice;* David Crow, Lilly Kam, Debora Lui, Chris Toepel, and Dan Bersak are the main contributors to *MIT in Pocket.* Lilly Kam and Olivia Lee helped with the evaluation of the M-Views system and mobile cinema stories.

narratives. M-Studio directly generates XML-based story script codes and transfers them into M-Views Server, which presents mobile cinema stories according to these codes. The server is also connected to M-Views Presenter, which is based on wireless PDAs. M-Views Server and Presenter work together for detecting contextual information, delivering messages, parsing mobile cinema story scripts, handling video streaming, administering data, and managing other mobile cinema story production and sharing tasks.

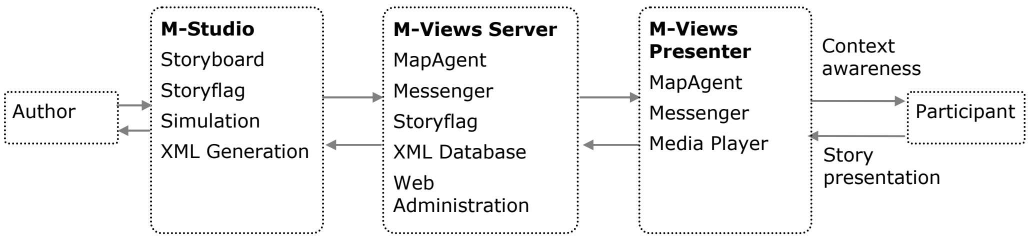


Chart 2 The architecture of the M-Views System version 0.2

M-Studio offers an author several critical frameworks: Storyboard, Storyflag, Simulation, and an integration layer between M-Studio and M-Views Server. Storyboard is a visual interface for designing basic mobile cinema content and structure. For example, it allows the author to associate a particular story video clip with a location; as a result, this clip is only played back while the audience is physically in the location. The interface of Storyboard looks like conventional nonlinear video editing applications, but it is designed to cope with problems which any mobile cinema creator will inevitably face. For example, Storyboard allows the author to construct mobile story plots both temporally and spatially. It means that a story sequence is selected and presented by the M-Views system based on both the time when the audience views it and the location where she or he is. Storyboard provides a storyline editor, clip editor, location editor, and location view to help the author construct, visualize, and modify mobile cinema content in intuitive interfaces. Storyflag provides a framework for the author to define how a story should be presented in order to maintain overall story structures. For example, clip B must be presented after clip A has been shown. To help define these kinds of clip relations, Storyflag offers a simple, but very flexible clip flag structure, which consists of a set of requirement clauses and a set of results clauses. Each clip is associated with both types of flag clause. The requirement clause contains flags that all must be true in order for the clip to play, and then certain flags' values are updated according to the set of results clauses. The Storyflag frameworks can be automatically converted to story scripts; thereby mobile cinema story structures can be created without any complex programming. The Simulation framework is a set of tools for the author to test how a mobile cinema unfolds, given different interactions by the audience. This framework consists of a Story Simulator, a Map Simulator, the storyline generator, and a tree explorer. These tools can simulate not only a range of possible walking paths, but also step-by-step story unfolding processes. For example, a storyline generator can automatically create all possible story paths given sets of story flags. The tree explorer offers the author an interactive interface, with which the author can choose the first clip, and then the explorer will offer all possible clips as the second clip candidates.

M-Views Server consists of several components for monitoring the audience's contextual information, calculating story scripts, and presenting story content. These components include a MapAgent server, a messaging framework, a flagging system, an XML database, and several open-source web server packages. MapAgent, which consists of a server and a client, is a location detecting and tracking engine that uses 802.11 wireless signal strength information to determine the position of M-Views Presenter. On the server side, MapAgent records the profiles of wireless signal strength at certain locations where mobile cinema stories take place. Using a combination of two techniques (trajectory tracking and triangulation), MapAgent can determine whether or not the audience is within a particular story location, given a tolerance radius *R.* The messaging framework supports two types of messages: client-client messaging and client-server messaging. As a result, an audience member can not only receive messages from the server, but also share information with other members. The flagging system parses the XML story scripts generated by Storyflag of M-Studio in order to present mobile cinema stories according to XML scripts. An XML database includes both story scripts and media files. Finally, Linux, Apache Tomcat, Java, and other open-source packages are used for developing and maintaining M-Views Server.

M-Views Presenter has two main features: presenting mobile cinema stories and creating/editing story maps. A presentation process involves three components: a messaging framework, a media player, and a MapAgent client. The messaging framework is generalized, and can be used to display and edit messages from the server. Sensor data is polled and sent to the server along with the pertinent user input. When a message arrives at an M-Views Presenter with an associated media URL, a streaming media player is launched. In order to create/edit story maps, the author can use the MapAgent client to calibrate map coordinates and to manage location profiles based on wireless signal strengths.

How to use the M-Views system to create mobile cinema stories depends on different creators. For example, a creator may start from M-Studio and use it for designing story concepts and testing story plots. She may create several story structures and simulate them with the Story Simulator or the Map Simulator. Another creator may start from calibrating story location profiles to create story maps. Therefore, she has a good sense of where the wireless signal is strong and where it is weak. By walking through physical locations, she is also able to record interesting building or street landmarks in order to build several ideal walking paths. Afterwards, she may use M-Studio to develop characters, story arcs, and possible actions. I will discuss specific production and workflow examples in later sections.

In the following sections, I first describe M-Studio. What are the main features that it offers? What are the main components? And how do these components work together? Second, I discuss M-Views Server and Presenter, answering similar questions.

5.3.1 M-Studio

M-Studio consists of three frameworks (Chart 3): the first framework is Storyboard, which provides an author a visual interface for connecting story content with specific geographical locations and laying out basic story lines; the second framework is Storyflag, which supports the author in defining the relationships among story content and rules for unfolding story content; the third framework is Simulation, which enables the author to simulate story threads, possible walking paths by the participant, and possible story structures given applied story flags. The Storyboard framework is the central design place for constructing story lines, as well as providing story scripts to both the Simulation framework and the M-Views server.

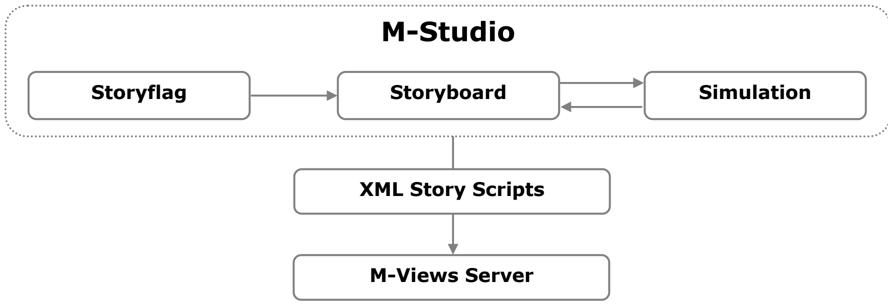


Chart 3 M-Studio framework

The Storyboard framework includes:

- A storyline editor, in which basic story lines are constructed and visualized.
- A location editor, by which story content can be placed on an interactive map. With the location editor, the story author can have a picture of how clips or messages are distributed in space.
- A clip editor that allows the author to create and modify both story content and context. For example, a message can be associated with a hyperlinked video clip and a picture thumbnail.
- A location viewer that offers an alternative way to visualize location-based story structure in addition to the storyline editor.

The Storyflag framework offers a way of maintaining the state of a story. Each clip or message is associated with a set of flags that must be true in order for the clip or message to display. The value of the flag can change as the story continues, which could lead to a change of story structure. The structure of flags is straightforward, with a requirement clause of flags and a result clause of flags. Each clause has a mapping from flags to their values. These two types of flags determine the relationships among all clips or messages. In M-Studio, there are different types of flags, such as numerical flags, list flags, clip flags, and date/time flags. A story author can also create new types of flags according to the needs for designing story structure.

The Simulation framework consists of four simulators: a Story Simulator, a Map Simulator, a storyline generator, and a tree explorer.

- A Story Simulator can produce a sequence of story content that would be viewed by the audience that took that path. Story sequences can be created either automatically or manually.
- A Map Simulator can allow an author to see many possible computer-generated walking paths by a participant and help the author to understand how a story may unfold in different ways in space and time.
- A storyline generator gives an author the capability to generate possible story lines, subject to constraints of location, time, and flags.

- A tree explorer provides an alternative way to visualize story paths and uses the same algorithm as the one used in a storyline generator.

The details of all components will be described in later sections.

5.3.1.1 Authoring Processes

An authoring process is a process of defining the contextual information of all clips in a mobile cinema. In other words, all clips will be associated with both physical and story context that determine how a mobile cinema will unfold through the participation by the audience. An authoring process is also a process to create story structures, which can be either very simple or quite complicated. Although there is no guideline to create story structure, and different authors may employ different approaches to authoring a story, Chart 4 illustrates the main steps for authoring a mobile cinema story.

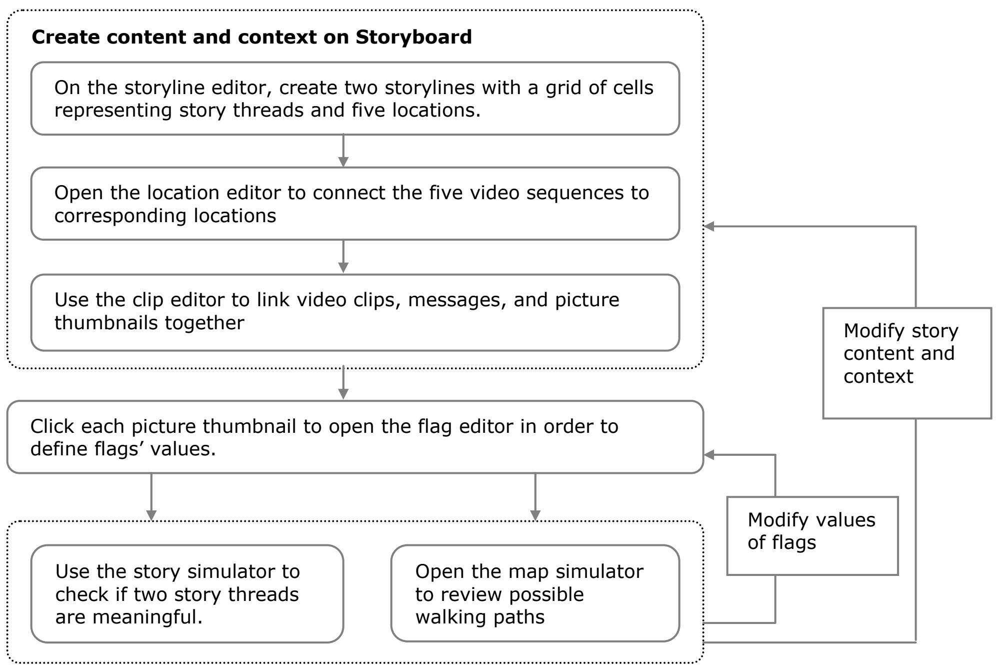


Chart 4 Main steps for authoring a mobile cinema story

First, both story content and context need to be created. On the Storyboard, main story lines can be constructed; on the clip editor, video clips and picture thumbnails can be associated together; on the location editor, each clip can be linked to physical location information. Second, an appropriate story structure needs to be designed. In this case, the structure is straightforward. However, the tricky part is the process of setting up flags' values. For example, after sequence *d* is viewed, the flags of both sequence *c* and *e* must be false, so that a participant cannot see those two sequences. Similarly, after a participant views sequence *c* or *e,* the flags of sequence *d* need to be false. Third, both Story Simulator and Map Simulator can help to validate the setup of flags. For example, the two simulators can check the values of the flags of sequence *a* and *b* to make sure that the clips are not repeatable.

5.3.1.2 Design Strategy of System Structure

Although each component of M-Studio aims to solve a particular problem, they share some common design strategies.

First, the M-Studio structure aims to satisfy some basic needs for authoring discontinuous, participatory, and cinematic mobile stories. These needs include constructing story lines, creating and editing contextual information, viewing distributed story elements, testing the participation by the audience, and understanding different ways of presenting mobile stories. These needs are discovered by observing different design approaches deployed by different authors in conjunction with actual story productions and authoring processes. Both *Another Alice* and *MIT in Pocket* showed many concrete needs of constructing mobile cinema. For example, the creators of *Another Alice* originally used Post-its, campus maps, and whiteboards to lay out story elements. These design tools inspired us to create computer-based Storyboard and a location editor in a particular way. In *Another Alice,* several characters often compete with each other for the audience's attention. If the audience follows a secretary character, she or he would be able to view a pre-med's clip. These kinds of creative constraints suggested the Storyflag framework.

Second, the current structure design takes participation uncertainty into account. One of the design strategies is to increase predictability through simulation and visualization. In *MIT in Pocket,* the story clips as a whole involve more than 15 different locations and 40 characters. A complete experience is intended to unfold over the course of an entire day, making it impractical to physically walk around the campus and simulate it all at once. Furthermore, an evaluation through physically walking around a campus often requires the evaluator to record the paths he or she has taken. In many cases, the evaluator cannot find out where the exact cause of a particular problem is. The Map Simulator is designed to allow the author to interact with an animated map and adjust time and context information item by item, even though the author is uncertain about how a participant will interact with the story. A story simulation process may only take as little as a few minutes; therefore, the author is able to do many rounds of simulation in order to understand the possibilities of how a story unfolds.

Third, the design of M-Studio provides both simplicity and flexibility to satisfy various design needs. On the one hand, M-Studio provides a simple-to-use graphical interface for ordinary creators who have no knowledge of programming or the implementation of the system. Therefore, they need to use visual components, such as storyline and map editor, to create location-based stories. M-Studio can also automatically generate story flags based on the visual structure laid out in the storyboard, using simple generation rules. On the other hand, sophisticated story creators can use flags to define clip context, edit presentation constraints, and create multiple-threaded mobile stories. Furthermore, advanced story creators can access XML-based story scripts that are generated by M-Studio and modify them on the server side. I will discuss this feature in the section on the integration with M-Views Server.

5.3.1.3 Storyboard

Main function: (1) create story content and context and (2) build main story lines.
Components: a storyline editor, a clip editor, a location editor, and a location viewer.

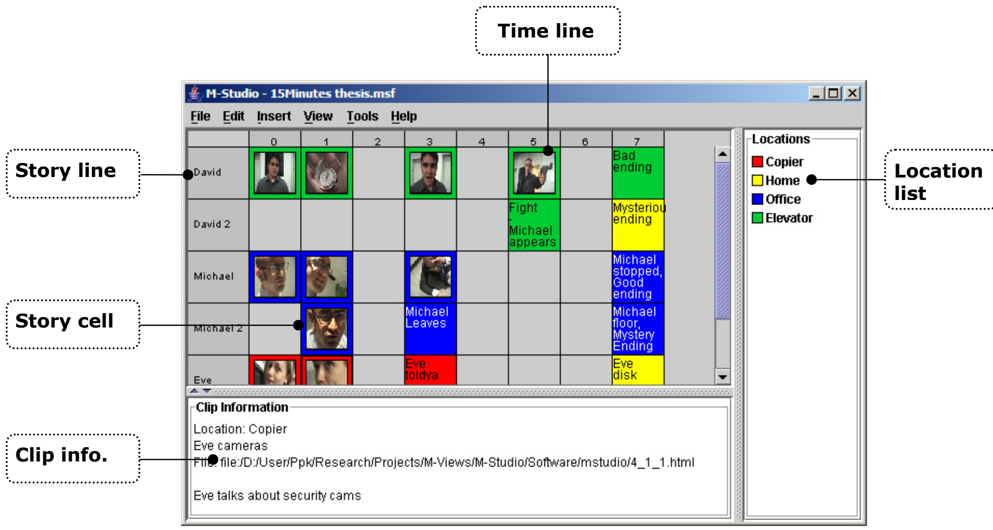


Figure 1 Storyline editor

On the storyline editor (Figure 1), the x-axis represents story time and the y-axis represents story line, making a grid of cells, and each cell represents either a video clip or a message. The grid indicates and keeps track of the relationships among clips. By right-clicking on a particular cell, related information, such as a clip name and a location name, is displayed at the bottom of the storyline editor. On the right side of the editor, there is a window for listing locations in different colors.

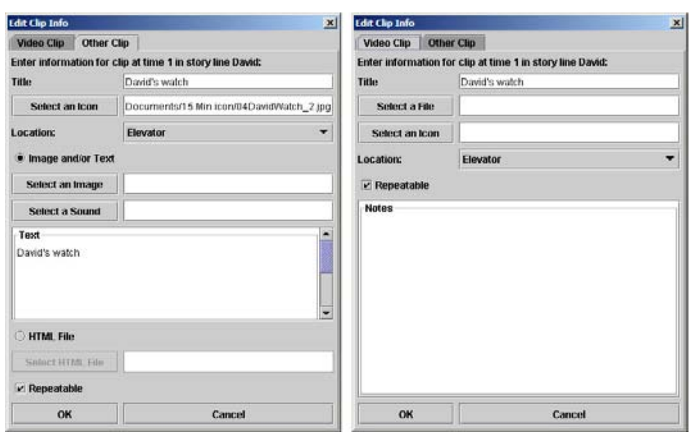


Figure 2 Clip editor

A mobile cinema clip is an abstraction that encapsulates story content data, such as a video clip, an image, a text file, or a piece of sound. With the clip editor (Figure 2), an author can create and modify clips. There are two types of clips: "video clip" and "other clip." "Video clip" represents a standard video clip and is used for post-production; "other clip" represents other forms of clips, such as images, text, and sound. These "other clips" are useful for designing a mobile cinema story idea or for pre-production.

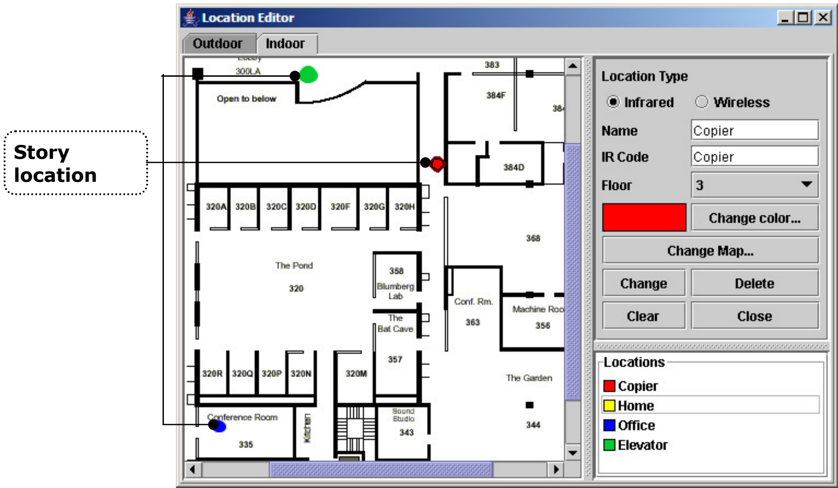


Figure 3 Location editor

The location editor (Figure 3) provides a convenient way for an author to create, edit, and visualize the locations where story clips are presented. The editor also offers other useful information, such as what other story locations are nearby or an estimated distance between two places. Each location has a name, a unique ID, and a visual representation. The colors of locations correspond to the colors of clips. M-Studio supports three different location detection schemes: GPS, 802.11 Wi-Fi, and infrared. The location editor is designed to coordinate with the M-Views server; the server can therefore parse and match the same predetermined sets of location data.

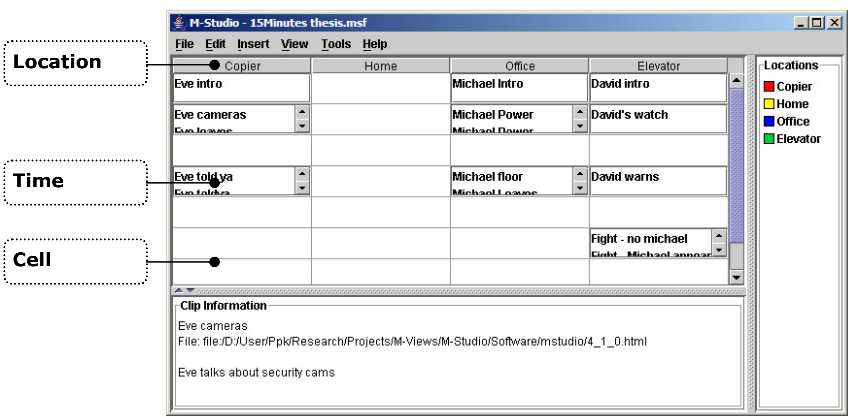


Figure 4 Location view

In addition to the storyline editor, the location view (Figure 4) offers another visual way of representing story structures. The x-axis represents story locations; the y-axis represents story time as a column of cells; and each cell contains a list of clips, representing that more than one clip could happen in the same location at the same time. The benefit of the location view is threefold:

- It allows the author to see where and when the story happens.

- It is useful for enabling story producers/directors to organize scenes to be shot on the basis of location.
- It is also useful in determining story density and anticipating participants' interactions.

In summary, Storyboard allows the author to create story threads, associate clips with locations, modify clips' properties, and visualize a story in several ways.

5.3.1.4 Storyflag

The Storyflag framework consists of two parts: clip flags and a global flag table. The clip flags are combinations of both a set of requirement flags and a set of result flags. The requirement set of flags must be true in order for the clip to play, and then the result flags update the values of flags on the global flag table according to the specified function after the clip is played. The working flow is illustrated in Chart 5.

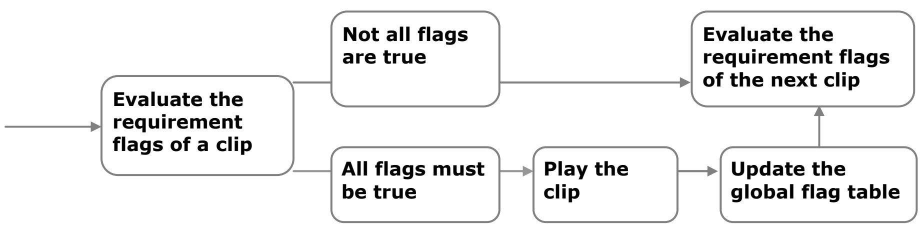


Chart 5 Storyflag framework

The following code is an example copied from the M-Views Server.

```
<Requires>
    <Fight expr="==" value="0" />
    <Michael_value expr="==" value="4" />
    <Eve_value expr="==" value="3" />
    <_ expr="?RT" value="ID22,1,-1" />
    <_ expr="?LS" value="ID22" />
</Requires>
 <Message>
    <Text>Eve is grateful for your help...</Text>
</Message>
<MediaURL>:http://keaton.media.mit.edu:8080/Media/11EveCard.mpg
 </MediaURL>
 <Results>
    <Eve_value oper="=" value="4" />
</Results>
```

In this code, there are five flags that must be true in order for 11EveCard.mpg to play. After the clip is played, the value of the flag of Eve_value is set to 4 and the global table is updated.

5.3.1.4.1 Flag Types

M-Studio offers several types of flags and a mechanism for story authors to create new types of flags. These types of flags have been designed in the process of creating two mobile cinema productions. Some types of flags are more generic than others. The details of story flags are described in Appendix One.

Global flag: This is the most basic type of flag, consisting of a single integer value that is evaluated and updated as clips play. The evaluation process includes: 1) fetching the values of a clip's flags from the flag table and 2) comparing the given values using the given comparators. Global flags can also provide an easy approximation for Boolean values by using the setting operators and equality. For example, an introductory message needs to be played at the beginning of a mobile cinema story. All video clips can have a global flag called INTRO, whose initial value is 0. After the introductory message is played, the value of INTRO is updated to 1. To evaluate whether or not INTRO is true, the equality operator can be used to see if its value equals 1.

Multi-valued flag: This is an extension of the global flag, consisting of a list of values. Clips can require that a certain value be a member or not be a member of the list. Members can be added or removed from the list. For example, a video clip has three characters, which are represented by three values. A story author can use these three values to track how often characters have appeared in the story.

Clip flag: This type of flag is used to track what clips the audience has already seen. Clip flags are an extension of the *Multi-valued flags,* allowing the author to require a set of clips to have been seen or not seen in order for another clip to play. Because clip flags are used to track the audience's interaction, these flags cannot be set by other operations of result flags.

Relative time flag: This type of flag allows the author to set timing between clips; story pace could thereby be controlled. With relative time flags, an author can define a clip trigger and the minimum and maximum amount of time that can pass in order for a clip to play.

Date/time flag: This allows an author to set a range of dates and times during which a clip can play. Unlike a *relative time flag,* a *date/time flag* is evaluated using absolute date and time information.

Context flag: In addition to using the above flags, an author may need to use other possible types of context information, such as weather, to augment mobile cinema experience. *Context flags* allow an author to associate clips with any other kind of context data. There are two types of context flags, one that handles numerical values and one that matches descriptive strings. Both types are based on multi-valued flags, allowing the author to specify all the possible values that meet the context condition.

5.3.1.4.2 Heuristic

There are four ways of presenting multiple clips at a given location.

- Pick the first clip in the list
- Pick a random clip from multiple clips
- Use a rotating stack model
- Use a heuristic method

A *heuristic* is a particular semantic term associated with a clip. The heuristic framework allows an author to use a rating mechanism for presenting personalized mobile cinema content. For example, if the participant always chooses to watch romantic content, the mobile cinema system would notice this kind of information, and rate related romantic clips with higher weights. As long as a predetermined weight threshold is reached, more romantic mobile stories could be delivered to the participant. This mechanism is created in conjunction with managing the user's profile. We believe that in the near future, more mobile cinema content will be created, so that a heuristic mechanism will be necessary.

5.3.1.4.3 Flag Editor

The flag editor allows an author to create and edit all flags associated with a given clip, rather than using a direct scripting language. The editor has three sections: 1) one for editing requirement flags, 2) one for editing results flags, and 3) one for editing heuristics. Each section consists of a flag panel and an editing panel. The flag panel lists all the flags currently associated with the clip and their values; the editing panels are customized for the given flag types. The structure of the editor is illustrated in Chart 6.

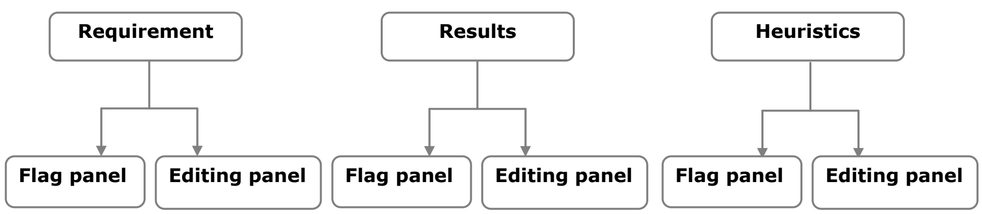


Chart 6 Three sections of the flag editor

The flag editor offers a series of functions:

- Creating new flags, or adding existing flags to other clips
- Specifying or examining flag values
- Modifying a flag or a group of flags
- Adding flags to all clips, or to clips in a particular time slot, storyline, or location
- Defining rules for creating flags

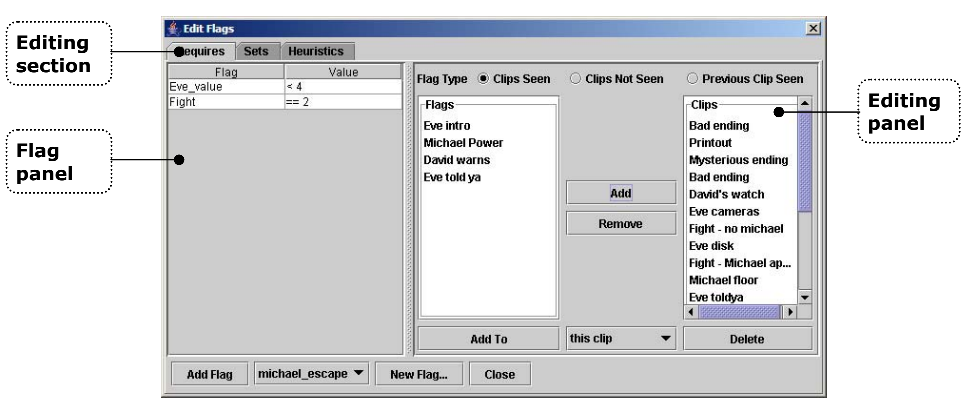


Figure 5 Flag editor

5.3.1.4.4 Automatically Generating Flags

If each clip needed to be associated with flags manually, the whole authoring process could be very time-consuming and cumbersome. The Flag editor offers several methods for generating flags automatically according to certain rules: 1) "impose sequential ordering"; 2) "order based on clips seen"; 3) "order based on previous clip"; and 4) "allow crossover." The interface is shown in Figure 6.

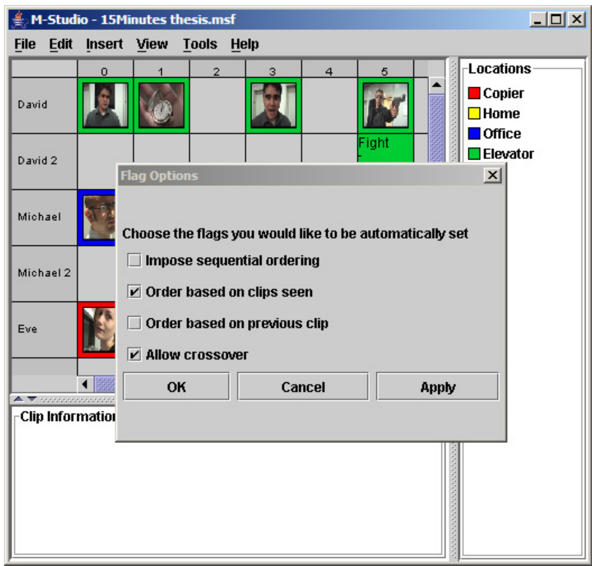


Figure 6 Generating flags

For example, the rule "impose sequential ordering" assumes that any events that happen in the same time slot take place at the same time. Therefore, after seeing an event in a certain time slot, the user cannot see events from earlier time slots. In other words, the story is presented sequentially, without any flashbacks. Both "order based on clips seen" and "order based on previous clip" allow an author to structure flags based on the story grid of the storyline editor. Finally, M-Studio also offers a rule of "allow crossover," which means that there is another clip at the same time and location as another possible preceding clip. Through this rule, a clip can not only require that a preceding clip have been seen, but that it was the most recent clip to have been seen.

5.3.1.5 Simulation

The Simulation framework consists of four modules: a Story Simulator, a Map Simulator, a storyline generator, and a tree explorer. The reasons for developing this framework are four-fold:

- A static visual Storyboard framework cannot fully help an author to design mobile cinema content, because mobile cinema is often a participatory experience. With the Storyboard framework, the author can see story lines and where walking paths may take place, but the Storyboard does not present mobile cinema stories from the user's point of view.

- Although the Storyflag framework provides flexible ways for authoring mobile cinema stories, it has the potential to make an authoring process much more complicated. The author needs to make sure that 1) all flags are set up correctly; 2) all flags' values are appropriate; and 3) the relationships among all flags are right. Even when the general order is correct, an author needs a way to determine if the story sequences are meaningful and coherent.
- Even if all authoring processes are found to be correct, the author also wants to imagine the audience experience if certain unexpected things happen. For example, a user may receive a personal phone call and discontinue a mobile cinema story for a half hour. How would the disruption change the whole experience? If an important video clip was not presented appropriately, how can a new message be sent out to tell the user necessary story information?

To address these difficulties, a set of simulation tools has been created.

5.3.1.5.1 Story Simulation

This tool allows an author to simulate a linear story sequence on a timeline-based interface. Through storyline simulations, the author can review story arc, continuity, and understandability, as well as checking the flag settings. This tool is also useful for the author to check sequences that cross between story lines.

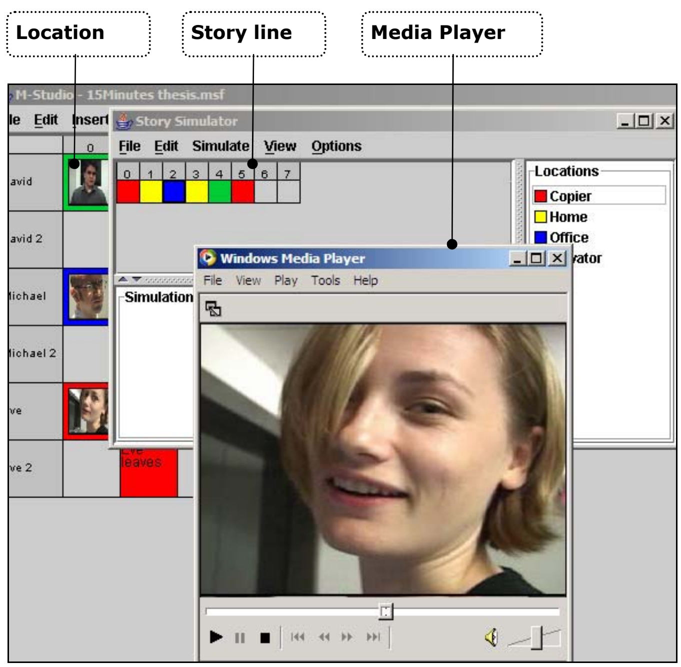


Figure 7 Generation by time

Creation and Modification: To create a story sequence on the timeline, the author just needs to click on clips in the storyline editor sequentially, and then the same set of clips is input into the timeline correspondingly. There are two options on the Simulator: Interval and Default Context. The former option asks for "How many minutes should each time step represent?" The latter one offers a window for inputting date or context information. The Simulator also provides "Clear" and "Clear Times" functions to ease the task of re-creating a timeline.

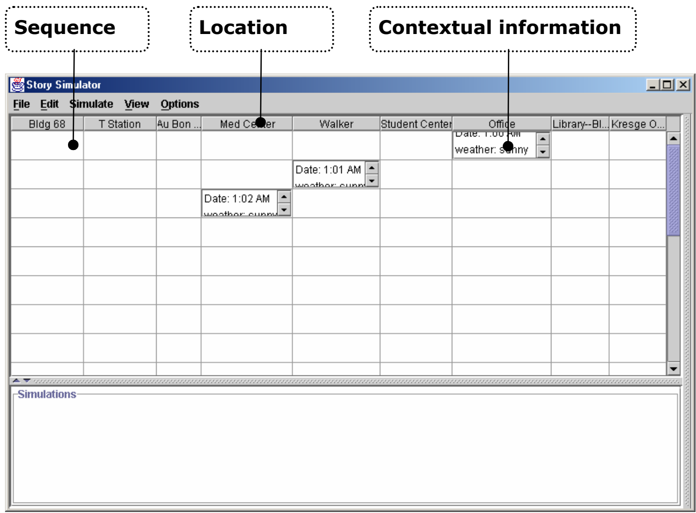


Figure 8 Generation by location

Visualization and Generation: There are two ways of visualizing a timeline: "By time" and "By location." The "By time" mode is shown in Figure 7 and visualizes temporal story structure. The "By location" mode is shown in Figure 8 and allows the author to specify what the exact time and contextual conditions would be as these locations are visited. In this mode, rows represent story sequences; columns represent locations; and cells contain different types of contextual information. Data in these two modes are synchronized, so that the author can easily switch between them.

To play a simulation, the Simulator asks for contextual data from the global flag table, and generates different types of play lists. If all clips are video data, the Simulator creates a temporary WVX play list that can be recognized by Microsoft Media Player. If all clips are texts and pictures, a web page is generated. If both video and texts are mixed together, a web page with a Media Player embedded within it is built.

5.3.1.5.2 Map Simulator

The Map Simulator helps an author visualize how a story unfolds in space and time without actually walking around. To use this Simulator, the author needs to select a starting place. After the first clip is played, the Simulator asks the author to choose the second place where the author thinks that a user might go. As long as the Simulator knows the two places, it will generate a simple animation showing a black dot moving from the first place to the second place. The black dot represents the audience. If there is an available clip at the second place, the clip is played. To simulate more walking paths, the author just repeats these steps.

Use and Constraints: This Simulator can be used for checking several critical story design issues. For example, it allows the author to understand how clips are distributed in space, how clips are related to locations, and how the audience traverses space to watch clips. The Simulator also gives the author a sense of story density. Are two locations too close or too far from each other? Does the audience have too many choices for traveling from point A to point B? How will the audience make a decision on where to go next if a previous clip does not give much explicit information? Through simulation processes, the author can anticipate certain possibilities regarding these questions.

For various reasons, however, there is no way to reconstruct a real mobile experience on a single simulator. For example, the current implementation of the Map Simulator does not take details of real physical situations into account. It cannot simulate walking speeds of the audience. It cannot recognize if there is a wall between two places. Furthermore, the Simulator does not synchronize location detection data with the M-Views server. In other words, the Simulator cannot display where the wireless signal is strong and where it is weak.

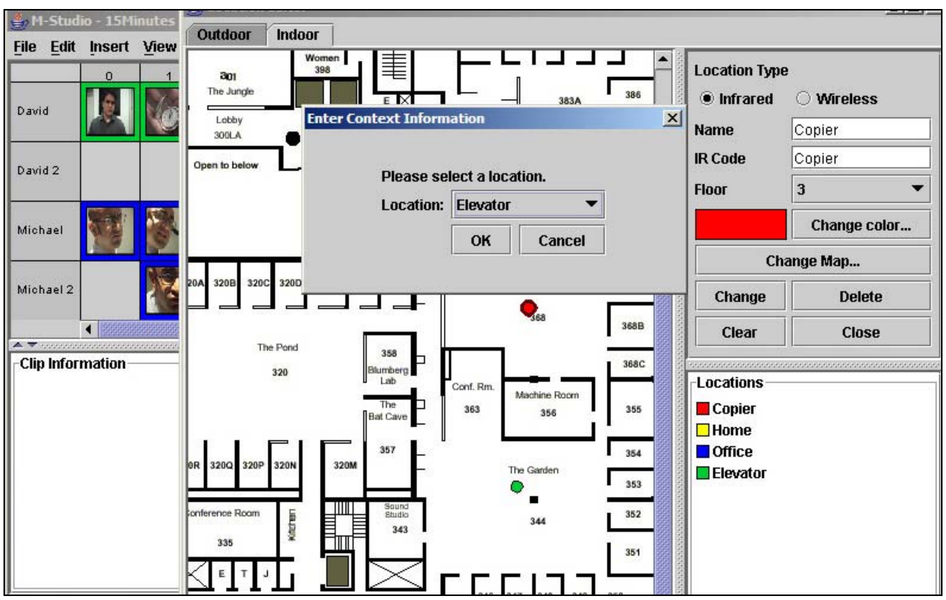


Figure 9 Map simulator

Storyline Generator: Both the Story Simulator and the Map Simulator require an author to either explicitly create a story path or select a specific starting and ending point manually, in order to simulate a story experience. These manual processes are time-consuming. It is onerous for any author to check all possible story lines on either Simulator. To solve this problem, storyline an generator has been created to help the author see all story lines that are possible according to the settings of story flags. Although the storyline generator cannot give the author a complete picture of possible story lines, it indeed allows the author to quickly understand the scope of the possibilities.

5.3.1.5.3 Algorithm

The basic algorithm for generating story lines is straightforward, but it could generate hundreds or even thousands of possible paths, which makes this algorithm impractical to use. The generator offers three constraints: time, location, and context information. By applying these three constraints into simulation processes, an author has more control over possible outcomes from the generator.

Basic Algorithm: When there is no location, time, or context constraint, the generation outcome is solely determined by flags. The generator starts off with an empty flag table and a list of all the clips in the story. By evaluating flags, the generator checks initial conditions to decide which clips could play and to create a path for each possible starting clip. Then, the same steps are called recursively with each of these paths, until there is no further clip to be added. Chart 7 illustrates a generation process example, in which a story has *n* numbers of clips and the default setting is that all clips are not repeatable.

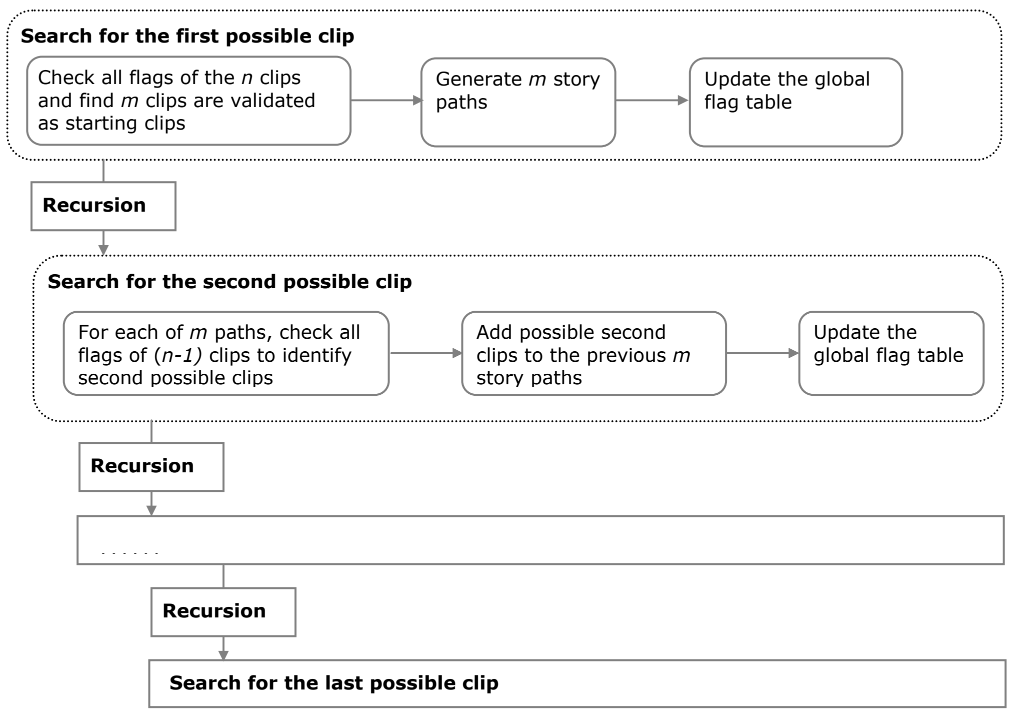


Chart 7 Algorithm for generating story lines

Constrained Algorithm: This algorithm is similar to the basic algorithm, but takes story design constraints into account. In many simulations, the author wants to see the possibilities of story lines under a specific set of circumstances. For example, a particular mobile cinema story always starts at the same location, making story lines with other starting points irrelevant. Hence, the storyline generator allows the author to define a location constraint explicitly.

M-Studio offers three types of constraints: location, time, and context. The three types of constraints can be used individually or collectively. With location constraints, the author is able to specify a complete or partial location path, and find all the possible story lines that could unfold, given this path. The generated story lines correspond to the specifications. For example, the first clip displayed in the story path must be at the first specified location, with the second clip at the second location, and so forth. For those clips that are not applied with any specific constraints, simulation processes are done based on the basic algorithm. For each slot with a location constraint, the generator can rapidly reduce the number of possible story lines. Therefore, the total time of simulation is much faster than with unconstrained simulation. Like location constraint, time constraint can be used to specify time-based story paths. For example, an author can define a starting time slot for the first clip, in order to filter out irrelevant clips with different starting time slots. Context constraints can be used in the same way as location or time constraints.

Generation and Visualization: Depending on the structure of a story, both generation and visualization of simulation could be very complicated. In many cases, a generation may consume extraordinary computer resources; and simulated results may be too chaotic to be useful. To ease generation for the computer and visualization for an author, several techniques have been employed in M-Studio. 1) The simulation data is not maintained in memory entirely. The data only needs to exist in memory when the author wants to interact directly with it. 2) If the author changes a certain clip's flags, a new simulation can be rebuilt from the clip. The generator assumes that this will not affect the ordering of the clips in the path before this clip. Thereby, only a smaller portion of the simulation needs to be recalculated. 3). To speed up creating possible story paths, the generator first eliminates clips that cannot be played, given the last clip in a path. 4) If too many story paths are generated, M-Studio only displays 150 paths at a time and notifies the author that more paths could be generated. 5) Applying location, time, or context constraints when generating possible paths is always advisable. If story clips are repeatable, a path could theoretically be infinite. Applying one or a combination of constraints can make simulation more practical. An example of generated results is shown in Figure 10.

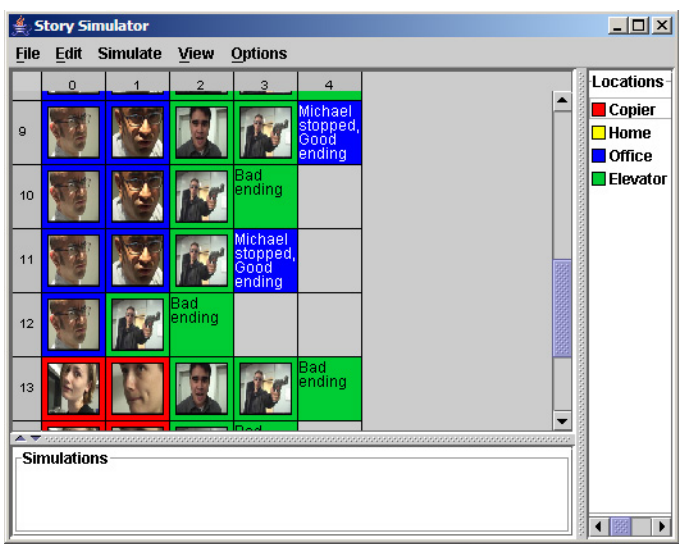


Figure 10 Generated results

Tree Explorer: In addition to the timeline-based interface of the storyline generator, M-Studio also allows an author to explore possible story lines through a tree explorer, shown in Figure 11. Unlike the timeline-based interface, the tree explorer offers an interactive interface, illustrating processes of generating story paths step by step. At the beginning, the generator presents all possible starting clips and lets the author select one clip. The generator then checks all the next possible clips there could be, given that the author made that first choice. Besides the interaction feature, the tree explorer requires far less memory and time to run, compared to the storyline generator.

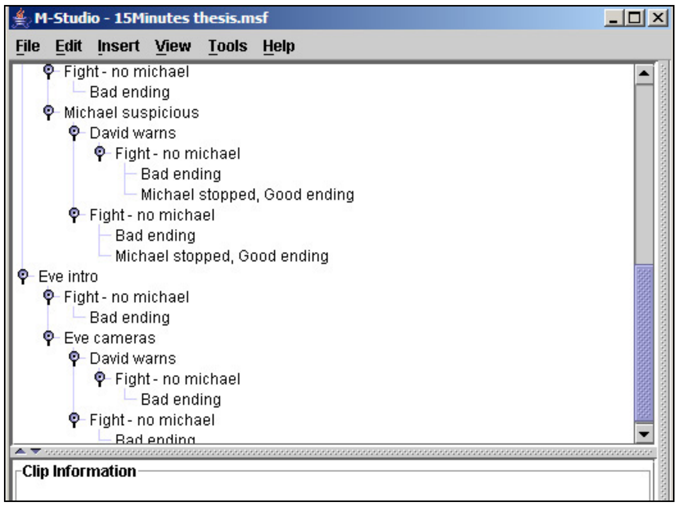


Figure 11 Tree explorer

5.3.1.6 Integration with M-Views Server

The design of integration with M-Views Server follows two principles: 1) Communication between M-Studio and M-Views Server is as simple as possible. An XML schema is designed for communication between M-Studio and the server. 2) Both M-Studio and the server are able to work independently. After a story designer finishes an authoring process, she or he can export XML-based story scripts to the server along with all media content. This design principle allows the designer to focus on authoring processes without knowing any details of how the server works.

Figure 12 shows the general format of the XML schema in M-Views. The M-Views XML story script is a list of possible story events that correspond to a clip (video or other clip) in both M-Studio and the server. The basic structure of an XML file is as follows:

StoryName*:* Each story script is encapsulated in a pair of *StoryName* tags, in order to differentiate this mobile cinema story from other stories on the same server.

ID*:* Each story event has a unique *ID,* shown as *ID1, ID2, ID3,* etc. A pair of ID tags contains all information about the event, including a header tag, requires flag clause, message clause, media URL, results flag clause, and heuristics. A header tag has several attributes that are used to generate the corresponding message for the M-Views client. *Context* field refers to the location of presenting events, and should be recognized by and synchronized with whatever location detection systems are being used, such as GPS, infrared, or 802.11-based MapAgent. The *Repeatable* field determines if a clip can be watched more than once. The *Story* field refers to the title of this particular event.

Requires: This clause consists of a list of flags that must be true in order for the event to play. Each flag has a unique tag and an expression that will be evaluated with respect to the given value in order to determine the state of the flag.

Message Clause and Media URL: *Message* is useful for providing story information that cannot be expressed in video clips. A *Media URL* can refer to a local media file address or a WWW address. The media format should correspond to whatever media players are available on the M-Views client.

Results Clause: This consists of a list of flags that are updated after this clip has been played. Meanwhile, the global flag table will be altered as well.

Heuristics: This can help to decide which clip is the best one among a list of candidates happening in the same location. Each heuristic tag corresponds to a category that describes something about what happens in a clip. Each tag also has a value, which indicates how likely it is that the clip will be ranked at the top of a list.

```
<?xml version="1.0"?>
 <storyName>
        <ID1 Context="location" Date="" From="" Repeatable="false"
           Story="storyName" Subject="clipTitle" To="">
         <Requires>
                <variableName expr="evaluator" value="number"/>
                <OR>
                  <_ expr="?S" value="clip1ID"/>
                  <_ expr="?S" value="clip2ID"/>
          </OR>
                ...
          </Requires>
          <Message>
             Text message to be displayed along with video.
          </Message>
          <MediaURL>
           storyName/filename.mpg
          </MediaURL>
         <Results>
           <variableName oper="operation" value="number"/>
           <listVariable oper="$+" value="string"/>
                ...
         </Results>
         <Heuristics>
           <categoryName value="number"/>
              ...
         </Heuristics>
        </ID1>
        <ID2>
        ...
        </ID2>
        ...
 </storyName>
```

Figure 12 XML schema

In summary, this section describes three key frameworks of M-Studio: Storyboard, Storyflag, and Simulation. These frameworks support the author to create story structures, define presentation schemas, and simulate possible story participation by the audience. In the next section, I describe the M-Views Server and Presenter.

5.3.2 M-Views Server and Presenter

The M-Views Server and Presenter are built on a client-server architecture and connected to each other through broadband wireless networks, such as 802.11 networks. The architecture connects multiple handheld computers, such as iPAQ PDAs, to M-Views Server, using an account/subscription service model. The design of M-Views Presenter is generalized enough that numbers of contextual sensors can be incorporated with its software for particular application requirements. The server is also built by taking scalability and maintainability into account; therefore, one M-Views Server can handle multiple mobile cinema stories simultaneously and most maintaining work can be done through standard web interfaces. The framework of the main components and the information flow of M-Views client-server are illustrated in Chart 8.

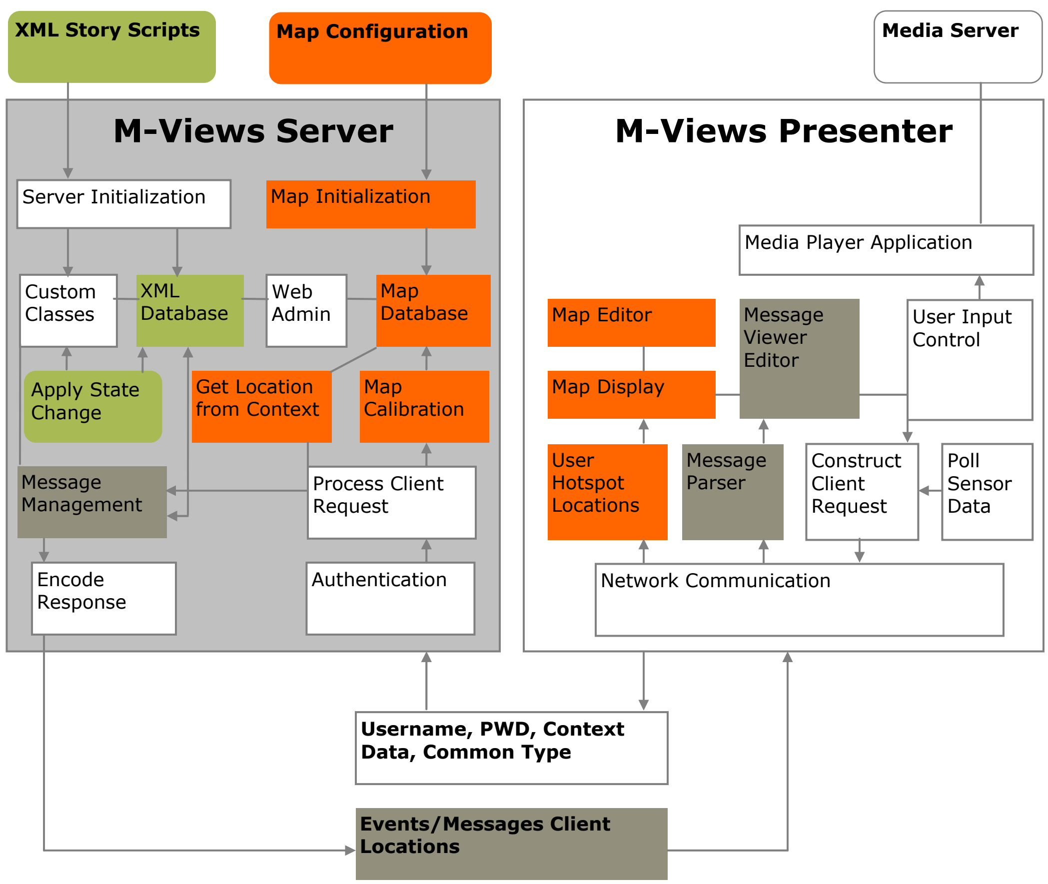


Chart 8 M-Views client-server structure

On both the server and client sides, M-Views system consists of four main categories of components: 1) MapAgent components (shown in orange blocks of the above diagram), 2) a messaging framework (in dark gray blocks), 3) flagging and XML database (in olive green blocks), and 4) administration, communication, and third-party components (in white blocks.)

M-Views MapAgent consists of several modules:

- Map Configuration: This module is connected with M-Studio and allows M-Views Server to import location information and story configurations from M-Studio.
- Map Initialization: This module runs at the beginning of loading server setup and converts map information into the map database on M-Views Server.
- Map Calibration: It basically reads wireless signal strengths and access point addresses from M-Views Presenter and calculates the distance between the user's location and story location to determine whether or not the server should send the client a message. The details of calibration theory and design will be discussed in later sections.
- Map Editor and Display: These two components run on M-Views Presenter. The editor is used by a story author to create story locations; the display can be used to monitor story locations and the audience's mobile behaviors.

The M-Views messaging framework is also based on both M-Views Server and Presenter. On the server side, there is a message management module that is connected to the XML database and responds to the requests from the client. On the client side, this framework consists of a message parser and a message viewer/editor. It is associated with a portable media player application, which is able to communicate with a media server through wireless networks.

The flagging system and XML database are the central components that decide how mobile cinema content is presented. The main algorithm of the flagging system on the server side is similar to the algorithm of Storyflag in M-Studio. An XML database can contain many sets of story scripts.

In order for M-Views to work appropriately, several administration and communication components are needed. For example, a web-based administration interface can help the author to manage, monitor, and modify story content and audience data. Communication requests sent from Presenter to M-Views Server include authentication, context, operation mode, and command information. Requests are parsed, context information is processed, and story scripts are examined to identify new content for the clients. After processing requests, the server responds to each client with event, messaging, and location information.

What are typical authoring processes and how could the audience use the M-Views system? A typical authoring process involves several critical tasks before any story is presented. For example, story hotspots need to be calibrated with the map editor on the M-Views Presenter in order for the server to compare story location information to individual audience location. If there is more than one story existing on the server, a technical administrator needs to configure server initialization and content management in order for multiple stories to co-exist on the same server and network. These tasks often are iterative and need to be tested several times. To participate in a mobile cinema, the audience needs to subscribe to it. The M-Views system supports multiple mobile cinema stories as well, because each story functions independently. Theoretically, a story participant can experience and interact with several mobile stories simultaneously.

5.3.2.1 Design Strategy of M-Views Server and Presenter

The design of M-Views Server and Presenter follows several strategies.

First, the overall design has to be suitable to the current wireless infrastructure. No current wireless infrastructure is perfect for supporting mobile cinema experience. The bandwidth of the current cellular wireless network in the United States is too narrow to transfer video content; the 802.11 standard was designed for local wireless networks and doesn't support moving connections very well. Therefore, the design of M-Views has to make the most of current wireless infrastructure as well as provide its own techniques to eliminate the disadvantages of the infrastructure. For example, the current client supports both streaming video and local caching, so that Presenter can play back video content even if the wireless connection is not so smooth sometimes. The M-Views system is also designed generically enough that it can adapt to new wireless network infrastructures in the foreseeable future.

Second, the architecture of M-Views Server and Presenter is designed to cope with technology uncertainty in numerous ways. For example, the map viewer of M-Views Presenter allows the author to view the overall distribution of the strengths of wireless signals, so that the author can avoid placing mobile cinema stories in the locations where the signal is weak. Furthermore, the map editor provides editing tools for the author to define the attributes of location profiles. Even in a location with low wireless signal, the author can enlarge the area of a hotspot. As a result, MapAgent has a better tolerance for detecting the audience's location.

Third, like M-Studio, both M-Views Server and Presenter take participation uncertainty into account. Unlike M-Studio, which helps with visualizing mobile cinema content and simulating story experience, the Server and Presenter employ a messaging mechanism that is designed to enhance the flexibility of story presentation. The messaging framework can be used not only for early stages of story design, but also for sending necessary information to the audience if she or he misses critical messages.

Finally, the design of the M-Views system needs to deal with several practical concerns, such as scalability issues and costs of maintaining mobile cinema stories. By scalability, I mean that several stories co-exist on the same server. A simple and scalable subscription model has been employed to deal with multiple story scenarios. A high cost of managing mobile cinema content could be a barrier keeping the author away from designing new stories. Hence, most maintenance work for mobile cinema must be done through a simple web-interface, which also decreases the cost of monitoring multiple M-Views systems simultaneously.

In the following sections, I will discuss the details of the four sets of components: MapAgent, M-Views messaging framework, M-Views flagging system and XML database, and administration and communication components.

5.3.2.2 MapAgent

How does MapAgent work? What is the basic theory behind its current implementation? What are the main algorithm and the architecture? The following sections will answer these questions.

5.3.2.2.1 Theory

MapAgent uses wireless network signal strength information, which refers to the Received Signal Strength Indicator (RSSI), to detect and track location in the current version of system implementation. RSSI is associated with each 802.11 access point within range of the client device. The basic hypothesis of using RSSI to detect location is that the stronger signals should yield greater difference scaling. Figure 13 supports this hypothesis—it demonstrates the range of RSSI values for communication between two 802.11 adapters as a function of distance. Measurements were taken in a typical office environment with minimal RF interference, few direct obstacles between adapters (moving bodies only), and without the presence of other wireless networks. The logarithmic relationship and increase in RSSI range with distance support the idea that strong signals give a more accurate indication of position.

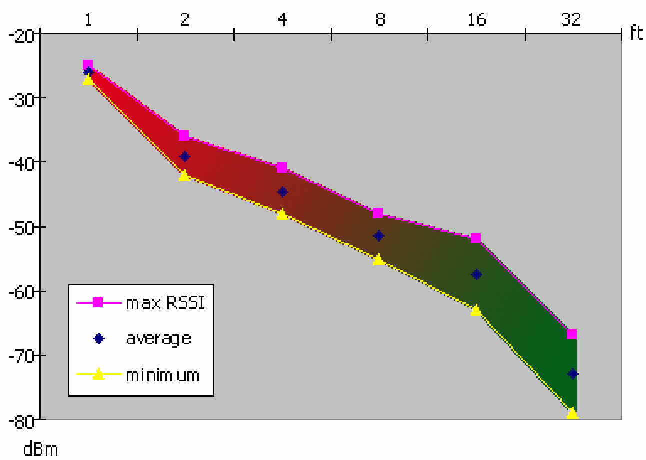


Figure 13 Function of distance

There are two practical issues with using this theory to design MapAgent.

- Physical environment is critical to the value of RSSI; and different environments certainly affect both the creation and the detection of location profiles based on this theory. Furthermore, some physical environments may be re-arranged over time. These kinds of issues inevitably change the previous recorded profiles of the locations and cause problems for presenting mobile cinema. Therefore, the design of actual algorithm and system implementation must offer flexible means for the story author to deal with not only signal strengths, but also physical environments.
- Both the author and the audience could move around in physical space while creating or receiving mobile cinema stories; therefore signal strengths may change rapidly. The speed of either the author or the audience needs to be considered in order to employ this theory.

5.3.2.2.2 Algorithms

MapAgent requires that a map can be calibrated using M-Views Presenter, with which location hotspots are defined. There are two purposes for defining hotspots: 1) creating hotspots can ensure that all mobile stories work at their locations; and 2) so that the threshold for matching a story to a physical location is appropriate.

Each hotspot consists of both the *two-dimensional coordinates* of the measurement and a *tolerance radius*, and is associated with the RSSI data measured at the time of creation. The radius defines the spatial threshold of the hotspot. Using circular hotspots instead of point-based measurements also allows for easier calibration than other comparable location detection systems (Bahl and Padmanabhan 2000 and Capkun *et al.* 2002). This is because calibrated measurements are not simply averaged together to estimate position; instead, the system attempts to build a distance metric for use when performing triangulation and trajectory analysis. A table of access point MAC addresses and associated RSSI values is used to describe the location in signal space (Table 7). Global variables are defined for use in all signal-to-location estimations performed with the given map.

Global Map Variables		**Location Hotspot RSSI Data**			
scale:	70.0	[1] MAC:	00601D1D215C	RSSI:	-88
relevance:	0.9	[2] MAC:	00601D1D2151	RSSI:	-88
neighbors:	3	[3] MAC:	00601D1D2162	RSSI:	-44
gravity:	0.05	[4] MAC:	00601D1D2176	RSSI:	-89
penalty:	0.0	[5] MAC:	00601D1D2111	RSSI:	-68
max signal:	-10	[6] MAC:	00601D1D2167	RSSI:	-80
min signal:	-120	[7] MAC:	00601D1D217E	RSSI:	-82
max idle time:	10000	[8] MAC:	00022D29286D	RSSI:	-85
max speed	50	[9] MAC:	00601D1D2163	RSSI:	-87

Table 7 Location profile in signal space

Hashtable of MAC Addresses			**Hotspot 1**	
MAC 1	Hotspot 1		1	
	Hotspot 2		2	
MAC 2	Hotspot 1		3	
MAC 3	Hotspot 2			
MAC 4	Hotspot 1		**Hotspot 2**	
	Hotspot 2		1	
			3	

Table 8 Hashtable that maps each access point MAC address to a list of hotspots

As new hotspots are defined on Presenter, MapAgent builds a hashtable that maps each access point MAC address to a list of hotspots with which the MAC address is associated on M-Views Server (Table 8). This optimizes performance during the search to match client RSSI data with known hotspots when calculating a client's location. Hotspots cannot overlap each other, and should not overlap with areas of the map that are impassable (thick walls, large immovable furniture, etc.).

After a story map has been created, MapAgent can detect an audience's location by calculating distance measurements based on the RSSI of her M-Views Presenter. The detection consists of two steps.

First step: A distance measurement (weighted summation of RSSI differences) is calculated for each possible hotspot that M-Views Presenter might currently occupy (those that were previously defined with measurements for access points that M-Views Presenter now sees). These distances span approximately from the center of each hotspot to the current position of Presenter. The calibration of the distance is done according to the following difference calculation (D_{ab}):

$$\sum_{n} C \cdot \left| a_n R^{|a_n - s_{max}|} - b_n R^{|b_n - s_{max}|} \right| + \sum_{m} P \cdot (a_m - s_{min})$$

D_{ab} = estimated physical distance between client ***a*** and center of hotspot ***b***

a_n = current RSSI value between client and AP MAC address ***n***

b_n = recorded RSSI value between hotspot and AP MAC address ***n***

a_m = current RSSI value between client and *unknown* AP MAC address ***m***

s_{max} = maximum possible value for RSSI (i.e. –10)

s_{min} = minimum possible value for RSSI (i.e. –100)

C = scale variable, empirically determined for each map/environment

R = relevance variable, empirically determined for each map/environment

P = penalty variable, used to penalize *all* D_{ab} when encountering any a_m
(often not used—set to zero)

Note that for ***n*** in ***b*** that are *not* observed in the client measurement ***a***, a_n is assumed to be s_{min}. In contrast, for any ***n*** in ***a*** that are *unknown* to the MAC address hashtable, such RSSI values a_m are only used in penalty summation.

Equation 1 Distance calibration

Second step: According to the above difference calculation, MapAgent can detect whether or not a Presenter is within a hotspot. By finding the lowest D_{ab} such that the difference calculation is also less than the given tolerance radius of the associated hotspot ***A***, the Presenter's location can be estimated to be inside of location ***A***. Using a combination of trajectory tracking and triangulation between ***A*** and the two nearest neighbor hotspots (those with second and third lowest D_{ab}), the client's location is calibrated (Figure 14).

The reason for employing a triangulation method is not only because it is flexible and easy to calculate, but also because it is able to deal with signal noise, which is inevitable in real buildings. Furthermore, triangulations can improve localization by looking at additional neighboring hotspots (hotspot gravitation).

The reason of using trajectory tracking is two-fold. First, trajectory tracking can deal with offset errors or inaccuracies with triangulation. Second, it can be used to calibrate the speed of the audience by comparing prior and current location coordinates, given a time interval. If the speed is greater than the maximum possible speed, the triangulated coordinates are found to be invalid. In the event of triangulation failure, the algorithm will then estimate the current position of the client based solely on the current trajectory, if available. If both the prior trajectory and coordinates are unavailable, then the system infers that the user has moved outside the calibrated map area and is untraceable.

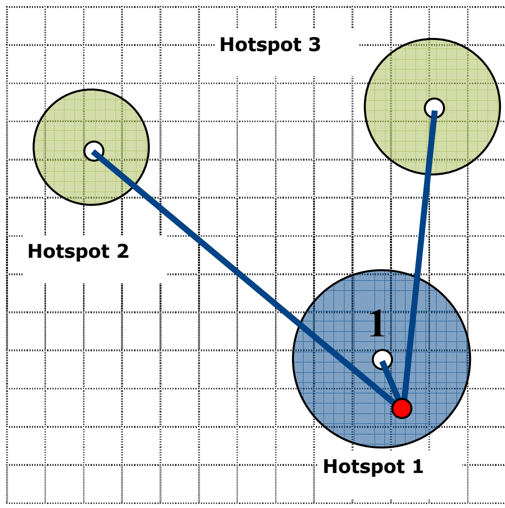


Figure 14 Combination of trajectory tracking and triangulation

Chart 9 illustrates the top-level implementation of the MapAgent algorithm. Each calibration starts from the same question: *Is there a neighbor hotspot that contains the client's location?* If the answer is yes, this algorithm uses a combination of triangulation and trajectory methods to estimate the location of the client.

The client of MapAgent is written in Microsoft eMbedded Visual C++ and it provides an author with interfaces to create story hotspots as shown in Figure 15. The interface on the left is a creation and modification mode, in which the author can use a stylus to draw circles for defining/modifying hotspots. The interface on the right is a following mode, with which the author can see her current location for testing the accuracy of location detection.

The server for MapAgent is written in Java. Each hotspot's data are stored in an internal vector and indexed in a hashtable. This data structure allows for fast MAC-to-hotspot searches when analyzing client RSSI. During the initialization period, the MapAgent loads all map variables from a single properties file. MapAgent also maintains data structures to keep track of each Presenter currently appearing on the map. The Presenter is visible as long as it is communicating with MapAgent (which must also be able to provide a position). If the Presenter connection drops or the location becomes incalculable, the Presenter will be regarded as idle. Idle clients are removed from the tracking list after a specified amount of inactivity.

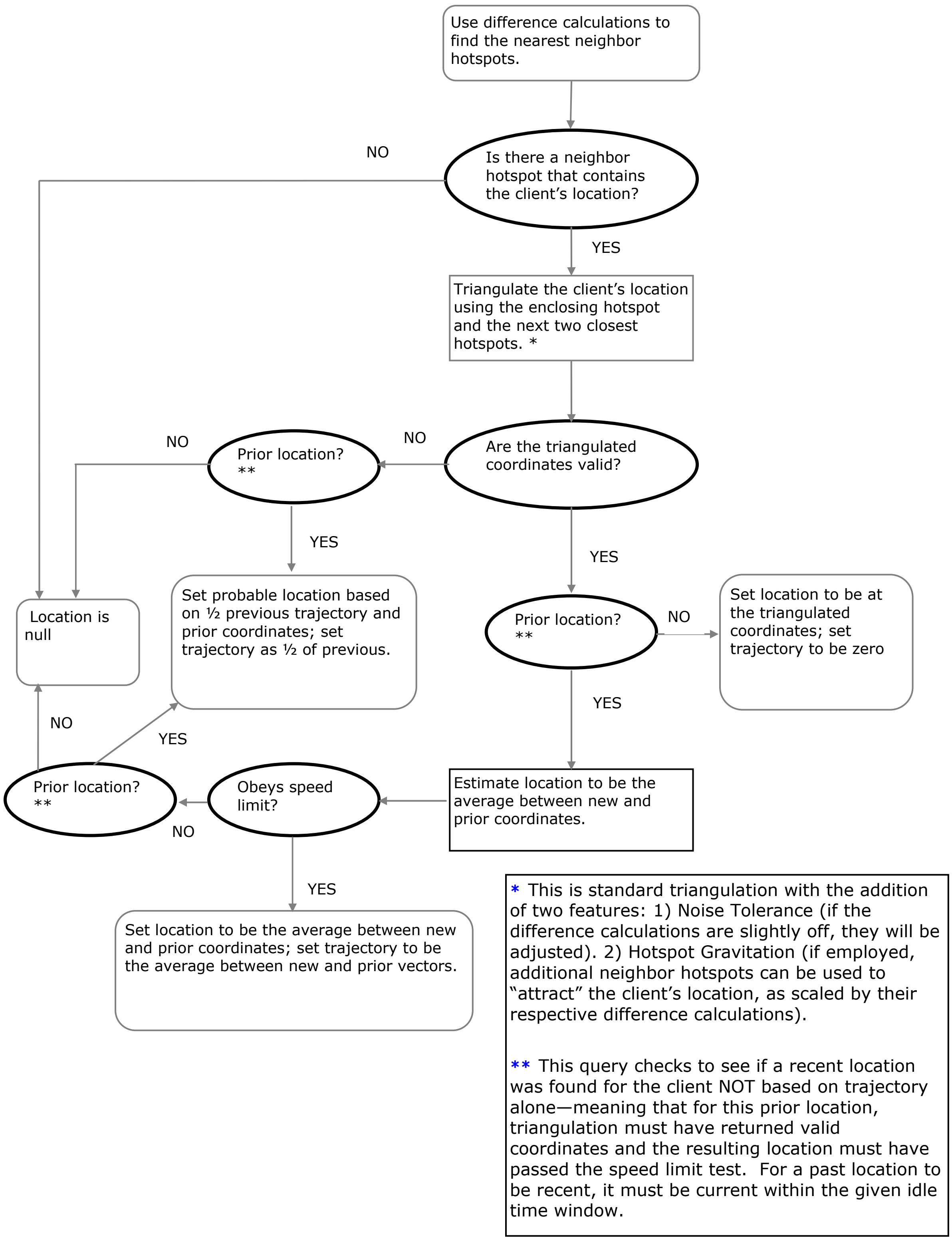


Chart 9 Implementation of the MapAgent algorithm

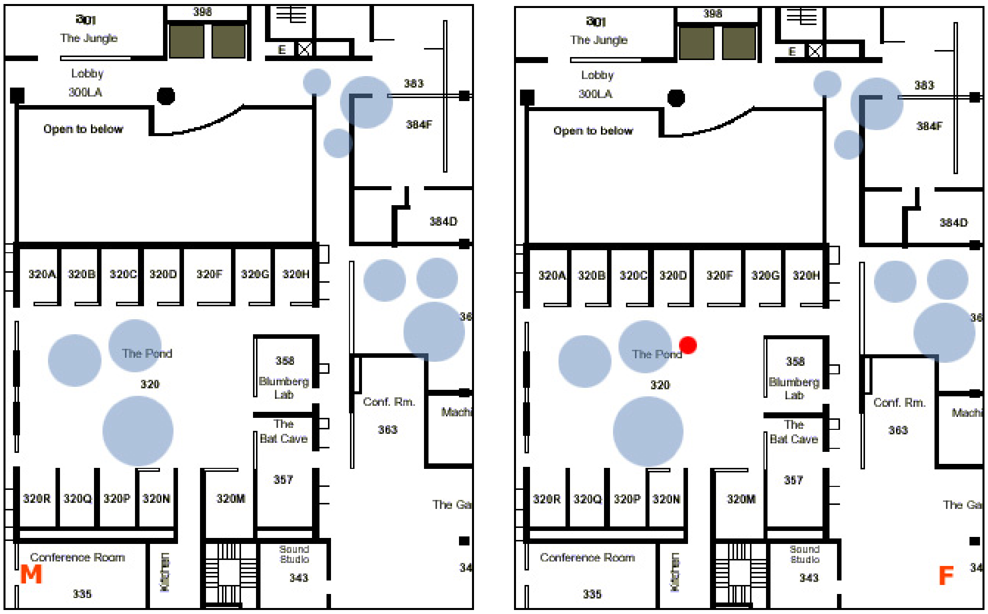


Figure 15 MapAgent client

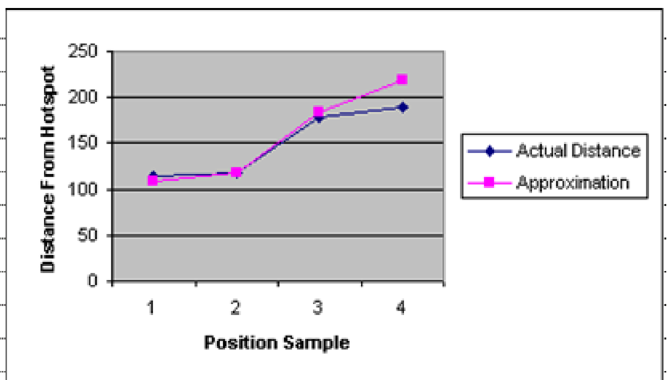
5.3.2.2.3 MapAgent Evaluation

A preliminary MapAgent evaluation, which is shown in Figure 16, has been done in typical office settings (with 6-8 access points evenly positioned and visible in all calibrated areas). The following table illustrates the result of this evaluation, which took place in the building of the Media Lab Europe. The accuracy varied from 1-3 meters.

Due to the uniqueness of signal propagation in any environment, special care must be taken to calibrate hotspots and global map variables. The size and density of hotspots can be used for maximizing accuracy vs. maximizing detection likelihood. For example, if measurement of the RSSI value of ***A*** = -74 and measurement the RSSI value of ***B*** = -75, that is not a significant difference and is probably due to RF of fluctuations or noise. On the other hand, if ***A*** = -74 and ***B*** = -78, the difference is more likely due to a change in client position. The RSSI measurements sent from the client to the server are also averaged over a specified time window (the default is 6 seconds), which also limits the effect of noise.

In summary, the precision and performance of creating/detecting hotspots depend entirely on the needs of the mobile application that uses MapAgent. The following sections describe the other components of the M-Views Server and Presenter.

Evaluation of Difference Calculation Vs. Distance from Hotspot					
Testing Environment: Media Lab Europe, Floor 02					
ACCESS POINT MAC	HS RSSI	POS #1 RSSI	POS #2 RSSI	POS #3 RSSI	POS #4 RSSI
0002B3BAEBCC	-65	-60	-59	-57	-60
0002B3BA7063	-72	-74	-73	-81	-85
0002B3C386A9	-89	-90	-90	-84	-90
0002B3BAE464	-67	-60	-60	-61	-57
X COORD (px)	504	395	431	462	325
Y COORD (px)	90	91	183	269	215
RELEVANCE VARIABLE	0.94	0.94	0.94	0.94	0.94
SCALING VARIABLE	77	77	77	77	77
MAX SIGNAL VALUE	-10	-10	-10	-10	-10
CONTRIBUTION FROM AP #1		42.91960189	52.57408217	73.02980011	42.91960189
CONTRIBUTION FROM AP #2		10.9864042	5.614800742	42.50540548	56.4369983
CONTRIBUTION FROM AP #3		2.552958976	2.552958976	14.77025043	2.552958976
CONTRIBUTION FROM AP #4		57.77419648	57.77419648	48.48955706	87.8843947
APPROXIMATION FROM HS (px)	**0**	**114.2331615**	**118.5160384**	**178.7950131**	**189.7939539**
ACTUAL DISTANCE FROM HS (px)	**0**	**109.0045871**	**118.2285921**	**183.8613608**	**218.3254452**



NOTE: Due to building architecture, noise, map resolution, or inaccuracies in calibration, approximations can vary greatly. Due to this, the triangulation process can stretch or shrink distances to make poor estimates more accurate. Combining this with trajectory monitoring, sufficient access points, and careful calibration, the algorithm consistently yields good location estimates.

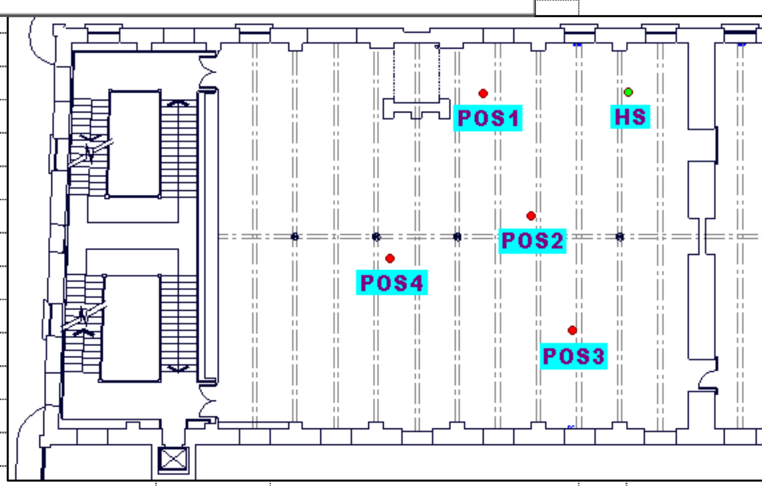


Figure 16 Evaluation of MapAgent

5.3.2.2.4 XML Database and Flagging System

MapAgent is connected to an XML database on the M-Views server, which shares the same XML schema and flagging mechanism with M-Studio. As discussed before, M-Studio is able to generate XML story scripts and send them to M-Server. On the server side, the profile of each M-Views user account contains specific variables for every subscribed story script. These variables are referred to as M-Views flags. For an M-Views Presenter to receive a story event, flag requirements for that event must be met (in addition to matching context and other conditions). The variables in the Presenter's user profile are examined using Boolean logic statements defined in the story script. Similar to the flagging system in M-Studio, all requirement flags must be true in order for a particular clip to be played. After the audience views the clip, the global flags' values are updated according to the result flags. In addition to the flagging system, M-Views Server employs the heuristic engine to enhance the user experience. After receiving an event, any associated heuristic information is averaged into the heuristics of the audience profile. For example, if an event contains content of a romantic nature, experiencing it could make the audience more prone to receiving romantic events in the future. Some applications use this functionality to track user preferences.

5.3.2.3 Messaging System

If a story location is detected, the flagging system calibrates flag values in order to decide how to present mobile cinema content through a *generalized messaging framework*. This framework is designed specifically to support mobile cinema. Under this framework, all messages—whether they are client-to-client instant messages or events encountered in a location-based story—are processed using the same computational mechanism. Messages can be generated through two modes: a server mode (Figure 17) and a client mode (Figure 18). The former mode supports all messages and events sent by the server. It means that messages and events are stored in XML story scripts that have been created by story authors. Some messages are associated with video clips, so that these messages can launch a PocketTV media player in order to show video content. Some messages play critical roles in unfolding stories, such as telling the audience to go to a certain location. Other messages can be background information for the audience to understand story context. The client mode offers the audience a means to communicate with each other. In several story brainstorming sessions, many story designers wanted to promote communication among the audience members in order to increase sociability. The client mode supports both one-to-one and one-to-many messaging modes for the audience to share their ideas, thoughts, and impressions of mobile cinema stories. In addition, the client messaging mode is designed to support future mobile games as well, so that multiple players can negotiate game strategies through the general message forum (to which all game players are subscribed and where client-to-client messages are created).

Furthermore, all messages, even those created through the client mode, can be made context-dependent and can have associated media URLs. These features, coupled with familiar functionality (i.e., message forwarding and group mailing), allow for an intuitive, robust, context-aware messaging experience.

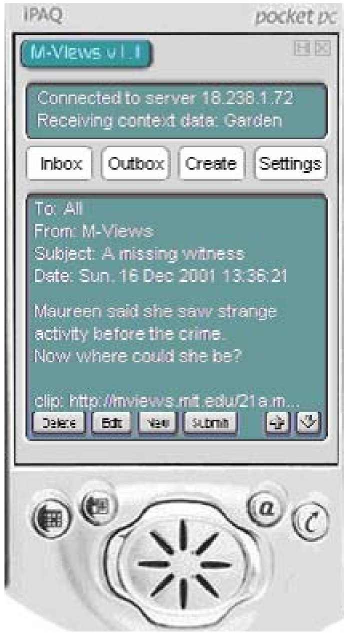


Figure 17 Server mode

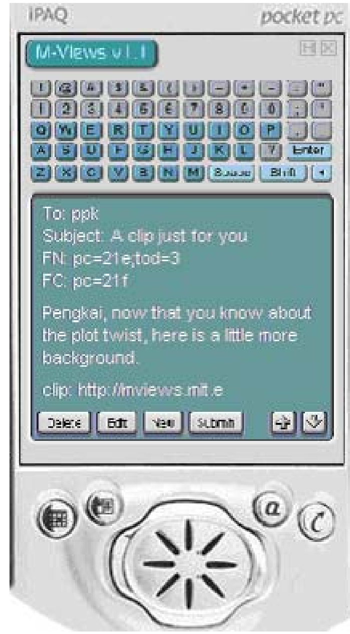


Figure 18 Client mode

5.3.2.4 Implementation and Web Administration

In this section, I first discuss an overview of the M-Views system implementation and web administration. Then, I offer several reasons for implementing and managing the system in the current way.

The server is written in Java, runs in Apache Tomcat or Java Servlet Container, and consists of a servlet, a utility class with numerous static methods, a specialized exception class, and the default MapAgent. The servlet uses HTTP POST to communicate with the client and HTTP GET for administration, account creation, and maintenance. The servlet is deployed in Apache Tomcat on the M-Views Server, which stores user profiles, forum messages, and story scripts as XML documents. These components are loaded into memory when the servlet is deployed. The RAM-resident data scheme allows for fast manipulation of the XML data via the Apache Xerces package.

Each user profile includes the audience's information, such as username, password, email address, current context, current IP address, and latest connection time. The profiles also include the lists of subscriptions, maps, and team affiliations. Every subscribed story is associated with an entry in the profile that holds current story flag values, heuristics, and a list of experienced events with timestamps. Each user profile file also contains a list of current inbox and outbox messages.

A story script contains a collection of messages (events). Each message includes an ID, required context (coordinates, infrared signatures, etc.), sender username, recipient username(s), subject, and message-repeatable indicator. These messages also contain lists of flag requirements, flag updates, and heuristics that describe the content of these messages. Finally, each message has a text body and an associated content URL (pointing to a video file over HTTP, for example).

The M-Views client is written in eMbedded Visual C++ and runs on the Windows Pocket PC operating system. Parsed messages from the server are displayed and edited in the Presenter's GUI, while sensor data is polled and background threads perform network communication during the update cycle. The program takes advantage of the Windows Pocket PC Game API to provide a customized interface. In addition to the message editor interface, the client also features a map viewer/editor tool, as a part of MapAgent components. This allows users to view their server-calculated positions (plus the locations of other users) and also allows administrators to calibrate map coordinates using the standard client. In addition to the current wireless signal-based location detection techniques, MapAgent also supports IRDA to recognize infrared beacons for additional context detection (thereby allowing users to interact with objects). In a previous version of M-Views system, GPS receivers were used for detecting outdoor locations, but they are not implemented in the current version of MapAgent. A separate program, *M-Views Configuration*, has been created for configuring the client password, map, network settings, and helper applications. Helper applications are used for media playback and to assist in context-awareness.

A Web-based administration interface, shown as Figure 19, provides access to all server-maintained XML documents. It provides the statistics information of all stories and audience activities of message delivery through log files. The interface consists of two main features: monitoring the XML database and managing story maps. In order to help the author to understand how the audience interacts with a story, the interface provides the real-time information of the audience's interaction with a particular story through *User_State*. The author can also see all messages that are created by the audience members through *User_Forum.* Furthermore, all XML-based story scripts of active mobile cinema stories are displayed in the same interface. The second main feature, managing story maps, is also listed in the main administration interface. This tool allows the author to track the locations of users in real-time; therefore, the author can know the pattern of walking paths generated by the audience. The author may also edit map variables and calibration data via this same interface.

There are several reasons for current approaches to implementing the M-Views Server and Presenter.

- Open source packages, such as Linux and Apache Tomcat, are an emerging infrastructure for developing web applications. Because they are open source software, we can customize them with good control of how the system runs. This open source software has been approved to be scalable enough to host large-scale enterprise solutions. If the M-Views system is adopted by many people in the future, the current implementation offers a great base for further development.
- Both HTTP POST and GET methods are used, so that the M-Views Server runs on any Web-standard infrastructure without considering the underlying layers. The current wireless standard is 802.11b, which has been evolving since its birth. Our web-based approach allows M-Views system to be compatible with any new standard in the foreseeable future.
- Video content still demands too much from any portable wireless devices in terms of computational capability and power consummation. Furthermore, the small size of form factor of any wireless device cannot offer a pleasurable human-computer interaction

experience. Both eMbedded Visual C++ and Pocket PC Game API could offer reasonable means for us to develop a client to partially solve these two problems.

- Lastly, given the very limited resources we had and the many technology uncertainties, the current approaches to implementing the M-Views system allowed us to develop a working and relatively reliable platform for creating different genre and scale mobile cinema content in the past three years.

Figure 19 Web-based administration

5.4 Summary

In this chapter, I have first described the prototype of M-Views system 0.1 and a mobile cinema production, *Another Alice.* Both the system development and story creation have provided me with tangible data that not only helped address critical issues, but also reflected major technology uncertainty and participation uncertainty in system developments. Based on these considerations, I have then proposed three computational approaches to designing the M-Views system version 0.2, which consists of M-Studio, M-Views Server, and M-Views Presenter. For each component, I have articulated design strategy, algorithms, and implementation details. In the next chapter, I discuss two mobile cinema productions that have been created based on the M-Views system 0.2.

6.0 PRODUCTION AND REFLECTION

In this chapter, I first discuss *MIT in Pocket,* a hybrid mobile cinema story mixing fictional clips with documentary stories to tell a day of the lives of four MIT undergraduate students. In accordance with evaluation from *MIT in Pocket,* several different story development approaches have been employed to create *15 Minutes,* which is a compact, reliable, and fun fictional mobile cinema story. More than fifty people have participated in this mobile cinema story at several locations, such as UbiComp 2003 and the MIT Media Lab. At the end of this chapter, I discuss three design principles and reflect on what I have learned from the development of two versions of M-Views system and the three story productions.

6.1 *MIT in Pocket*

MIT in Pocket was created in conjunction with the M-Views system 0.2. It aims to reflect a day in the lives of four MIT students and features four main narrative threads told in over fifty fictional soap-opera-style scenes. A few documentary scenes are interwoven into this larger multithreaded narrative about life on the MIT campus. The characters are a variety of MIT students, professors, tourists, and staff members who interact with each other over the course of a single day. Given the broad range of characters, many of whom acted for only a short scene, *MIT in Pocket* provides a more casual viewing experience than *Another Alice.* The viewer can walk around the MIT campus to receive location-based messages, which are associated with the video clips of *MIT in Pocket.* After receiving these messages, the viewer can follow one individual story thread, or sample all of them, to see how all the main characters meet and interact. Two viewing scenarios of *MIT in Pocket* are in Table 9.

Scenario One		
Time	**Location**	**Scene**
10:00 AM	Lobby Ten	A student rushes to a morning class in the nearby lecture hall.
10:15 AM	Lobby Seven	A girl is talking to another MIT student to set up a blind date.
11:00 AM	Student Center	A *Tech* editor gives a memory disc to another reporter and asks him to write an article to cover a piece of hacking news at MIT.
Scenario Two		
10:05 AM	Killian Court	A tourist asks an MIT jogger to take a photo for him.
11:30 AM	Building 56	A girl is checking the MIT website and finds a piece of breaking news about the hacking happening at the Stata Center.
1:00 PM	Building 4	A student puts his hands on the Eastman Statue to pray for a good test result.

Table 9 Two viewing scenarios of *MIT in Pocket*

In following sections, I first describe this story production, such as background information, the production goal, story structures, production processes, story characters, locations, and production evolutions; then I outline the findings from preliminary evaluations.

6.1.1 Background

After the evaluation of *Another Alice* was finished in fall of 2001, my supervisor, Professor Davenport, and I decided to hold an IAP[56] 2002 workshop for creating alternative types of mobile cinema and improving the M-Views system 0.1. About ten MIT students joined this workshop and Professor Davenport introduced students to her interactive narrative development methods, which she developed with creative and practical input from Janet Sonenberg, the Major/Minor advisor for Theater Arts and an associate professor at MIT, and other colleagues Jeff Cleaverly of the BBC. These methods consist of two main steps. The first step is that each student creates his or her own character. What is the profile of a character? What are his or her short-term goals, as well as long-term goals? For example, a student may create his character, Peter, who is a twenty-two-year-old computer programmer and wants to apply to an MIT graduate school. Every weekday morning around 8:00 a.m., he is waiting for the T at Downtown Crossing. Another student may create her character, Amy, a twenty-year-old student at Mass Art, looking for a design job. On the morning of May 3rd, she is also waiting for the T at Downtown Crossing and trying to find a computer design center for packaging her design profiles. The second step is that people think about various possibilities of how different characters might meet. Do two characters have similar interests? Is there any conflict between the two people? For example, at Downtown Crossing, Peter is attracted to Amy and wants to help Amy. However, Amy is not interested in dating Peter, although she would like to have Peter's help. In this case, these two people may continue discussions, but with different intentions.

These story development methods were learned quickly by the students and various characters were created. For example, Rich created a character who is a journalist and interviews homeless people in Harvard Square. Ali had her character, who loses a wallet near a flower store; and Deb's character happens to pick up the wallet and finds multiple ID cards in the mystery wallet. At the end of this workshop, we filmed about fifteen scenes at Harvard Square, where multiple characters interacted with each other. Although the M-Views system 0.1 had technical problems associated with the wireless network at Harvard Square and we were unable to test these clips on PDAs at that moment, we did learn these interactive narrative development methods and knew that these methods were practical, easy-to-learn, and useful for mobile cinema productions.

In May 2002, we started to develop the second version of M-Views system, because (1) the first version was so limited for coping with both technology and participation uncertainty, and (2) the IAP 2002 workshop required various features that the M-Views system 0.1 was unable to support. For example, the authoring processes for dealing with multiple characters became more complicated than the authoring processes (manually coded) that we used for *Another Alice.* To help develop the M-Views system 0.2, I wanted to initiate a new mobile cinema production, which would provide us a tangible story model for developing the improved M-Views system. Professor Davenport was very supportive of this idea and allowed me to recruit motivated undergraduate students who wanted to create mobile cinema. After several rounds of interviews, four people joined us: Chris Toepel '00, Debora Lui '03, Lilly Kam '04, and David Crow '03. Both Lui and Crow were involved in MIT Theater and doing acting and music in several productions; and both Toepel and Kam were MIT tour guides.

[56] http://web.mit.edu/iap/

In June 2002 there were many brainstorming sessions, in which the team tried to understand what mobile cinema was and how it could change story production and user experience. As she did for the 2002 IAP workshop, Professor Davenport taught this group the interactive narrative development methods and asked each student to imagine scenarios for creating their characters. The whole team pre-planned and wrote many possible ideas and coordinated with people who could act in the production and participate on the crew. There were many meetings where they didn't really do any productive work, mostly with everybody there. After many sessions of imagination, brainstorming, and discussion, in the middle of July we made several production decisions collectively.

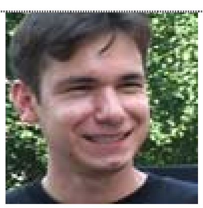
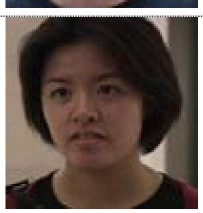
- The main content of the mobile cinema story is to reflect the spirit of MIT, which has numerous legendary figures, cutting-edge inventions, and diverse students. We named this production *MIT in Pocket*, which indicates a cinematic story that can be carried in a pocket. We discussed two approaches to reflecting MIT. The first approach was to create fictional characters by using Professor Davenport's methods. The fictional stories form a narrative about quirky members of the MIT community (including students, professors, and mysterious characters) whose lives intertwine throughout the day, resulting in chance encounters and plot twists. The second approach was to make documentary-style stories, which showcase real events on the MIT campus, such as a robotic design competition in the Johnson Athletic Center or a ballroom dance competition at Walker Memorial. These are meant to give the viewer a sense of the energy and spirit of the physical space they are situated in. For creating the whole *MIT in Pocket,* our goal was to mix the fictional stories with the documentary-style stories, so that the audience would be able to witness fictional stories as well as virtually "meet" legendary MIT figures in a relatively dense story web.
- The potential audience of *MIT in Pocket* is MIT visitors, such as prospective students, families, tourists, and corporation sponsors. These people are usually interested in having a tour of the MIT campus, so MIT and its various famous labs offer tour programs. In many cases, these tourists have their own agendas and constraints. For example, they may only stay at MIT for a few hours. Taking these constraints into account, the production team wanted to give the potential audience more participation freedom. Unlike *Another Alice,* in which the audience had to start from a fixed location, *MIT in Pocket* can be started from and ended at any of the fifteen story locations.
- The story database of *MIT in Pocket* is extensible. Inspired by *Evolving Documentary* [57], we wanted to create a story form that was extensible for two reasons. First, mobile cinema story experiences required a certain density of video clips. If there were only one video clip in an entire building, the audience member might get bored. In summer 2002, the whole production team only had five people and we thought that we were unable to make many

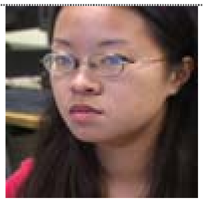
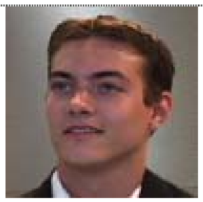
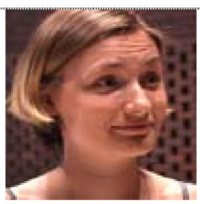
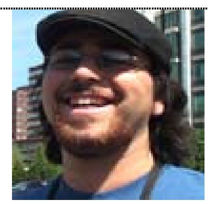
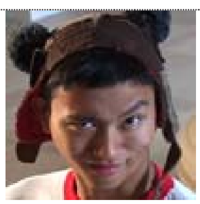
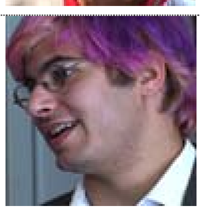
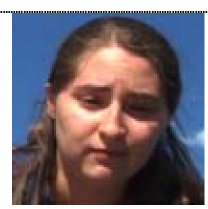
[57] Evolving Documentary was originally coined by Professor Davenport and Michael Murtaugh (1995) in an interactive narrative system, ConText. The primary idea of this form of narrative is to separate content, description, and presentation into interconnected pieces and redefine the relationship between the story, the viewer, and the author. "For the viewer, repetition and revisitation of the story experience is encouraged and no constraints are placed on the duration of a session. For the author, the tasks of content gathering and sequencing take on new dimensions because the content base is extensible, and the author is separated programmatically from the exponentially complex task of explicitly sequencing the material for each viewer visitation" (Davenport and Murtaugh 1995). This novel method has been used for developing several interactive documentary stories, such as *Jerome B. Wiesner, 1915-1994: A Random Walk through the 20th Century*.

video clips for fifteen buildings during one summer. Therefore, an extensible story database would allow us to add more content continuously. Second, for growing *MIT in Pocket*, we wanted other MIT community members to contribute their story clips, such as lectures, social activities, and ceremonies, into this story database. In addition, MIT has collected thousands of hours of video content and these collections of video content could be digitized and uploaded to our database.

After we made these production decisions, the production team started to portray characters themselves. For example, Chris Toepel jogged every day and did a lot of stuff by routine. As an MIT tour guide, he knew a lot of facts about MIT. Based on his personal experience, he created a character, Anthony. Debora Lui was an architecture major and her primary role in the movie was to be a character, Betty, who is also an architecture student at MIT, and goes around sketching buildings on the campus for most of the day. The first character that they filmed was Anthony, who was jogging around MIT and met various visitors. After reviewing the edited clips, every team member confirmed that the self-portraying approach would work. Then there was a full month dedicated to storyboarding in the summer of 2002. They spent all their time in front of the whiteboard drawing little South Park-looking figures that would give them an idea of how they wanted each shot to look. Film production started at the beginning of August 2002.

By the end of summer 2002, four major characters in *MIT in Pocket* had been created. There is Claudia, played by Lilly Kam, who is absent-minded and flighty and tends to miss deadlines, oversleep, and procrastinate. She is friends with two characters, Anthony and Betty, who do not know each other, so Claudia decides to set them up on a blind date. Anthony is very organized and plans everything out. He is the opposite of Claudia. Betty spends much time on sketching in front of several distinctive buildings at MIT. She also talks to Dean Mitchell at the beginning about architecture and, for her blind date, she misses Anthony because he changes the location of the meeting and Claudia forgets to tell her. So Betty and Anthony don't actually meet on the blind date until the very end. Finally, there is Damien, who is having problems with school and wants to be a musician and has a UROP at the Media Lab. Aside from these four major characters, there are other characters as well, such as hackers, eccentric musicians, artists, and faculty members. All characters are described in Figure 20.

Leading Characters of *MIT in Pocket*	
	Anthony (Chris Toepel 00) is a junior who is studying Aero/Astro Engineering. He likes to take a daily jog every morning around campus, where he constantly runs into tourists asking for directions. He doesn't mind, though, because he also works as a tour guide during his free time. He is single and his friend Claudia is constantly trying to set him up with her friend, Betty.
	Betty (Debora Lui 03) is a shy Architecture student who is obsessed with the new buildings around campus. Her big project is due soon, but she can never seem to find a place where she can work on her project and sketch in peace. She likes to play piano in her spare time, but is very private about her music. In an attempt to loosen Betty up, her good friend Claudia sets her up on a blind date.

	Claudia (Lilly Kam 04) is an energetic student who is always in a hurry. Although she is late to her big physics test and has a paper due, she still manages to find time to set up her two good friends, Anthony and Betty, on a blind date. She has other things on her mind as well, such as the mysterious email she received from a student claiming to have found her father's class ring.		
	Damien (David Crow 03) is a UROP student at MIT's Media Lab. He is continually busy with either his school or UROP work, but he'd rather be thinking of other things, like his music. He just bombed a big physics test and is very stressed out. To relax, he goes to the student center to play guitar with some friends. He has other business; sometimes he submits anonymous editorials to the school newspaper.		
Supporting Characters of *MIT in Pocket*			
	Roger (Tim Sutherland 04) is a totally unpredictable MIT student. Whether he's working on something with his friend Franz or randomly zephyring people on Athena, no one can ever tell what he is going to do next.		Harvey (Alan Brody F) is a tourist visiting MIT. He likes to examine everything thoroughly with his binoculars, as well as photograph the campus with all of his numerous cameras.
	Gloria (Welkin Pope G) is a visiting Architecture student from Harvard who just wants to see the MIT Chapel.		Ethan (Dan Bersak 03) is usually busy snapping pictures for the *Tech*. When he's not in the field, you can almost always find him at the *Tech* office.
	Ngotorious (Dave Ngo 02) doesn't let anyone get in his way. Especially when he wants to play the guitar.		Dean Mitchell (Bill Mitchell F) has a lot to say.
	Fred (Dan Katz 03) or Franz, as he likes to be called, is always up to something. When he ends up finding an old class ring under strange circumstances, he begins a hunt to find its owner.		Surj (Surj Patel G) doesn't get much sleep.
	Pengkai (Pengkai Pan G) wants to know what Damien is doing.		Olivia (Rachel Kline 01) is somewhat interested in Smoots.

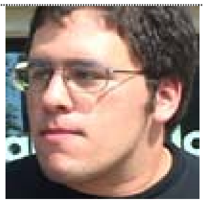
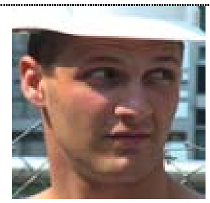
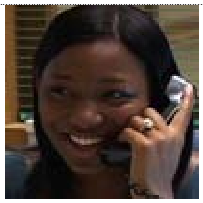
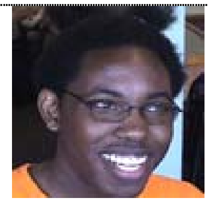
	Mike (Mike Razo 03) plays a mean bongo drum.		Sayre (Sayre Neufield C) is curious about the ring.
	Imani (Imani Ivery S) works at the Alumni Office. When Fred comes in with the ring, she knows exactly what to do.		Yao (BaoYi Chang 02) and Zedrick have long conversations with each other.
	Olga (Olga S) wants to clean the Eastman Statue.		Zedrick (Jumaane Jeffries 02) just wants some respect. And a ticket to the movie.

Figure 20 Story characters of *MIT in Pocket*

6.1.2 Location

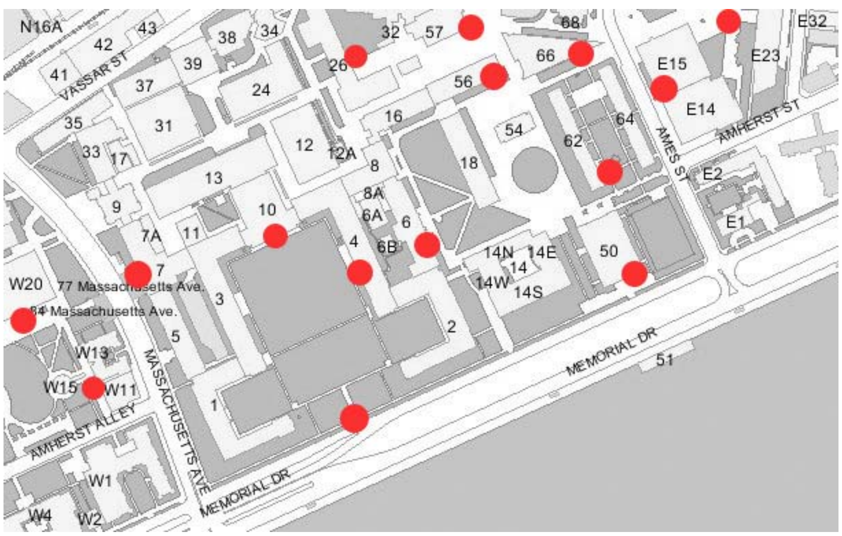


Figure 21 Story locations of *MIT in Pocket*

To reflect the spirit of MIT, fifteen indoor and outdoor story locations, all of whose physical environments have unique atmospheres, were carefully chosen from across the MIT campus (Figure 21): Kendall Square, the Media Lab, Walker Memorial, Stata Center, Killian Court, East Campus, Building 4, Building 26, Building 56, Building 66, Lobby 6, Lobby 7, Lobby 10, The Chapel, and the Student Center. The criteria for choosing these fifteen places were twofold: 1) most leading characters' activities happen at these locations. For example, Anthony likes to take a jog every day and is an MIT tour guide. He often appears around several famous buildings.

Betty is studying at the MIT Department of Architecture, so that she likes to go to the new buildings that were designed by well-known architects, such as Frank O. Gehry. 2) Many MIT visitors would like to go to these places. Two of the production team members, Chris Toepel and Lilly Kam, were MIT tour guides, and they knew popular tour places on the campus.

6.1.3 Story Evolution

In *MIT in Pocket,* story evolution means not only adding new content into the existing story database, but also changing filmed content as the team got more ideas along the way. For example, two characters were independent at the beginning, but later could be linked together. One of the characters was Damien and he had an exam at Walker and did a really bad job on it. The clip showed him in the testing room, writing stuff down, and in his head he was imagining himself crumpling up the paper and walking out, really angry. That clip was filmed early on in the summer and done with. A couple of months later, in the IAP 2003 workshop, students were interviewing professors for possible clips for adding into M-Views. One of their professors, Professor Sadoway, was talking about how, on an exam once, a student drew a little stick figure man holding one foot and hopping on the other one. On the bottom it said “Do the hurtin’ dance.” They thought that was a really funny story, and that immediately made them remember the story with Damien, so they went back and filmed more for that scene. They filmed Damien actually writing down everything Sadoway said, as if Professor Sadoway were talking about Damien in that clip. It was a great way to connect those two clips and no one could tell that it was a revised story.

After spending five months on filming, editing, and music composition, the team started to discuss how to deploy this story at the MIT campus. Two versions were proposed. The first version would take place over three hours, each hour being a different time of the day condensed. The first hour would be the morning, the second would be the early afternoon, and the third would be late afternoon. The events that occur during those times would appear during those hours. The second version was in absolute time. The events occur when they would occur. That would be an all-day experience. The production team wanted to see how people responded to that. With the all-day experience, people would also be doing other things, their life chores. *MIT in Pocket* would hopefully be a nice, pleasurable experience, sort of like listening to a Walkman.

6.1.4 Story Evaluation

The M-Views system version 0.2 was also under development in 2002. At the beginning of 2003, after the MapAgent was finished, David Crow and Lilly Kam created story scripts for *MIT in Pocket* and preliminary evaluations began. Within the first few months of 2003, we invited about ten MIT students and faculty to walk through *MIT in Pocket*. A typical story experience started with an instruction about the M-Views system 0.2 and the background information of *MIT in Pocket.* During each walk-through, a team member had to carry a laptop, as the M-Views server, escorting the audience. The reason for carrying a server was because of technical constraints. (MIT’s sub-network was unable to assign dynamic IP addresses to iPAQs at that moment.) During each round of evaluation, our team member would only give the audience help if asked to. In this way, we were able to observe relatively objective mobile cinema experience. Overall, participants enjoyed the experience. People felt that the connection between the clips and the actual locations in which they were standing was very powerful and compelling. They enjoyed the fictional stories about the students, and wished that they could learn more about certain characters. Some participants also wished that there were more clips, distributed over more places on campus. Several participants generally enjoyed learning about MIT events they knew little about through the documentary clips. Several specific findings and considerations are as follows.

- Technically, *MIT in Pocket* works, but requires tremendous maintenance efforts. For example, the escort had to be mindful of the battery and charge it up from time to time. For the three-hour version of *MIT in Pocket,* changing the battery twice was inevitable. Due to the sub-network issue, a laptop server had to be carried by either the participant or the escort. There are at least two issues that cause the high-cost maintenance. First, David Crow did discuss various technology constraints, but the whole team was optimistic about solving these technical problems in the future. The whole production team was mainly focusing on character development, plot design, location, and other production issues. They knew that battery life would be a critical problem, based on what we had learned from the evaluation of *Another Alice,* but no one really paid attention to this issue. Second, the M-View system 0.2 had not been completed by the time of evaluation. The location detection technique was incompatible with the set-up of the MIT wireless networks.
- The major production of *MIT in Pocket* was completed, but the overall expectation was too high. Compared to linear film or TV productions, a mobile cinema production required several additional processes, such as annotating contextual information, converting video formats, authoring story scripts, and field tests. The team had made a production plan in the middle of summer 2002, but the overall cost of making *MIT in Pocket* was more than twice what they expected in terms of production time and effort. After filming major actions, it took Lilly Kam and Deb Liu a whole semester to do post-production, convert formats, and create story scripts with M-Studio.
- From a user perspective, the interface of the M-Views Presenter needed to be simplified. In the previous system implementation, a typical user experience to launch a video clip require three steps: receiving a message, reading the message, and clicking a button to launch the associated video clip. We collected comprehensive feedback from participants about specific things they liked about the stories, how easy or difficult it was to use the PDA or M-Views interface, and other aspects of the mobile storytelling experience. From the feedback, we have simplified the interface design of the M-Views system 0.2 by adding an "auto-playback" feature, which launches video clips as soon as they are received. In this way, the viewer would not have to press or click any additional buttons.
- The story database of *MIT in Pocket* is extensible. After finishing the four major story threads in summer 2002, much new content has been filmed and some of it has been added into the story database of *MIT in Pocket:* interviews with Professors Donald R. Sadoway, Bill Corbett, Ted Selker, Melissa Nobles, and Alexander Slocum; the F&T Diner dedication ceremony in Kendall Square; the Class of 1953 Cardinal and Grey Dinner and Dance at Walker Memorial, and other 2003 Tech Reunion events; 2003 Commencement at Killian Court; RoBallet (a special children's dance and programming workshop held at the Media Lab); a Buddhist blessings ceremony; and so on. The open structure can work very well for adding new stories, but we have not yet tested the overall user experience after these new clips were added.

In summary, the whole process of creating *MIT in Pocket* was creative, productive, and fun. The multidisciplinary team members learned, worked, and created together. Through brainstorming, exploring, and prototyping, we have opened much broader creation domains than what we learned from *Another Alice.* Although there were still several technology and design issues that needed to be addressed in order to cope with various sources of uncertainty, this project indeed led us to a new mobile cinema creation in June 2003. In the next section, I will describe the details of this new production.

6.2 *15 Minutes*

15 Minutes is a murder-mystery-like story, created by David Crow and produced by a small group of MIT students. In the story, everything happens in three places in a corporation building, a division of InfoSafe, where critical government data are stored and some wild rumors about a robbery threat are floating around. To participate in the story, the audience carries an M-Views client, walks around the building, watches video clips on the M-Views client, and investigates what is going on in the story. However, unlike *Another Alice,* in which all participants need to start from the same location and pay full attention to what each character tells them, *15 Minutes* allows participants to start the story from any of three locations. Whether the participant follows the character's suggestions or not, the M-Views server and the story scripts of *15 Minutes* are always able to maintain story coherence. Based on the sequence of the locations visited, there are many possibilities of how a particular plot folds. For example, there were more than 18 different ways of experiencing *15 Minutes* at the UbiComp'03 conference in Seattle. Unlike *MIT in Pocket,* which involves more than 40 characters and 15 different locations at MIT, *15 Minutes* involves only three characters and three locations. Hence, it is relatively easy for the participant to decide where to go next, compared to the decision-making in *MIT in Pocket.* Furthermore, *15 Minutes* can be set up in any 802.11-enabled building with spaces that mirror the locations of the story, such as a conference room, an elevator lobby, and a copy machine room. The story aims to offer a neat, reliable, and scalable story experience that can be finished within 15 minutes.

6.2.1 Story Content

6.2.1.1 Locations

There are three story locations: Elevator, Copier, and CEO Office. These three locations are covered by three corresponding wireless hot spots. This means that the M-Views server knows whether or not a participant is within a particular location. These three locations may or may not be the original story production locations at the MIT Media Lab: the lobby, a conference room on the second floor, and a hallway with a copier on the fourth floor. *15 Minutes* was also presented at the UbiComp'03 conference, where the three story locations included a conference room, a telephone desk, and an elevator. The current setup for evaluation at the MIT Media Lab is deployed on one floor (Figure 22) for participants' ease in walking around the building. The advantages and disadvantages of linking the presentation locations to the production locations will be discussed in a later section.

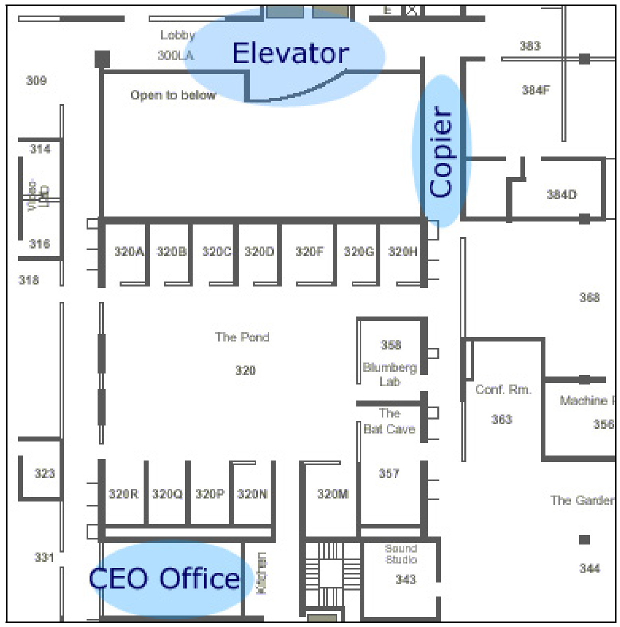


Figure 22 Story locations of *15 Minutes*

6.2.1.2 Characters

Figure 23 Story characters of *15 Minutes*

There are three main characters (Figure 23)—David (an elevator repairman), Michael (head of a division of InfoSafe), and Eve (Michael's secretary)—and three supporting characters, Chi, a professional hit man, and two policemen. In the story, the three main characters proactively talk to participants and ask them to do certain tasks or go to a certain location, but the participants choose whether or not to follow their words. Based on different walking paths, participants may find the different sides or characteristics of each character. For example, one viewer may know Eve as Michael's secretary in one walking path. However, it may be revealed to another viewer that she is a special government agent in another walking path. The different characteristics do not conflict with each other, since each participant has a unique walking path.

6.2.1.3 Plots

There are three possible starting points and five possible endings (Chart 10). Because each participant has his or her own walking pattern, various combinations of received story sequences present different story experiences. Appendix Two indicates detailed storyboard and scripts. The following chart illustrates the overall structure of *15 Minutes.*

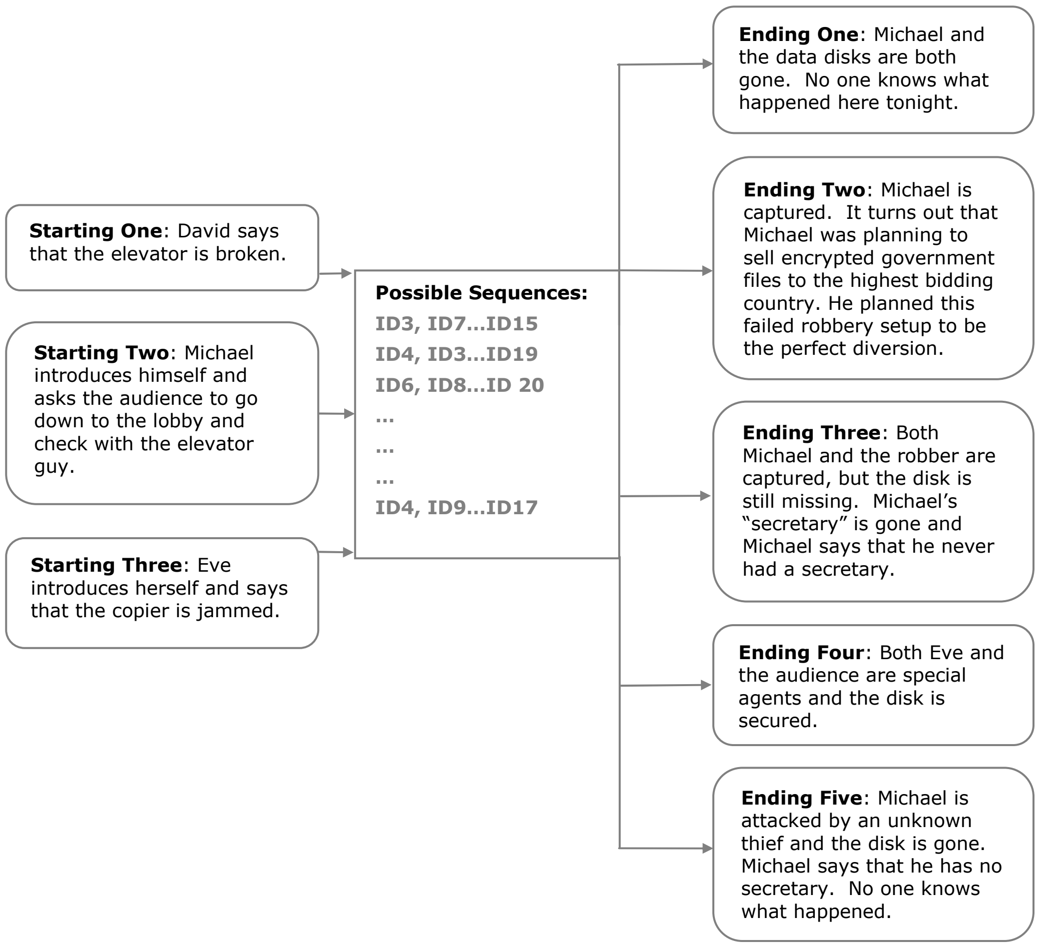
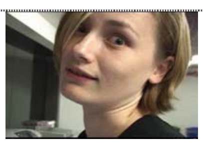
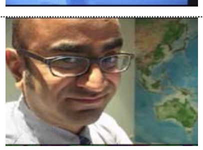


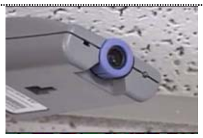
Chart 10 ***15 Minutes*** **plots**

The following tables (10 and 11) are two examples of many story experiences from the log files on the M-Views server.

The First Walking Path Generated at UbiComp 2003			
1st Sequence	**2nd Sequence**	**3rd Sequence**	**4th Sequence**
IDX(00Intro)	ID1(02MeetMichael)	ID0(01DavidEnters)	ID2(03EveIntro)
5th Sequence	**6th Sequence**	**7th Sequence**	**8th Sequence**
ID20(05MichaelPower)	ID7(08EveCameras)	ID12(13Fight)	ID14(17GoodEnding)

Details of the First Walking Path Generated at UbiComp 2003
IDX

	**Any place** An introduction clip
ID1	
	**CEO office** The participant meets Michael. Michael: You're working late. That's something I like to see in my staff, although….
	I don't think we've been introduced. I'm Dr. Michael Beauford, and I head up this division of InfoSafe. Pleasure to make your acquaintance. Listen, when you go down to the lobby, check with the elevator guy for me. I've been having trouble with my phone, and I can't call down. Thanks.
ID0	
	**Elevator** The participant meets David. Although frustrated with the lift, he appears to be troubled by other things.
	David: Elevator's busted. You're gonna have to take the stairs. Hang on a minute. Sorry, I didn't mean to put you off. I'm David - I was called to fix the lift. Looks like the secondary power just came on... Strange...it still won't run, but tell Michael if you see him.
ID2	
	**Copier** The participant meets Eve. She's an attractive and talkative young woman, perhaps the only friendly face around. Eve: Can you help me with the copier? I think it's jammed. Wait... there it goes.
	Oh, I'm Eve, Michael's secretary. You're new here... I've been seeing some new faces, especially at these hours. But it's good to have company. Uh, please don't tell Michael I'm still making copies. They're a few hours late.
ID20	
	**CEO office** There's some unsettling satisfaction in Michael's voice. He puts on his jacket as he lectures the participant, so suddenly animated that he seems almost excited as he leaves.
	Michael: Great, so the elevator's still useless. What good is power if you can't use it? You know, a machine is only as reliable as its weakest part. Just like this company. And something as irrelevant as a broken elevator can fuck things up like you can't imagine. Sorry, I'm a little tense. This is quite a day. If you'll excuse me, I must take care of something. I have to make sure the rest of this place isn't falling apart.

ID7	
	**Copier** Eve glances at the participant with certainty, as if expecting agreement. And there appears to be some purpose behind her curiosity.
	Eve: Hey, something's going on here. I'm sure of it. Security's been very tight, especially since they installed those new surveillance cameras. But it's silly to think it'll help. It's just gonna make everyone jumpy.
	Hey listen, I can't go downstairs. I've got my own work to do. But maybe you could check it out for us. I been hearing some wild rumors about a robbery threat...
ID12	
	**Elevator** David: You shouldn't be here! There's a possible security risk and you are required to-
	David's warning is cut short. The dead elevator has sprung to life; its doors slide open to reveal Chi. The two stop abruptly. There is a paralyzed moment of bitter recognition. Chi aims a gun at David and the viewer. David: BACK OFF, Chi! You stepped into the wrong fucking room! Game over.
	Chi: You're just as pathetic as I left you. Ach... wasting your time... Can't even see... when you're being played-
	David: Got something to tell me? Well speak up! As if I'd believe anything you have to say!
	David and Chi fight.
	Chi: The robbery, our reunion... both arranged by the same person... we've been set up.
	A flashback shows Michael paying Chi for the job.
	David: Go stop Michael! We can't let him get away!

ID14	
	**CEO office** Michael is in good spirits until he realizes why you are here. When you refuse to move, he reaches into his coat. David interrupts just in time. Michael: Ah, my late worker. This is a surprise. I'm afraid I was just on my way out... How about that. I guess I'll just have to take care of you too-
	David: Hold it, Michael! That's far enough.
	Cut to the ending: David: Michael had a data disk on him with some encrypted government files. From his records, it looks like your company was storing names of government agents on "special" assignment in unfriendly places. He was planning to sell the disk to the highest bidding country. Michael planned this failed robbery setup to be the perfect diversion. But I guess he didn't count on you. Thanks.

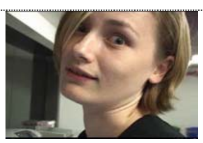
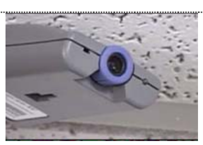
Table 10 The first walking path generated at UbiComp 2003

The Second Walking Path Generated at UbiComp 2003

1st Sequence	2nd Sequence	3rd Sequence	4th Sequence	5th Sequence
ID1(02MeetMichael)	ID0(01DavidEnters)	ID3(04DavidWatch)	ID2(03EveIntro)	ID7(08EveCameras)
6th Sequence	**7th Sequence**	**8th Sequence**	**9th Sequence**	
ID20(05MichaelPower)	ID6(07EveLeaves)	ID13(14Fight)	ID16(18BadEnding)	

Details of the Second Walking Path Generated at UbiComp 2003

IDX	
	**Any place** An introduction clip
ID1	
	**CEO office** The participant meets Michael. Michael: You’re working late. That’s something I like to see in my staff, although….
	I don’t think we’ve been introduced. I’m Dr. Michael Beauford, and I head up this division of InfoSafe. Pleasure to make your acquaintance. Listen, when you go down to the lobby, check with the elevator guy for me. I’ve been having trouble with my phone, and I can’t call down. Thanks.
ID0	

	**Elevator** The participant meets David. Although frustrated with the lift, he appears to be troubled by other things.
	David: Elevator's busted. You're gonna have to take the stairs. Hang on a minute. Sorry, I didn't mean to put you off. I'm David - I was called to fix the lift. Looks like the secondary power just came on... Strange...it still won't run, but tell Michael if you see him.
ID3	
	**Elevator** An old pocket watch looks out of place in David's hand. He holds it painfully, not like a memento, but like a curse. David: Sorry, power's on, but the lift's still broken.
	Do you have the time? This old watch runs a bit slow. So useless... I bet you're wondering why I carry around this piece of junk... well... I don't know. Doesn't make sense really. To remind me of someone, I suppose. Flashback to David giving the pocket watch to an old friend (Chi).
	David: Hey, I should really get back to fixing this. Don't work too late.
ID2	
	**Copier** The participant meets Eve. She's an attractive and talkative young woman, perhaps the only friendly face around. Eve: Can you help me with the copier? I think it's jammed. Wait... there it goes.
	Oh, I'm Eve, Michael's secretary. You're new here... I've been seeing some new faces, especially at these hours. But it's good to have company. Uh, please don't tell Michael I'm still making copies. They're a few hours late.
ID7	
	**Copier** Eve glances at the participant with certainty, as if expecting agreement. And there appears to be some purpose behind her curiosity.
	Eve: Hey, something's going on here. I'm sure of it. Security's been very tight, especially since they installed those new surveillance cameras. But it's silly to think it'll help. It's just gonna make everyone jumpy.

	Hey listen, I can't go downstairs. I've got my own work to do. But maybe you could check it out for us. I been hearing some wild rumors about a robbery threat...
ID20	
	**CEO office** There's some unsettling satisfaction in Michael's voice. He puts on his jacket as he lectures the participant, so suddenly animated that he seems almost excited as he leaves.
	Michael: Great, so the elevator's still useless. What good is power if you can't use it? You know, a machine is only as reliable as its weakest part. Just like this company. And something as irrelevant as a broken elevator can fuck things up like you can't imagine. Sorry, I'm a little tense. This is quite a day. If you'll excuse me, I must take care of something. I have to make sure the rest of this place isn't falling apart.
ID6	
	**Copier** Eve's confidence in the participant is reassuring. She makes her plan sound cute, albeit childish. Then she leaves.
	Eve: So Michael's stepped out? Do you think I have enough time to stop by his office? I need to drop off these copies. I'll just walk in and out. He'll never know I was even there. And we both know he's too busy to notice anyway. I'll be printing some documents while I'm away. You'll stay and watch the printer for me, right?
ID13	
	**Elevator** David: You shouldn't be here! There's a possible security risk and you are required to-
	David's warning is cut short. The dead elevator has sprung to life; its doors slide open to reveal Chi. The two stop abruptly. There is a paralyzed moment of bitter recognition. Chi aims a gun at David and the viewer. David: BACK OFF, Chi! You stepped into the wrong fucking room! Game over.
	Chi: You're just as pathetic as I left you. Ach... wasting your time... Can't even see... when you're being played-
	David: Got something to tell me? Well speak up! As if I'd believe anything you have to say!
	David and Chi fight.

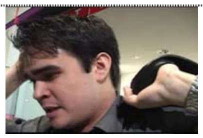
	Chi: The robbery, our reunion... both arranged by the same person... we've been set up.
	Chi's attempts are cut short, as Michael intervenes. It seems his own plans have had an unexpected turn of events. He points his gun at the participant. Michael: You stupid monkeys! One of you three has it. WHERE IS THE LIST?
	But the confrontation ends with the arrival of the police. Policemen: FREEZE! It's all over.
ID16	
	**Elevator** If Eve never finds out about Michael's actions or the robbery threat.
	David: Michael and the data. They're both gone. I've got no clue what went down here. We might never really know.

Table 11 The second walking path generated at UbiComp 2003

6.2.2 Story Evaluation

In this section, I discuss two sets of story evaluations: UbiComp and Media Lab. The first round of intensive evaluation for *15 Minutes* occurred at UbiComp'03 in Seattle on October 13, 2003. We had a short paper published in this conference and were invited to give a demonstration. The second round of intensive evaluation took place at the MIT Media Lab, where the original story production took place, in spring 2004. Both sets of evaluations have been analyzed by the same methods.

6.2.2.1 UbiComp'03

At UbiComp'03 (Figure 24), about twenty-five people signed out M-Views clients, walked around in the lobby on the second floor, and watched *15 Minutes* clips during a four-hour demonstration section. At the beginning, each participant was given a brief introduction to how the system works and some background information about the story. At the end, some participants volunteered to fill out a questionnaire regarding their experience of *15 Minutes.* The story experiences were evaluated based on the questionnaires (14 sets of answers were valid), several on-site interviews, and log files from the M-Views server.

Figure 24 *15 Minutes* at UbiComp 2003

Questionnaire:

How many times did you play 15 Minutes?

13 people played once, and 1 person played twice. The person who played twice got a different ending each time and his perception of the characters or events changed. In his first trial, he received the ending where the police arrest Michael in the lobby, but the data disk is missing, having been mysteriously stolen from Michael. This is actually the result of how the viewer failed to pay enough attention to the Eve character. Since the viewer failed to go to Eve's location at certain critical moments of the story, she was able to sneak into Michael's office (off screen) and steal the disk. This is unknown to the viewer. In his second attempt, the viewer was able to stop Michael in his office just in time to retrieve the disk from him, and before Eve could carry out her secret plans. This participant was especially serious about strategizing the optimal paths for his experience. He did not just randomly walk around. In fact, his favorite aspect of the story was the strategy involved, especially during the second time he played.

How did you decide which locations to visit? (Answer all that apply)

A. The characters told me where to go.
B. I went to the characters and/or locations that appealed to me.
C. I believed that something might happen at a particular location.
D. I randomly chose where to go.
E. For different clips, I deployed different strategies.

Choices	A	B	C	D	E
Participants	6	2	5	6	3
Percentage	27%	9%	23%	27%	14%

Table 12 Choosing story locations at UbiComp 2003

Did you want to talk to other participants while playing 15 Minutes?

9 people did not want to talk to other participants, but 5 did. A few people actually interacted with each other, comparing their stories so far. One participant said, "One guy I passed was also

playing and spoke to me and startled me; then I did it to someone else!" There was much interest between participants in what their different endings were.

Was anything in the story confusing? If yes, could you briefly mention them?
8 people were not confused by anything, but 6 people were confused at certain points. This was mainly due to location-detection problems. For example, one participant wrote, "I tried to go to Eve, but the system thought I had gone to David; after that everything seemed kind of screwed up and I was getting messages from the wrong locations." Clips seemed to be triggered in the wrong physical context not because of problems with the story logic, but because of a lag in location detection. For example, if a viewer went to the "Elevator" but quickly passed the "CEO Office," he or she might receive a clip from the "CEO Office" first.

What was your favorite thing(s) about the story/experience?
Answers were various. For example, 5 people said that they loved the acting. Some responses include: "The video clips were entertaining, kind of creepy, well acted. It definitely created an air of suspense." "The action and the evil boss." Another favorite element was "choosing what to do;" another wrote, "choosing your own adventure rocks." One person felt "intimacy when people spoke directly to me." A few were also fond of the interaction with other participants in mobile cinema. One person enjoyed "surprise elements" in the story. Several people also expressed that they were thrilled by the concept of mobile cinema.

Was there anything you disliked about the story/experience?
The main complaint was the disconnection between story locations and physical locations: "It was hard to find the locations"; "Walking around an open space felt weird. It would be better at an actual location." Two people demanded more possible interactions with the characters in the story in addition to walking around to meet them. Another person wished that it were "easier to merge video clips."

Do you have any additional comments on the story and the mobile system?
Additional comments fell into four groups. First, two people felt that the story experience lacks agency. One responded, "The participation of the user, sometimes involved by the characters, seems too fictitious because he has no means to participate." Second, several people wanted to try the story again at the actual production locations, rather than the substitute locations at UbiComp. Third, several participants expressed appreciation: "This is a great idea. More, please, to download for my PDA. It is a useful learning tool and a good demo for context- and history-sensitive computing." Finally, two people suggested improvement of the interface design.

Based on the following scale, how enjoyable was the experience?

Scale	**1**	**2**	**3**	**4**	**5**	
	Not enjoyable		Okay		Extremely Entertaining	
Vote	0	0	3	9	1	Average=3.84

Table 13 *15 Minutes* experience at UbiComp 2003

Log File: Twenty-five log files have been collected from the server at UbiComp'03. The code in Figure 26 is an example for discussing what a typical data set is and how the data analyses are conducted. By analyzing log files, I am able to calculate the viewing time and the walking path by a story participant.

Time: Among the 25 participants, 20 people finished full versions of *15 Minutes*. That means that they watched one of five possible endings. (For five people who did not finish a full story, some of them said that they had certain technical problems; the others didn't have time to experience full stories.) The average viewing time was 11.2 minutes; the longest one was 16.72 minutes; and the shortest one was 9 minutes.

Participant	P1	P2	P3	P4	P7	P8	P9	P10	P11	P12
Minutes	10.4	10.0	10.0	9.5	10.1	9.1	14.4	11.2	9.1	9.3

Participant	P13	P14	P15	P16	P17	P19	P20	P21	P24	P25
Minutes	13.3	9.3	14.8	9.5	11.2	16.7	11.4	9.9	11.1	13.6

Table 14 Participation time at UbiComp 2003

Walking path: A log file recorded a sequence of message IDs, which indicates an order of viewed story clips and a walking path (the order of places where the participant has walked through.) For example, a log file recorded a sequence of message IDs:
IDX>ID2> ID7>ID8>ID20>ID12>ID18.

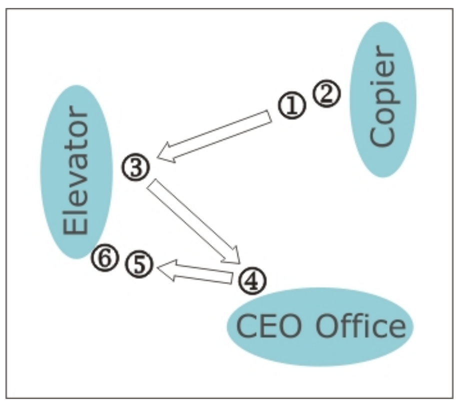


Figure 25 A walking path

The corresponding video sequence is:
00INTRO>03EVEINTRO>08EVECAMERAS>09DAVIDWARNING>05MICHAELPOWER>13FIGHT>16BADENDING

And the corresponding walking path is:
Copier>Copier>Elevator>CEO Office>Elevator>Elevator

File head

```
<MV3 Connected="Mon Oct 13 19:25:16 EDT 2003" Context="Idle
UBICOMP(423,779)" Email="mviews@media.mit.edu" IP="10.201.121.226"
Password="pwd">
  <Subscriptions>FifteenMinutes</Subscriptions>
  <Maps>UBICOMP</Maps>
  <Teams>TEAM_C</Teams>
  <Stories>
- <FifteenMinutes>
```

Story flags

```
  <Flags>
  <David_value value="3" />
  <Michael_value value="3" />
  <Eve_value value="2" />
  <Fight value="3" />
  </Flags>
  <Heuristics />
```

Seen video clips

```
- <Seen last="ID18">
  <IDX time="1066086726603" />
  <ID2 time="1066086791997" />
  <ID7 time="1066086878361" />
  <ID8 time="1066086945037" />
  <ID20 time="1066087002039" />
  <ID12 time="1066087089434" />
  <ID18 time="1066087350430" />
  </Seen>
  </FifteenMinutes>
  </Stories>
```

Received messages

```
- <Inbox>
  <ID18 doc="FifteenMinutes" viewed="1066087362577" />
  <ID12 doc="FifteenMinutes" viewed="1066087095483" />
  <ID20 doc="FifteenMinutes" viewed="1066087322499" />
  <ID8 doc="FifteenMinutes" viewed="1066086945928" />
  <ID7 doc="FifteenMinutes" viewed="1066086883949" />
  <ID2 doc="FifteenMinutes" viewed="1066086806457" />
  <IDX doc="FifteenMinutes" viewed="1066086732110" />
  </Inbox>
  <Outbox />
  <Seen />
  </MV3>
```

Figure 26 An example of a log file

Figure 25 illustrates the walking path generated by a participant at UbiComp. Table 15 and Chart 11 list all walking paths generated by 20 participant members. (C=Copier, E=Elevator, and O=CEO office.)

Participant	P1	P2	P3	P4	P7	P8	P9	P10	P11	P12
Walking Path	C	C	O	C	C	O	C	E	C	C
	C	E	C	C	E	E	C	O	O	O
	E	C	E	E	E	C	O	E	E	C
	O	O	C	E	E	C	C	C	E	E
	E	E	E			E	O	C	E	E
	E	E	E			E	E	E		
							E	E		

Participant	P13	P14	P15	P16	P17	P19	P20	P21	P24	P25
Walking Path	C	C	O	O	O	C	O	O	O	O
	O	O	C	O	E	O	E	O	E	O
	C	C	O	O	O	O	O	O	E	O
	E	E	O	E	E	O	E	E	O	E
	C	E	C	C	C	E	C	E	E	C
	O		C	E	C	C	C		E	E
	O		O	E	E	O	E			E
	E		E		E	E	E			
	E		E			E				

Table 15 Walking paths at UbiComp 2003

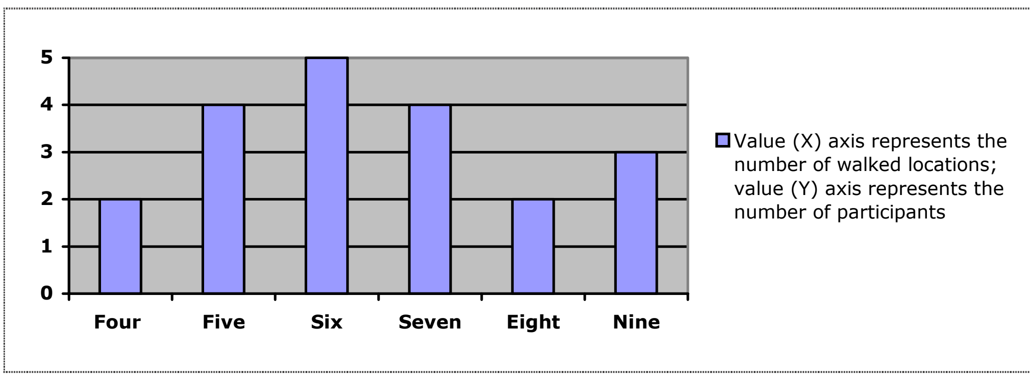


Chart 11 Histogram of user data at UbiComp 2003

There were only two pairs of participants (12~14 and 16~25) who had the same walking paths.

6.2.2.2 Media Lab

Since we established *15 Minutes* at the MIT Media Lab last November, more than fifty people have participated in this story. Some of them were Media Lab's sponsors and tested it in ad hoc fashion. In order to evaluate *15 Minutes* in more structured ways, we formally invited eighteen people to Media Lab both in March and April 2004. They walked around on the third floor and watched *15 Minutes.* The evaluation procedures were similar to those in UbiComp'03. A team member gave them a brief introduction to how the system works and some background information about the story. At the end, all participants filled out a questionnaire regarding their experience of *15 Minutes.* The story experiences were evaluated based on the questionnaires (18 sets of answers were valid), several on-site interviews, and log files from the M-Views server.

Questionnaire:

How many times did you play 15 Minutes?
15 people played once, and 3 people played twice. All the three people who played twice got a different ending each time and their perception of the characters or events changed.

How did you decide which locations to visit? (Answer all that apply)

A. The characters told me where to go.
B. I went to the characters and/or locations that appealed to me.
C. I believed that something might happen at a particular location.
D. I randomly chose where to go.
E. For different clips, I deployed different strategies.

Choices	**A**	**B**	**C**	**D**	**E**
Participants	7	9	11	4	4
Percentage	20%	26%	32%	11%	11%

Table 16 Choosing story locations at Media Lab

Did you want to talk to other participants while playing 15 Minutes?
10 people did want to talk to other participants, but 8 did not. People who wanted to talk to each other were people who paired together for evaluation. Paired people often first took different strategies to walk around, and then met at a common place to discuss what they had watched. Several paired people didn't talk to each other until the very end.

Was anything in the story confusing? If yes, could you briefly mention them?
Seven people were feeling smooth about participating in *15 Minutes;* 11 people were confused at certain points. These confusion points can be classified into three categories: technical issues, story content issues, and the audience's role. For example, one participant wrote, "Some of the clips didn't seem to flow too well." Several people didn't understand parts of the story details: "I'd like to know more about the relationship between the two guys who fight - the flashback makes me curious about the back-story." "One thing that was strange: at the beginning of one clip, you see a surveillance video, which I suppose Michael is watching. Isn't he suspicious when he sees Eve, if he doesn't know her?" This story is less than 15 minutes, and a longer version might be able to tell more details about the characters and their relationship. Finally, two people said that they would have liked to "have a better introduction," so that they would know their roles in the story at the very beginning.

What was your favorite thing(s) about the story/experience?
Most audience members were impressed by story contents, such as characters, acting, action, music, and videography. "The acting, especially Michael's, was stellar! He seemed very sinister and boss-like." "The story and acting were generally good, the video was professional, which adds a lot to the entertainment value." "I really liked the music with the clips, it made me feel like I was in an action movie!" Several people liked to be involved in this story. "What I liked most was the involvement in the actual movie because oftentimes people want to be actively

involved, and this system allowed me to." "I like the characters asking me to do things." Many people were impressed by the concept of mobile cinema. "I liked the feeling of being in the space where the action took place - it intensified the experience." "I liked the back and forth of being a spectator and part of the action." "It was exciting to actually move around instead of being forced to sit around through an entire story." "I like that it was interactive and that you could get a different ending."

Was there anything you disliked about the story/experience?
The main complaint was fourfold. First, four people thought that this story is too short. "I would like to experience the characters more, become familiar with who they are and what they do at the company." "It was too short the first time, not enough build-up to the final scene. But it does leave me wanting to play again with a different strategy." Second, there were several technology issues that bothered people. "I didn't like how clips were played when I was trying to leave an area. I got two clips in a row at the same place, which I didn't want." "Only thing I disliked was trying to capture the wireless connection and the slow download." Third, three people wanted to interact with the characters, but the system didn't provide this feature. "I wanted to interact back. When a character would say something, I wanted to respond." "Lack of interaction with characters: I was able to alter my location, but not really communicate with the characters." Finally, several people were still confused by parts of the story content and experience. "Sometimes you aren't sure if you missed something."

Do you have any additional comments on the story and the mobile system?
Most additional comments are related to the previous two questions. For example, "I wanted to be able to communicate with the characters, that would have been cool. Like some options of things to say." "I wish I felt more involved in what was going on. Yeah, my choices affected events, but I wasn't conscious of that during the experience. I felt like I was bumbling along and accidentally caused events to happen." "It gets pretty complicated with just 3 locations, but it would be cool to have a 'mystery' location which the viewer may or may not discover." "As the technology develops it would be great to develop the plot into something more complex, and the locations into greater nuance places. More surprise is needed, and more time for greater immersion." These suggestions are valuable for future mobile cinema productions.

Based on the following scale, how enjoyable was the experience?

Scale	**1**	**2**	**3**	**4**	**5**	
	Not enjoyable		Okay		Extremely Entertaining	
Vote	0	1	2	11	4	Average=4

Table 17 *15 Minutes* experience at Media Lab

Log File: Eighteen log files have been collected from the server at the MIT Media Lab. Some findings are as follows.
Time: For the eighteen participants, three people participated in this story twice, and one log file was missed. The average viewing time is 17.6 minutes; the longest one is 29.42 minutes; and the shortest one is 9.8 minutes.

Participant	P1	P2	P3	P4	P5	P6	P7	P8	P9	P9-2
Minutes	18.8	9.8	12.7	11.5	29.4	10.7	13.2	14.2	17.6	11.2

Participant	P10-2	P11	P11-2	P12	P13	P14	P15	P16	P17	P18
Minutes	13.2	15.3	16	13.5	18.8	22.6	22.2	23.4	11.8	11.2

Table 18 Participation time at Media Lab

Walking path: We also parsed the log files to show the audience's walking paths.

Participant	P1	P2	P3	P4	P5	P6	P7	P8	P9	P9-2	P10-1
Walking Path	O	C	E	C	C	C	E	E	E	C	E
	C	E	O	C	C	E	E	E	E	O	E
	E	O	E	E	E	C	O	O	O	O	C
	E	C	C	C	C	E	E	C	C	E	E
	O	E	C	E	E		E	E	E	E	E
	C		E		E			E	E		
	E		E								
	E										

Participant	P10-2	P11	P11-2	P12	P13	P14	P15	P16	P17	P18	
Walking Path	C	O	C	O	O	C	O	E	O	E	
	O	E	C	O	C	O	C	E	C	O	
	C	E	E	O	E	E	O	O	C	E	
	O	O	O	C	E	O	C	C	E	E	
	E	C	C	C	O	E	E	E	E		
	E	E	O	E	C	E	E	E			
		E	E	E	E						
			E		E						

Table 19 Walking paths at Media Lab

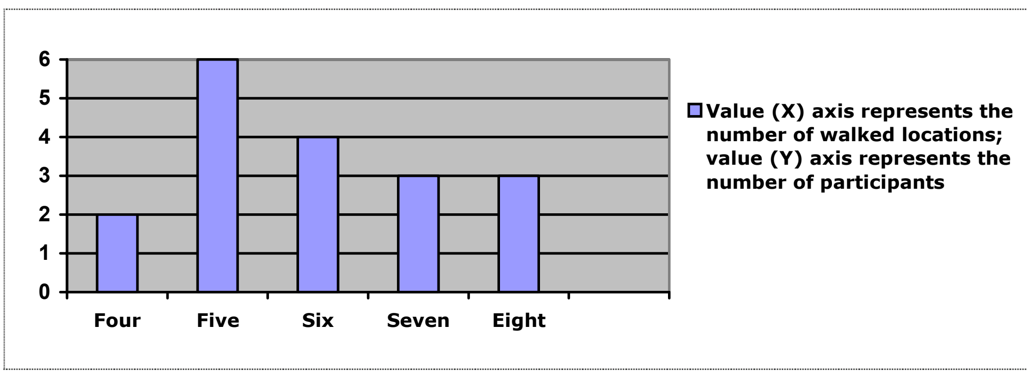


Chart 12 Histogram of user data at Media Lab

There are only two pairs of participants (P1~P13 and P9~P16) who had the same walking paths.

6.2.2.3 Comparison Notes

We interviewed participants from both UbiComp 2003[58] and the MIT Media Lab after their experience. In this section, I compare data from both questionnaires and interviews to understand what were the main different and similar story experiences between the two sets of evaluations. I will also discuss possible reasons behind these differences and similarities.

Location matters: Several sets of interview findings and log file data indicate that matched locations (story presentation locations match story production locations) help mobile cinema experience in general. Participants spent more time and walked longer at matched locations. At the MIT Media Lab, the average participation time was 17.6 minutes compared to 11.2 minutes. People walked to more locations at the MIT Media Lab. In UbiComp, there was one person who participated in *15 Minutes* twice; in contrast, three people at Media Lab walked the story twice. All the four participants had different endings each time.

From both the interviews and questionnaires, people prefer to participate in mobile cinema stories at the matched locations. At Media Lab, people said that the matched locations “intensified the experience.” Another person felt the connection to the physical space could further be strengthened if we could “incorporate artifacts of a space as well as the location.” In contrast, the main complaint from the participants in UbiComp was the disconnection between the substitute locations that were spread across the lobby floor and were sometimes hard to find. They did not connect with the physical space at all, which was a major disadvantage to their experience. A few people even suggested we do an M-Views film on the conference demonstrations. One person said, “The only thing that bothered me was the way the setting of the story and the physical locations were disconnected. Perhaps if you made a movie about UbiComp, the location aspects would be more interesting. For example, it would be interesting to see a location-based guide to all the demos here.”

There are several reasons why people prefer matched locations. First, matched locations help them to navigate physical spaces. At UbiComp, several people could not find story locations; even though we put a clear sign in a relativly small hotel lobby. Second, when the presentation and production locations match, the ambient environments reinforce the story atmosphere and the participants are more likely to immerse into stories. At a result, they are willing to spend more time and walk to more locations.

Content is key: Although locations matter, the key to engaging the participant is still the story content. By comparing what people liked and disliked in both UbiComp and the MIT Media Lab, I found many similarities. For example, many people from both groups of participants enjoyed watching the fight scene, the acting of the evil boss, and the mystery secretary. A major factor in making good mobile cinema is having coherent content. Each of the sequences experienced by participants made sense, and no clips contradicted one another. Although coherent stories were maintained, some people preferred more explanations and clearer narrative. One viewer expressed, “I was a little confused with who the characters were, until I saw more of the story, and eventually when I got the endings.” This could have been achieved

[58] All UbiComp evaluations happened within an intensive four-hour demonstration section, in which the participants might have been busy seeing other demos as well. At the MIT Media Lab, all participants had devoted enough time to testing.

with better introductions to each character. A better overall introduction clip, or "trailer" that provided back story could have helped as well.

In addition, both videography and music also play important roles. Based on viewers' reactions, one of the strongest features of *15 Minutes* was its acting and first-person perspective. Many participants were impressed with overall style of *15 Minutes,* and felt that it was well suited to the small scale of the PDA screen. The acting was engaging; one participant said, "The way the characters spoke directly to me caught my attention." Another said, "I really like the personal feel from the characters directly addressing you." Yet another admitted, "I was a little reluctant to try it out, but in the end I was really surprised with the quality of the video and editing. It was as good as watching a Hollywood movie at times. I was especially impressed with the action scenes."

Social interaction is different from expectation: When we designed the M-Views system 0.2, we expected to see people use the messenger to communicate with each other. In fact, people just talked to each other face-to-face, but the interactions patterns were slightly different. At UbiComp, participants shared their M-Views Presenter with other participants and people who just passed by. They compared story content, discussed about characters and plots, and laughed together. Many interactions were more informal and ad hoc. At Media Lab, several paired people came in and participated in this story together. They often had more structured ways to communicated with each other and discuss possible strategies in the middle of their tests. In many cases, people were interested in each other's endings.

Both the story content and technology design have affected the social interaction. *15 Minutes* is a very compact story, in which a participant can finish within 20 minutes. And the story locations are limited to a small place. There is no need for communicating with other people. Second, more participants were the first-time users of the M-Views system. They even didn't think about using messenger, which requires five minutes instruction as well.

Agency is needed: Still, there was the question of whether or not the viewer had real agency. One person said, "I did not get the feeling of agency. Although the story was entertaining, I'm not exactly sure how I was affecting the plot. The only interactive aspect I see is with walking around in order to access the clips. Perhaps a story that made it more obvious to the audience that they had to strategize by moving around would be the next step."

The biggest problems with the story were the assumptions the M-Views system made about the viewer. For example, at one point in the story, by simply visiting Michael's Office, it is assumed that the viewer "told" him that the elevator was still broken. There was no option whether or not to actually tell him; no real feedback mechanism. However, the story also supports the idea that Michael would know regardless, since he has security cameras set up in the building, and has planned for the elevator to fail as part of his diversion. Still, the participant influences the story only by his or her walking path, and not by any other input.

In summary, *15 Minutes* has provided a working test bed for us to know user's feedback. The compassion data between UbiComp and Media Lab, in particular, help us to understand user experience, story formals, and technology implementation. These understandings support me to reflect on the thesis hypotheses, design approaches, and story construction in the next chapter.

6.3 Reflection

In this section, I discuss three design principles for developing mobile cinema stories and reflect on the M-Views systems. What are the main differences in development strategies between conventional media and mobile channels? What recommendations can I provide to people who may be motivated to develop cinematic stories for mobile channels? What are the main lessons that have been learned from the two versions of the M-Views system? These discussions and reflections are presented from various perspectives: the creation perspective, the technical perspective, and content distribution perspective. These discussions and reflections may be helpful for story authors who want to create mobile cinema in the future.

6.3.1 Design Principles

In this section, I articulate three design principles that have been discovered through both content creation and system development: they pertain to the audience's situation and motivation, user mobile experience, story location, and production process.

6.3.1.1 Design for the Audience's Situation

The major question which every mobile cinema audience and author need to ask is, what is the motivation for walking around physical spaces to unfold stories? What are the most important benefits of walking around? According to our productions and evaluations, this question can be addressed by considering of the audience's situation.

The first type of audience situation is that the audience member knows where to go next and mobile cinema stories are additional information to augment her tourism experience. In other words, story authors design mobile cinema stories to adapt to the audience's plans, walking paths, and schedules. People like to experience stories about landmarks and places as they walk around—landmarks and places are visual, so a visual medium is appropriate. Many related tour guide systems (Abowd *et al.* 1997; Reinhard *et al.* 1999; Petrelli *et al.* 1999; and Cheverst *et al.* 2000), as well as *MIT in Pocket* and *Old Granary Burying Ground Ghost Tour*[59], fit this type of audience's situation.

For example, several audience members of *MIT in Pocket* expressed that this story had provided a window enabling them to see the daily life of ordinary MIT students, such as struggling with hard studying, dating stories, and hacking activities. These stories could not easily be found through a simple tourist experience. By interacting with these characters, they learned diverse aspects of MIT student life. In addition, physical buildings, such as Lobby 10, Lobby 7, and MIT Chapel, have very distinctive atmospheres, which helped M-Views systems to present stories and engage with the audience. Michael Epstein, a former student in the MIT Comparative Media Studies Program, developed *Old Granary Burying Ground Ghost Tour* and

[59] http://web.mit.edu/m_e/www/ghosts.html

presented it on Halloween 2003 in Boston. On that day, many people had plans to visit this place and Michael's mobile stories augmented the extraordinary atmosphere with Flash movies.

To design stories for this type of audience situation, the author first needs to know the audience's interests or schedules. What are possible walking paths? How long would the audience stay at one place? Which buildings would most audience members visit? By knowing this information, the author can design adaptive mobile stories that fit the audience's interests or schedules. In *MIT in Pocket,* two story authors were MIT tour guides, who had given dozens of tours to MIT visitors, such as prospective students and their families, visiting scholars, and business people. According to their tour guide experiences, these two authors suggested two versions of *MIT in Pocket*, a three-hour version and a whole day version. The three-hour version was specifically designed for tourists who spend less than a half-day at MIT. Second, the author needs to know the physical environments as well. All the story authors of *MIT in Pocket* were MIT students, who know MIT buildings, cultures, life, and even gossip. They crafted these stories from their personal experiences, memories, and feelings. In summary, to design to promote the *adaptive motivation* of the audience, story creators need to know both the audience and the physical environments.

The second type of audience situation is that the audience receives clues from mobile cinema stories and makes corresponding decisions about where to go next. A close analogy to this type of motivation can be found in location-based treasure hunt games. To play these games, the participants often have a very clear goal to achieve. They investigate all necessary clues, explore possible locations, and use logic to infer various possibilities. *Another Alice* employed a similar strategy for inviting the audience to participate in this mobile story. At the beginning of the story, the participant was instructed to meet Professor Eugenie at her office, where the participant met one of three characters. In the following sequences, each story character either told the participant explicitly where to go or gave her two options to pique her curiosity. In *15 Minutes,* the audience is motivated to walk around by the characters, actions, and plots in the story.

To proactively motivate the audience to walk around, the author first needs to create appealing stories. Why should the participant care about the characters? What kinds of information does the participant need to know, so that she can choose where to go next wisely? What kinds of information must be kept back at the beginning to raise the participant's curiosity? Story design is critical in this situation. Second, the author should also know the audience. Is the audience familiar with the story locations? Can they run fast enough? Knowing the participants can help the author not only design appealing content, but also build a smooth interactive story experience.

The major difference between the first type and the second type of audience situation is that the latter can drive the participant to go somewhere she has not planned to go. To motivate the audience, the author need not know the audience's schedule. For example, the audience members of both *Another Alice* and *15 Minutes* were willing to spend an indefinite amount of time on participating in stories.

Nevertheless, to design for either type of audience situation, the author needs to understand, develop, and keep story coherence; the author needs to create certain levels of curiosity at the right time and locations; and the author needs to test story experiences both on M-Studio and in the field.

6.3.1.2 Leverage Mobile Experience

The second design principle is to leverage mobile experience. What is mobile experience? What are possible implications of these mobile experiences and behaviors in terms of story presentation and design? Many great questions around this topic could be investigated, but in this thesis, I only reflect on five sub-topics that have been observed through story development.

The first, and the most obvious, mobile experience is that the audience walks around in physical spaces. They need to know where to stop to watch video clips on PDAs or cell phones and where to go after watching a video sequence. Because the audience often stands at a location and watches stories, each clip of mobile cinema is usually short. This clip length constraint could cause problems for the story author, who sometimes wants to present longer content. To deal with this issue, the authors of all three of our productions spent much time on designing dialogues, setting up production, editing sequences, and creating short messages to craft compact and appealing video content.

The second point is story density, which represents the number of story clips placed within a certain area. Story density is a relative concept. For example, if there are twenty clips of one mobile cinema story on the third floor of the Media Lab, the density of the story is relatively higher than another story that only has three clips on the same floor. Many factors could determine the density of a story: story content, story locations, the audience's motivations, and technology constraints. For example, in a museum, a story density could be very high because many art objects provide a rich story base and a wireless network could be set up perfectly. In contrast, in a city, the density of a fiction story may have to be loose, due to the distances among physical buildings and various wireless network issues.

The third mobile cinema experience is repeatability, which means that the audience can participate in the same story more than once. At UbiComp 2003*,* there was only one person who tried *15 Minutes* twice; but more than three people had participated in the same story twice at the Media Lab, where the story was originally created. I cannot draw a definitive conclusion about this difference, but can suggest the following ways to look at it. For example, the UbiComp participants were very busy exploring different projects during the demonstration section; and the three story locations didn't match the three production locations. In contrast, people who came to the Media Lab to participate in *15 Minutes* usually planned to spend a certain amount of time on this story. In addition, the match between the presentation locations and the production locations was more engaging to the audience.

The fourth point related to mobile experience is story presentations on small screens. People are used to watching and enjoying cinematic stories on big and wide screens with three-dimensional sound. What are appropriate production and presentation strategies for small LCD screens? In our three productions, the camera people used many close-up shots, or even extreme close-up, to capture and present action details. In particular, if a story clip was narrated from the first-person perspective, close-up shots drew the audience's attention very well. Furthermore, sound could also be a very useful source for crafting story atmosphere and giving hints to the audience. In *15 Minutes,* all music was composed after the story production; therefore, most music was matched to the visual presentation of the characters, actions, and plots. Good music could make the characters on small screens come alive.

The fifth point is about the communication among multiple participants. We have witnessed several cases, both at UbiComp 2003 and Media Lab, where audience members gathered to compare different story clips that they received. Mobile cinema experiences could be designed to become social activities, because different story experiences pull people together. Unlike our expectation of using wireless messenger, all social activities happened face-to-face. At UbiComp 2003, there were often more than two people gathered together to compare and discuss *15 Minutes;* at Media Lab, paired people compared their stories either in the middle of the story or at the end of evaluation to find out the different endings. This kind of social experience is very different from regular use of a mobile phone, which is a typical private device.

6.3.1.3 Choose Story Location Wisely

In mobile cinema, when the author thinks about or chooses story locations, she has to consider broad contexts. These contexts could consist of physical locations, story content, the audience's knowledge about story locations, and location detection technology.

Where to develop mobile cinema is often the first question that many creators may ask. Does this place have unique atmosphere? Is it a historical place, a dangerous place, or just a beautiful place? Would the audience like to visit here? Is this place already associated with many legendary stories? Before the production of *MIT in Pocket,* the authors had collected much information about MIT—its history, its famous people, and their stories. For example, two students spent several days at the MIT Museum to search through MIT archives; Professor Davenport also contributed her personal video collections about the stories from F&T Restaurant, which was one of the favorite dining places attracting MIT community members. Based on these pre-production investigations, the authors chose fifteen unique interesting places at MIT for this production.

The second question is about story content. Is the content strongly associated with the locations? Could the same story be deployed at other places? What are the relationships between the story elements and physical locations? Who are the characters? What are the conflicts and resolutions? Where would the characters hang around? How could the characters communicate with the audience? In *Another Alice,* the author designed the three main characters first, and then chose story locations based on these characters' positions and roles at MIT. For example, the pre-med student often eats at Walker Memorial and visits the MIT medical center; therefore, these places were selected. Whether to choose locations first or to design story content first really depends on the author. I have also observed that many authors used iterative processes to identify story locations and develop content details.

A critical design decision is about matching locations. Should the story presentation locations be matched to story production locations? In both *Another Alice* and *MIT in Pocket,* the authors matched the story presentation locations to the story production locations. In *15 Minutes,* the presentation locations could be different from the production locations. Several factors could affect this decision. For example, the type of motivation of the audience matters. If the audience has an existing walking plan, the matched locations would provide the audience much convenience. If the author decides to invite the audience to devote a fixed time to participating in her story, unmatched locations may also work out. A second factor would be the scalability, which means that the content creators want to duplicate the same mobile cinema at multiple locations. In particular, locations such as Six Flags or Starbucks have similar settings and physical characteristics. These places are great location candidates for deploying large-scale mobile cinema stories.

The third question is related to the audience's knowledge about these story locations. If the audience knows the story places very well, it gives the author much freedom to craft stories. In contrast, if the audience is unfamiliar with the story locations, the author has to take a different approach to eliminate participation uncertainty. Knowing the audience's location knowledge is not a simple task. In particular, many tour guide-like stories are developed for tourists. By default, most tourists are new to the story locations; and they can easily get lost. To solve this problem, several solutions may be useful. For example, at UbiComp 2003, we used both paper signs to indicate location names explicitly and a map to show the overall setting of the third floor.

The last, but not least, question that every mobile cinema author must ask is how accurate location detection data would be. Any story author should know where a location-detection technology works and where the wireless signal is too weak to work with. The author should be aware of technology uncertainty before any mobile cinema production. In many cases, mobile cinema creators don't pay enough attention to this issue, because conventional cinematic storymaking doesn't require any consideration of this type of issue. For example, we didn't know this issue before the production of *Another Alice,* and we didn't pay enough attention to this issue again during the production of *MIT in Pocket.* Many production works had already been done before we finished most location detection work. As a result, several places could not be detected, due either to physical environments or to discontinuous subnet roaming. We had to spend additional effort to either change the story productions or re-calibrate the story maps. Our suggestion to all mobile cinema authors is to detect locations first, and then design stories. Until an ideal location-detection technology is invented, wisely choosing story locations is a very practical approach to avoiding technology uncertainty. In the next section, I will discuss the details of this issue from a story production process perspective.

In summary, I have discussed three design principles: design for the audience's situation, leverage mobile experience, and choose story locations wisely. In the next section, I will articulate what I have learned about the M-Views system through story productions.

6.3.2 Reflection on the M-Views System

In this section, I review the system design and implementation of M-Views 0.2 in the context of story creation and distribution. The key question that guides the following discussion is: what are the advantages and constraints of the M-Views system 0.2 in mobile cinema story development? We use *MIT in Pocket* and *15 Minutes* as two main case studies to illustrate these reviews of the M-Views system. These reviews are classified into four categories: basic story creation tools, coping with uncertainty, design approaches, and content production and distribution.

6.3.2.1 Basic Story Tools

Basic story tools are always necessary for creating mobile cinema stories, no matter whether uncertainty is coped with or not. For example, in M-Studio, these basic tools include a storyline editor, a clip editor, a Location editor, and a Location viewer; in Presenter, both the streaming video player and the media server are basic tools; in M-Views Server, a web server manages different accounts, monitors the audience's activities, and records all log files. Mobile cinema is indeed a storytelling medium and these basic tools are foundations for creating and presenting stories. Based on the interviews with story creators and the feedback from the audience, several findings are summarized as follows.

M-Studio: Most mobile cinema authors, including students at both the IAP workshops, were able to learn the basic tools in M-Studio quickly. For example, the authors of *MIT in Pocket* said that it was easy to use and allowed for clear visualization of complex stories. Many authors felt the Location editor was useful, because the location-based authoring process suddenly became straightforward. "In particular, users responded positively to M-Studio's usability, finding the interface intuitive and usable with only minimal instruction. People were particularly fond of the easy to understand transition from locations on the maps of the location editors to locations on the storyboard. Authors first picked scene locations, and entered in text clips about theoretical scenes. Then, they could take pictures at those locations. Adding in these images created a richer picture of what a scene would look like" (Kastner 2002). The author of *Another Alice* compared the M-Studio-based authoring processes to the paper-based processes and pointed out that she would have made her story more complex, adding in more crossovers with M-Studio. In addition to the creation of original written features, M-Studio is also able to generate XML-based story scripts and this feature saves the author much authoring time.

M-Views Presenter and Server: The basic components of the Presenter include the messaging system, a streaming media player and server, XML database, and web administration. Most components worked quite well during the evaluations of *MIT in Pocket* and *15 Minutes,* except the slow wireless networks. Among the eighteen participants, seven of them found that their Presenter's screens were frozen occasionally in the middle of their experiments. The 802.11 wireless networks were still not fast enough to stream smooth video and audio to an M-Views Presenter. We discussed several approaches to solve this problem. For example, we could cache video content into local memories on the Presenter. Or we might find a better compression algorithm for streaming the same quality of video content with fewer bits. Although wireless streaming was not our main research focus, it indeed affected the overall performance of mobile cinema experiences.

6.3.2.2 Coping with Uncertainty

In addition to creating the basic components of the M-Views system, most of our development efforts have been devoted to coping with uncertainty. Many functions, such as Simulators, Storyflag, MapAgent, and the messaging system, have been used by mobile cinema creators in both *MIT in Pocket* and *15 Minutes*. These two production processes have shown that the most challenging part of developing the stories was designing appropriate story structures and designing story flag schemas to support these story structures. In the following sections, I outline the findings from using the M-Views system 0.2 to create mobile cinema stories and deal with various sources of uncertainty.

Simulations: Story creators have used various simulations intensively to imagine the audience's possible ways of participating and to examine story coherence. For example, Kastner (2002) describes how simulations helped the author to see potential issues of designing *MIT in Pocket.*

> *At first, the clips were divided into three time periods, morning, afternoon, and evening. However, within those periods, they needed to be laid out with respect to each other. For instance, one character's morning focuses on him going for a run. Since this takes place over several clips, they could not all happen at the same time. Furthermore, a user could not see the first clip of him tying his shoes in preparation for the run after seeing him running already. So, sequential time was used within each of the three major blocks to prevent seeing clips out of order. This became our initial prototype for simulation.*
>
> *Simulation raised several issues. One major point was causality. For instance, one of the final clips of the story involves one character returning a ring he found to its owner. It was discussed whether or not a user should be able to see this clip without having seen any of the previous clips pertaining to the ring. While all the scenes for this story were designed to work as standalone scenes, there are story arcs that travel through the story. The question became whether or not to enforce them. Also, there was the issue of the time buckets. A user who shows up late in a time division or turns off his or her iPAQ for a period of time can miss many clips. Also, the sequential time model was preventing some backtracking that did not result in incoherent storylines. Furthermore, simulation showed that if a user went to a location from the last time slot of a block, no more clips could be received until the next time block started. Clearly, a different model was needed.*

According to the authors of *MIT in Pocket* and *15 Minutes,* the major benefit of M-Studio was that the effects of these different structures could be easily simulated and visualized from the single desktop software. The author could quickly enter the basic story structures, change several parameters, and then be able to see simulation results. If the author would like to see a specific or extreme user scenario, M-Studio simulations are often able to help to present this scenario. For example, in *MIT in Pocket,* one author imagined that the participant went to class for one hour and wanted to know how this pause would affect overall story experiences. She added a relative time flag and created a simulation where the user turned the M-Views Presenter off for an hour. This simulation was also saved for testing other different story structures. "Although it is impossible for an author to predict all the different scenarios in which a viewer will use the device, being able to see the results for a given path is a powerful tool" (Kastner 2002).

Story flags: Story flags design directly affects mobile cinema story structures, story presentation, and audience participation. Appendix Two gives the example of the story design of *15 Minutes*. Much of their authoring efforts were devoted to creating and testing story flags. According to the authoring process and story evaluations, two sets of lessons are worthy to be shared. On one hand, designing story flags could be really time-consuming and complicated. For example, *15 Minutes* has twenty clips, and each of them is associated with multiple flags. In addition, there is a central flag table that is shared by all clips. If one flag's value is changed, many other clips that are sharing the same flag will be affected; therefore, designing appropriate story flags to maintain story coherence is a big challenge for the author. On the other hand, the story creators said that the basic concept of story flag is simple (each clip is associated with two types of flags that determine how this clip is presented and how other related clips will be presented), and is extremely flexible (it supports a variety of design requirements.) The current version of M-Views consists of numerical flags, list flags, clip flags, and date/time flags. With these types of story flags, a story designer is able to create a great variety of mobile cinema stories. In addition, M-Studio also supports the author to create new types of story flags.

Nevertheless, not all aspects of the mobile stories can be authored using flags; and not all aspects of physical experience can be simulated using simulations. For instance, the location detection engine may prove unreliable at some locations where events take place. For this situation, the creators have taken advantage of context-free text messaging to test for coherency and guide users in the proper direction. For example, if the user misses critical clips, the M-Views server might send a message to reestablish story continuity by providing information or clues about what to do next. In *MIT in Pocket*, this is done with the occasional “mock emailing” between characters, thus giving the audience a glimpse of where the fictional students will go next.

6.3.2.3 Tools and Story Design Approaches

Different people often have their own ways to craft stories, and the M-Views system 0.2 supports various authoring approaches and styles.

An original mobile cinema story concept usually comes from one or two people. In our three mobile cinema productions, *Another Alice* came from Christina Chen; the basic creation method of *MIT in Pocket* was taught by Glorianna Davenport; and David Crow wrote *15 Minutes.* Different people used the M-Views system at different design stages. For example, both Chen and Crow designed the main plots of their mobile cinema stories with conventional tools, such as pen, paper, and Post-Its. During the creation of *15 Minutes,* the M-Views 0.2 was primarily used for authoring and distributing content. In contrast, *MIT in Pocket* was an evolving, collective, and open mobile story. M-Studio was frequently used for structure design, flag setup, and experience simulation.

In addition, the M-Views system 0.2 has been used for authoring different story structures. For *MIT in Pocket,* all story content was classified into three story sections, which reflect a typical time framework of a student’s daily life. For *15 Minutes,* each storyline was built around the activities of a particular character. The entire story is time-dependent. That is, all the events unfold in identical story-time regardless of the path the user takes. Therefore, if two characters are supposed to meet, they must both be in the same place, at the same time, relative to the total story.

Finally, the M-Views system 0.2 supports various sensor data, such as spatial, environmental, temporal, and other sensor data. The flexibility of M-Studio architecture allows for easy expansion of context annotations based on a story author’s requests. For example, the system 0.2 can read the weather information of a particular location and use this information for authoring weather-based mobile stories.

6.3.2.4 Content Production and Distribution

Mobile cinema production indicates the whole course from an idea's inception to the formal release of a mobile cinema story. In conventional filmmaking, a production process basically consists of writing, pre-production, production, and post-production. In mobile cinema, a production process requires more steps and tasks, because of various sources of uncertainty. For example, the author needs to know where wireless signals are good; the author needs to walk around several physical locations to understand how the audience may interact with a mobile cinema story; or the author may use Post-Its to simulate multiple story threads. Are there any general guidelines for developing mobile cinema stories that an author may follow? I suggest the following two guidelines:

> *If both technology and participation uncertainties can be eliminated, the author should focus on developing stories first.*
>
> *If there are too many sources of uncertainty, the author should investigate them before any story production.*

The first guideline illustrates an ideal situation, which can be archived if the author or distributor is willing to invest in sufficient infrastructure. For example, if Disney World wants to create mobile cinema for its theme parks, they can deploy reliable wireless networks or re-create physical settings, so that most technology uncertainty can be turned into certainty. In addition, they can ask their visitors to walk through particular paths to participate in mobile cinema stories. Under these kinds of conditions, the author has much freedom to focus on character development, plot design, production, and post-production.

However, most mobile cinema authors have to deal with many sources of uncertainty before they develop any stories. In the case of developing *15 Minutes* at UbiComp 2003, the wireless network was established and controlled by the conference organizers. Some network setups were unable to be changed. Therefore, we had to use MapAgent to detect the strengths of wireless signals at the third lobby of the conference center. In addition, we also paid much attention to various sources that could affect the audience's participation. What would the traffic flow be? How long would people stay at the poster section? Could other people's wireless access points interfere with our map calibration? In accordance with these investigations, we modified our story scripts to fit the physical environment. In the process of developing *MIT in Pocket*, dealing with uncertainty and developing mobile cinema stories took place in parallel. Various means, such as Post-Its, whiteboards, M-Studio, and paper games, were employed to simulate, feel, and understand participation uncertainty during story brainstorming sessions. We also used the messaging framework of M-Views Presenter to simulate a variety of participatory activities.

The last, but not least, technical reflection is on content distribution. The current system design and implementation have been done based on the standard 802.11 wireless networks (*MIT in Pocket* runs on the MIT wireless networks and *15 Minutes* operates on the MIT Media Lab wireless networks). There are other options of choosing different types of wireless networks, such as cellular networks and P2P local wireless networks. Although the major advantages and constraints of using 802.11 wireless networks have been discussed in previous sections, I would like to reflect on the following two points, which have been discussed by the M-Views team.

Network bandwidth and reliability: the first question about choosing a wireless network is how fast it is. Mobile cinema demands much more bandwidth than other voice, message, or picture-based mobile applications. People don't like technical problems, such as frozen video presentation. Therefore, bandwidth and network reliability are always key questions to be addressed.

User behavior and story structure: sometimes, a wireless network is fast enough and reliable, but it is still not suitable to certain types of mobile cinema stories. For example, the 802.11 wireless networks are difficult to work with if their subnets are not connected in a seamless way. In *MIT in Pocket,* we had to carry a laptop server to solve this problem. In addition, if the author wants to create a mobile cinema story for a city, current cellular networks may be the only solution. The author may have to cache many video clips on local memories.

Content distribution is also an evolving topic, because new wireless technologies are being created rapidly. Content distribution is also an interdisciplinary topic, because it requires a range of know-how in technical, creation, and business domains.

6.3.3 Summary

In this chapter, I have described the two mobile cinema stories (*MIT in Pocket* and *15 Minutes*), the production processes, and the evaluations. These production processes and evaluations have led the overall reflections on this five-year research project. I proposed three design principles and discussed the M-Views system in conjunction with two story productions. In the next chapter, I will conclude this thesis with discussion of possible future work and my contributions to this field.

7.0 CONCLUDING REMARKS

7.1 Future Work

In this section, I suggest the following four research domains, which deserve future investigation.

7.1.1 New Ways to Represent Story Structures

The current story structure representation schema is the flag mechanism. As we discussed in previous chapters, the advantage of using flags is its flexibility; and the disadvantage is that an authoring process could be time-consuming. Although a mobile cinema story can be created using only the M-Studio visual interface, an actual authoring process often requires a deep understanding of the flag mechanism. What would be alternative processes to create flags? One possible creating approach is learning-by-example. For instance, suppose a new mobile cinema creator loves *15 Minutes* and wants to create another similar fictional mobile cinema story for his apartment. He can inherit the flag setups of *15 Minutes* and modify them for his stories. By doing so, the author doesn't need to understand all flag tags, but instead can just focus on the features that he needs.

Kastner (2002) also suggested an alternative tool for generating flags.

> *Presently, flags can only be generated on explicit structural elements, like relative ordering, crossover, and location. It would be more desirable to be able to generate structure based on narrative elements within the story. To do this, users could annotate their clips with semantic information. This information could be used to imply connections between clips that do not have any obvious relation in the physical story structure. For instance, in MIT in Pocket, if each clip focusing on the ring was annotated as such, the tool could suggest flag connections between those clips. A rule could then be created to enforce an ordering amongst clips that refer to the same plot point. To further extend this idea, the semantic annotations could be combined with a common sense reasoning tool, allowing M-Studio to deduce structural elements, like that finding a ring would have to precede returning a ring. This would take some of the burden off the author in determining how to arrange flags to express such connections.*

Another way to improve the flag system is to know mobile behaviors earlier and more accurately. For example, a new version of M-Views Presenter could learn users' interaction patterns over time and provide the author with these patterns, so that M-Studio provides a preliminary flag setup for the author, who can use these flags as a basic authoring framework, and modify them for her story content and structures.

Furthermore, M-Studio could employ multiple story structure representation schemas for different types of stories. At the end of authoring, all story structures can be exported to the same XML-based story scripts. From the author's point of view, she will have different choices to construct her stories; from the M-Views Server's point of view, all story scripts are based on the same XML schema. As a result, the same M-Views infrastructure could support more people to create mobile cinema stories.

7.1.2 New Ways to Detect, Represent, and Model Uncertainty

Although much of this thesis research is about coping with uncertainty in mobile cinema, I have not implemented a systematic mechanism for detecting, representing, and modeling various sources of uncertainty.

The current M-Views system provides various tools for the author to cope with uncertainty, but the system has no quantitative measurements for modeling uncertainties. For example, what are practical methods for representing various sources of participation uncertainty? Can we rate these uncertainties? How does the computer monitor and visualize various uncertainties? If we can detect, represent, and model the sources of uncertainty, we may be able to build much smarter mobile cinema systems. For example, a mobile cinema story is to be deployed in a three-floor building. The mobile cinema system can detect that ninety-five percent of users would start this story from the first floor and only less than five percent of participants would go to the third floor first. The system can suggest this kind of predication data to the author, who can be aware of these user experiences and focus on the story design for the majority of the users.

Many of the sources of uncertainty have been identified through story simulation, field tests, and story productions; and much identification has been done in ad hoc ways. In the future, I want to see a systematic mechanism for detecting, representing, and managing various sources of uncertainty. For example, a new version M-Views system may remember all sources of uncertainty in *15 Minutes* and save these data into an "uncertainty folder," which could be used by other mobile cinema productions. For another example, a new version of MapAgent will be able to show overall signal strength, which suggests a new way to re-arrange access points. By doing so, the author would understand and cope with uncertainty systematically; As a result, more reliable, cost-effective, and scalable mobile cinema stories would be created.

7.1.3 Networked Context-aware Camera

With the current version of M-Views system, the production of a mobile cinema story is still very time-consuming, compared to conventional linear story production. One major reason is that the authors had to use a set of conventional production tools, such as digital video cameras and non-linear editing systems, to make video. In addition, the author had to manually annotate all contextual information for each clip. To solve these problems, about two years ago, we started another project, *Participatory Networked Camera (PNC).*

The existing version of *PNC,* which is based on a Pocket PC PDA, consists of three basic features: a capture module, a shared archive, and a messenger.[60] In the future, a new version of *PNC* aims to simplify the current production process of mobile cinema. It would have the ability to increase group awareness, enhance group communication, and help the creators to cope with various production situations efficiently. Several critical functions need to be developed. For example, a new version of *PNC* would feature an editing tool, an interactive map, and a communication and collaboration environment.

The editing tool will allow the author to edit images that are imported from both her video archives and other camera people's footages. The author will be able to drag clips from an image section (based on their thumbnail images and text information) and drop them onto an editing section, where she can play them for reviewing purposes, set in and out points on each clip, link the clips together, or unlink the clips.

The interactive map will be able to display information about each author's position, orientation, and recent locations. Each author will have an active spot. The transparent spot in each person's field will indicate the direction of motion of the cameraperson, and a red circle can indicate whether or not the cameraperson is filming. The task panel will be used to create, view, and manage both individual and group tasks. Furthermore, the interactive map can automatically generate basic contextual information, such as location and time, for each video clip. This contextual information could be exported to M-Studio automatically in order to save authoring time.

The communication and collaboration environment will provide a set of tools for group-based mobile cinema production. For example, the author can delete a task after she has filmed and saved the corresponding clip. She can look at the open tasks and begin work on them, or she can create a new task by tapping and holding the message line to bring up a menu relevant to the specific task. She can make new tasks, delete one, edit the information in a task, or message the owner of a task. The author can also keep track of tasks in a sophisticated editable production document/forum.

In summary, the main development approach to developing *PNC* cameras is to integrate multiple production functions into one easy-to-use camera. This specifically designed camera for mobile cinema productions will allow more people to create stories for themselves and for communities.

7.1.4 New Story Genres

[60] The capture interface is entirely for the current operator of the camera. The video display shows the scene that the camera is currently viewing. There are the standard record, stop, and play buttons, and scroll bar. Besides those buttons, there are *Save*, *Delete*, and *Upload* buttons. *Save* can be used to save not only Motion-JPEG files, but also the contextual annotation (where, when, and what) of the clip, into the *Archive* folder on *PNC*. The shared archive allows a team of people to view and share other people's video footage with permission. Each clip is associated with a basic text description and contextual annotation. The messenger is basically a chat room, by which the camera people can share messages to discuss production or other concerns. These messages can also be converted to tasks to be accomplished.

The fourth future research domain is to explore different genres of mobile cinema stories. In addition to fictional mobile cinema stories, such as *Another Alice* and *15 Minutes,* what are other possible story genres? The current M-Views Presenter, Server, M-Studio, and the flag mechanism are designed generally enough to support a variety of genres of stories. For example, the M-Views system can certainly support documentary mobile cinema stories, because many physical environments have already provided distinctive story atmospheres. The M-Views system also supports mobile games, which have clear goals and rate the player's achievements in real time. The heuristic mechanism can be converted to an activity recorder for mobile games. The M-Views system also supports various interactions between the participant and objects. For example, if the participant walks close to a copier machine, a special mission list can be printed out for her, so that the participant would feel strong agency in the mobile cinema.

Our three mobile cinema productions are only proofs of concept, working demonstrations, and case studies for developing the M-Views systems. I hope that self-motivated story creators will be eager to explore many styles of mobile cinema; and I believe that numerous novel research areas, applications, and design projects could be invented, based on the past five years' development, to enrich communication, expression, and human experience.

7.2 Conclusion

This thesis research has been initiated based on the passion that my supervisor, numerous team members, and I had. Our original intention was to explore mobile media for storytelling and making, and these explorations have led us to develop two mobile cinema systems and three mobile cinema stories.

This system development and story production have shown that creating mobile cinema was an extreme challenge, because we didn't know too much about this new type of medium. We were not sure how mobile devices and channels would be different from conventional desktop-based or Internet-based media. What were the benefits that the audience would be able to have via mobile channels? What would be the main obstacles to making and sharing cinematic stories for mobile devices?

These questions were gradually addressed while we realized that mobile cinema is indeed involved with various sources of uncertainty. In this thesis, I have defined two types of uncertainty:

> **Technology uncertainty:** What the mobile cinema system presents may not be what the creator intends.
> **Participation uncertainty:** What the audience does may not be what the creator expects.

These uncertainty problems, which are also associated with previous computational storytelling, must be addressed in mobile cinema. Otherwise, these uncertainties would lead directly to two possible negative consequences: (1) the audience experience becomes cumbersome, confusing, and unengaging; or (2) the cost of producing coherent mobile cinema content becomes extremely high. In order to build a successful mobile cinema system, which must be able to help create and present reliable and coherent cinematic stories at a relatively reasonable cost-benefit ratio, I proposed the research hypothesis:

> A mobile cinema system supporting better location detection technologies, new authoring tools for mobile channels, responsive story presentation mechanisms, and creative story production strategies can help cope with these uncertainties.

This hypothesis has guided me to do related research. In particular, I have learned numerous lessons from oral storytelling, participatory theater performance, interactive narrative, context-aware computing, and uncertainty research. Meanwhile, the first mobile cinema prototype, M-Views system 0.1, and the first mobile cinema story, *Another Alice,* were developed and have taught me practical and tangible lessons. Based on these related research lessons and prototype development, my team and I have built the M-Views system 0.2 and two more mobile cinema stories: *MIT in Pocket* and *15 Minutes.*

The original contributions to the interactive narrative fields are as follows:

This thesis examines the underlying requirements for coherent mobile narrative and explores two particular challenges which must be solved in order to make a reliable and scalable stream of content for mobile cinema: technology uncertainty (the fact that what the mobile cinema system presents may not be what the creator intends) and participation uncertainty (the fact that what the audience does may not be what the creator expects). The identification of these two types of uncertainty provides not only a measurement for exploring, developing, and evaluating mobile cinema, but also a language with which engineers, designers, and participants can communicate efficiently.

The exploration and analysis of these challenges involved prototyping two versions of the M-Views system for mobile cinema and three prototype cinematic narratives. Small user studies accompanied each production. The iterative process enabled the author to explore both aspects of uncertainty and to introduce innovations in four key areas to help address these uncertainties: practical location detection, authoring tools designed for mobile channels, responsive story presentation mechanisms, and creative story production strategies.

A major contribution of this thesis is three design principles for mobile cinema developments: design for audience situation; leverage mobile experience; and choose story locations wisely. These principles discuss audience motivation, story location, production approaches and process, mobile behavior, and content distribution. These three design principles have been discovered through the five-year investigation into both system development and story production and provide practical groundwork for the future evolution of mobile cinema.

In this thesis, I also discuss a parallel development framework for inventing a novel storytelling medium, mobile cinema. This framework is effective for both creating original storytelling forms based on the current wireless communication revolution and developing media systems based on a variety of design requirements for creating, presenting, and distributing mobile cinema content.

In addition to exploring, developing, and evaluating mobile cinema, I also hope this research will contribute to various creative, technological, and business domains, such as mobile touring, learning, entertainment, mobile computing engineering, and human-computer interaction. I hope that we can leverage the rapidly increasing mobile devices and channels to help people express themselves, learn, and live better.

Appendix One: XML Story Flags[61]

In XML, tags represent flags. The tag names correspond to the flag names. Each of these tags has two fields, the expression/operation field, which corresponds to the action taken on the flag, and the value field, which corresponds to the value used in the requested comparison or operation. All flag tags must end with a closing backslash.

Numerical Flags: The most basic flags correspond to integer values. By default, basic mathematical operations and comparisons are supported. If any expressions are preceded by a ! (logical not operator), they are negated. The value field should always be an integer.

Expression	Description
Comparisons	
==	equals
!=	not equals
#LT	less than
#GT	greater than
#LTE	less than or equal to
#GTE	greater than or equal to
Operations	
+	increment
-	decrement
/	divide by
*	multiply by

Table 20 Default numerical expressions/operations

List Flags: The server also supports flags that perform list operations. These lists can be tested to see if they contain a string or list of strings. They are updated by adding and removing values from the list. The value field for both comparisons and operations should be of the form of comma delimited values, e.g. `value="string1,string2,string3"`.

Expression	Description
Comparisons	
$=	List contains
!$=	List does not contain
Operations	
$+	Add to list
$-	Remove from list

Table 21 Default list expressions/operations

Clip Flags: Clip flags can be used in the requirement clause to test if certain clips have or have not been seen. Because they refer to the list of clips that have been seen, rather than using a specific tag name, they use the null tag (an underscore). The value of a clip flag should be a string corresponding to the tag name of a single event being referred to. Multiple clips can be

[61] This appendix is contributed by Carly M. Kastner (2002).

referenced by using <AND>/<OR> blocks (described later). The form of a clip flag tag should look something like this:

```
<_ expr=”?S” value=”ID1”/>
```

One special tag is used to express the fact that no clips can have been seen for this clip to play:

```
<_ expr=”!?S” value=”-1”/>
```

Expression	Description
?S	Clip must have been seen
!?S	Clip must not have been seen
?LS	Clip must have been the last clip seen
!?LS	Clip must not have been the last clip seen

Table 22 Clip flag expressions

Date/Time Flags: There are XML formats for both absolute and relative time flags. They can only be used as requirement flags. Again, since they do not refer to the actual flag table but other values, the null operator is used.

Absolute Date/Time Format: The value field is of the form D<MM/DD1-MM/DD2>, T<HH:MM AA1-HH:MM AA2>, where the first item is the span of dates (in month/day format) and the second is the span of times. If only date or only time is used, only one item should be used, but it should still be preceded by a D or T. An example:

```
<_ expr="?DT" value="D01/01-02/02, T9:00 AM-5:00 PM"/>
```

Relative Time Format: This flag expresses the minimum and maximum time that can have passed since a previous clip has been seen for this clip to play. Its value string should have three entries: the ID of the clip being triggered from, the minimum amount of time that has to pass for the clip to play, and the maximum amount of time that can have passed for the clip to play. All times are in minutes. If there is no limit to how much time can pass, then the maximum amount of time should be –1. This example shows that between 5 and 10 minutes should have passed since ID1 played to see this clip:

```
<_ expr=”?RT” value=”ID1,5,10>
```

<AND>/<OR> Tags

Sometimes an author might want to make a clip trigger if any one of a group of flags is true. However, by default, all flags listed under the requirement clause must be true for a clip to play. By nesting flags inside <OR> tags, the author can specify that the requirement is met if any of these flags evaluate to true. <AND> tags can be nested inside <OR> tags to allow for more control over statements. For instance, the following expression states that either flag A or both flags B and C must be true for this clip to play. Clearly, <AND>/<OR> tags are not legal within the results clause.

```
<OR>
   <A .../>
 <AND>
   <B.../>
   <C.../>
 </AND>
</OR>
```

Appendix Two: *15 Minutes* Story Scripts[62]

In *15 Minutes*, each clip has a set of story variables. The value of each variable is changed whenever the viewer enters a location. Each clip also requires that in order to view it, some variables must be at a certain value, or within a range of values. The plot is thus determined by the sequence of locations visited.

There are 4 major story variables. 3 are named after the main characters, "David," "Eve," and "Michael." A fourth variable "Fight" helps determine which climactic fight sequence and ending the viewer will see. When the viewer moves around the space, he/she is affecting the goals and motives of each character. The values of the character variables represent what stages they are at in their personal plans.

At the beginning, the values for each character are set at zero. For example, if you choose to go to Michael's office first, the Michael variable will be set to value = 1. In order to see certain clips, the Michael variable may have to be greater than 1, but not necessarily a set value. Using ranges in the requirements for many clips allows for greater freedom and variety of story possibilities.

Variables can have other kinds of values, not just numeric. *15 Minutes* also uses true/false constraints. It also has time constraints that make a clip accessible a certain amount of time after the experience has begun, or relative to another clip. The variables were easily defined using M-Studio, which automatically generates the XML script.

A Summary of Each Clip

Table 23 is a description of the 20 clips that comprise *15 Minutes.* The key story variables, values, and significance of each clip are explained.

Story script of *15 Minutes*	
ID0	**Introduction**
Location	Anywhere
Subject	Welcome to the next 15 Minutes
Message	Press the lower right button on the iPAQ to view video clips attached to each M-Views message. Have Fun!
Action	This is a brief "trailer" clip that is received by all players when they begin. It also sets the story variables at the following values: David value = 0 Michael value = 0 Eve value = 0 Fight value = 0
ID1	**The elevator is broken**
Location	Elevator
Subject	Elevator's broken...

[62] The *15 Minutes* story script was designed by Lilly Kam and David Crow.

Message	The elevator repairman is here.
Requires	David value = 0
Sets	David value = 1 Michael value = 1
Action	This clip introduces David. The elevator appears to have some power, but is still broken. David tells the viewer to tell Michael if he/she sees him. The story will automatically assume this transfer of information the next time the viewer visits Michael.
Significance	Unknown to the first-time player is that David is an undercover officer. Also, the unusual surge of power is actually a signal that Chi, the hitman is about to strike. Chi was hired by Michael to create a diversion so that Michael could steal a top-secret information disc. The clip requires that David value = 0, since the player has not met him before. After viewing the clip, it advances the value to 1. It also sets the Michael value to 1. If you have not already met Michael, this will affect the way you are introduced to him.
ID2	**Meet Michael**
Location	CEO Office
Subject	Meet the Boss
Message	Meet the CEO of Infosafe
Requires	Michael value = 0
Sets	Michael value = 1
Action	The viewer receives this introduction of Michael if he/she has not yet seen David. Here, Michael asks the viewer to go to the elevator to check up on the repairman.
Significance	Unknown to the viewer, Michael is waiting for the hitman to strike.
ID3	**Michael suspects**
Location	CEO Office
Subject	Michael suspects
Message	Michael suspects something may happen tonight...
Requires	Michael value = 1 David value = 0
Sets	Michael value = 2 David value = 2
Action	Michael reveals that a crime may be taking place tonight. The clip also shows Michael monitoring two security cameras on his laptop, with views of the elevator lobby and copy machine.
Significance	This clip appears if the viewer has met Michael, but still has not met David. The David and Michael values are advanced to 2, which reflect the fact that the viewer now has an important insight.
ID4	**Meet Eve**
Location	Copy Machine
Subject	Hello, Stranger
Message	Meet one of InfoSafe's finest.
Requires	Eve value = 0
Sets	Eve value = 1
Action	This clip introduces Eve, who appears to be Michael's secretary. She is making photocopies for him, which are late.
Significance	It is unknown to the first-time player that Eve is actually a spy whose mission is to prevent Michael from stealing the disc. Although Eve acts as if she has never met the viewer before, one of the story's possible endings is the revelation that that viewer has also been a spy, Eve's partner in the mission. Eve and the character played by the viewer have been acting for the security cameras.

ID5	**A flashback**
Location	Elevator
Subject	A Memory...
Message	David recalls a friend.
Requires	David value = 1
Sets	David value = 1
Action	David is looking at his pocket watch, which reminds him of someone from his past. A flashback sequence shows David with Chi, who used to be a close friend.
Significance	This clip does not change any values because it does not affect the characters' actions in any significant way. However, it provides interesting backstory to the upcoming fight sequence.
ID6	**Michael gets ready to leave**
Location	CEO Office
Subject	Michael awaits
Message	What good is power if you can't use it?
Requires	David value > = 1
Sets	Michael value = 3
Action	Michael gets ready to leave his office. The viewer gets this clip as long as he/she has seen David and knows the elevator is still broken. It does not require that the viewer has met Michael before, because time may have been spent watching Eve or David. Since the viewer knows about the power surge in the elevator, this clip assumes that the viewer told Michael about it.
Significance	Here, Michael is leaving because the power surge is a signal that the diversion is about to happen. He will soon try to steal a disc of top-secret information. Michael's value of 3 represents this stage of his plans.
ID7	**Eve leaves for Michael's office**
Location	Copy Machine
Subject	A Favor
Message	Eve needs a favor.
Requires	Michael value = 3
Sets	Michael value = 4 Eve value = 3
Action	The viewer receives this clip if they visit Eve after seeing Michael's plans to leave his office. The clip assumes that the viewer tells Eve about this. Eve says that she will use this opportunity to drop off some papers in his office, and asks the viewer to stay behind and watch the copy machine for her.
Significance	Eve is really leaving to stop Michael from stealing the disc, which is represented by her value of 3. Michael's value is advanced to 4 because Eve is now in pursuit of him.
ID8	**Eve suspects**
Location	Copy Machine
Subject	Eve suspects
Message	Eve senses trouble
Requires	Eve value = 1
Sets	David value = 2 Eve value = 2
Action	Eve is suspicious of the security cameras set up throughout the building. She heard a rumor that a robbery would occur tonight, and she asks the viewer to check around the building to see what is going on
Significance	This clip is another way of hinting to the viewer of the events to come. The viewer now suspects a robbery to occur tonight, which will affect the way David next reacts to him or her, thus his value is set to 2.
ID9	**David gives a warning**

Location	Elevator
Subject	You may know too much...
Message	You may know too much for your own good…
Requires	David value = 2
Sets	David value = 3
Action	The viewer appears to know that there will be a robbery attempt tonight, so David reveals that he is an undercover officer. He warns the viewer to stay out of the lobby, since things could get dangerous.
Significance	This clip sets David's value to 3, which affects the way Eve reacts to the viewer next.
ID10	**Eve's suspicions confirmed**
Subject	Eve knew it.
Message	Eve knew something was going on.
Requires	David value = 3 Eve value > 0
Sets	Michael value = 4 Eve value = 3
Action	The viewer tells Eve what he/she learns from David's warning. She goes to Michael's office to drop off some papers.
Significance	At this point, Eve assumes that Michael is on his way to steal the disc since David is getting ready for a confrontation. She will try to stop him.
ID20	**Printouts**
Location	Copy Machine
Subject	Something is printing...
Message	What does the printout say?
Requires	Eve value = 3 Michael value = 4 Last seen clip = ID7 or ID10 (Eve left for a while) over 1 minute ago.
Sets	Michael value = 4 Eve value = 3
Action	The viewer sees this clip only if he/she has just seen Eve leave, and stays in the same location for over a minute to watch her photocopies. As Eve is away, a nearby printer starts to print a list of names. Eve's name is on the list.
Significance	The people on the list are secret agents of the United States. This is actually the top secret information contained on the disc that Michael is after.
ID11	**Michael unconscious**
Location	CEO Office
Subject	Michael indisposed
Message	Michael's not too pleased...
Requires	Michael value = 4 Eve value = 3
Sets	Michael value = 5 Eve value = 4
Action	Michael is lying on the floor of his office. He was knocked unconscious but is waking up. He seems to have lost something, and is extremely irritated. He hurries out of his office.
Significance	Michael was knocked down by Eve, who has taken the disc. Michael's value is changed to 5, meaning he has left his office a second time. Eve's value is 4, meaning she now has the disc.
ID12	**Eve returns**
Location	Copy Machine
Subject	Eve returns

Message	Eve is grateful for your help...
Requires	Eve value >= 3 Michael value > 3
Sets	Eve value = 5
Action	Eve returns and thanks the viewer for watching her photocopying job. She slips the viewer her card and phone number.
Significance	Eve has actually slipped the viewer the disc she took from Michael. As her partner, the viewer will help her hide it from the police. Eve's value is now 5, indicating that the viewer carries the disc, whether he/she knows it or not.
ID13	**Fight Sequence 1 – Michael must be stopped**
Location	Elevator
Subject	Watch out!
Message	David gets a visit from an old friend...
Requires	Michael value < 5
Sets	Fight value = 1
Action	The elevator door opens to reveal Chi, the hitman. Chi and David fight. In a flashback, Chi reveals that Michael hired him to create a diversion, and they have been set up. David tells the viewer to hurry to Michael's office and stop him.
Significance	This is one of two possible fight sequences in the story. The viewer sees this clip if Michael has not left his office again, and his value is less than 5. The clip also sets a new variable "Fight" to value = 1, which will directly affect the endings.
ID15	**Michael is stopped and the disc is retrieved**
Location	CEO Office
Subject	Just in time...
Message	You're just in time!
Requires	Fight value = 1 Michael value <= 3 Eve value is < 3 Viewer has seen ID13 - Fight 1 within 4 minutes
Sets	Fight value = 2
Action	The viewer arrives at Michael's office, just in time to stop him from escaping with the disc. Michael aims a gun at the viewer, but David arrives to arrest him.
Significance	To get this happy ending, the viewer must have arrived within 4 minutes of seeing Fight 1. Otherwise, he/she would be too late and Michael would have escaped, and the viewer gets nothing. Eve's value has to be less than 3, and Michael's value can be no greater than 3. This ensures that Michael still has the disc, and Eve never got the chance to stop him. The fight value is changed to two, meaning the disc's whereabouts are known.
ID16	**Michael got away with the disc**
Location	Elevator
Subject	Too late...
Message	Why didn't you try to stop Michael?
Requires	Fight value = 1 Viewer saw ID13 - Fight 1 over 4 minutes ago
Sets	Fight value = 2
Action	The viewer gets this ending if they were too slow in stopping Michael. Michael and the disc are gone.
ID17	**Michael is unconscious, disc is missing**
Location	CEO Office
Subject	Just in time?
Message	Just in time?

Requires	Michael value < 5 Eve value >= 3 Fight value = 1
Sets	Fight value = 3
Action	The viewer finds Michael unconscious on the floor. He does not have the disc.
Significance	The viewer gets this ending if Michael has been knocked unconscious by Eve, who has taken the disc. However, he must not have regained his consciousness as in ID11. The fight value is set to three, which means Eve has the disc. David is puzzled because "Michael has no secretary."
ID14	**Fight sequence 2 – Michael interrupts**
Location	Elevator
Subject	Look out!
Message	David is visited by an old friend...
Requires	Michael value >= 4
Sets	Fight value = 3
Action	Here, Michael interrupts David and Chi's fight. He has lost the disc, and believes either they or the viewer has taken it. He aims a gun at the three, but is stopped by police officers.
Significance	This clip requires Michael to have lost the disc. It also sets the Fight value to 3.
ID18	**Michael is arrested, the disc is missing**
Location	Elevator
Subject	The End?
Message	Something doesn't add up...
Requires	Eve value >= 4 Fight value = 3
Sets	Fight value = 3
Action	The police have arrested Michael, but the disc is nowhere to be found. David is puzzled because "Michael has no secretary."
Significance	This ending follows the second fight scenario.
ID19	**Rendezvous with Eve**
Location	Elevator
Subject	Congratulations...
Message	You've done well…
Requires	Eve value = 5 Fight value = 3 Viewer saw ID14 – Fight 2 over 4 minutes ago.
Sets	Fight value = 4
Action	The viewer meets with Eve after the police are gone. She reveals where she hid the disc on the viewer, who has also been a spy all along.
Significance	The viewer only gets this clip if Eve was able to slip him or her the disc in ID12 – Eve Returns, setting the Eve value to 5. The viewer must also have waited in the elevator lobby long enough after the fight sequence to meet with her. Finally, it sets the fight value to 4, making it the ultimate ending.

Table 23 Story script of 15 Minutes

Table 24 is the XML-based story script of *15 Minutes,* which runs on the M-Views 0.2 server.

XML script of *15 Minutes*


```
<?xml version="1.0" ?>
- <FifteenMinutes>
  - <IDX Context="Idle" Date="" From="M-Views" Repeatable="false" Story="FifteenMinutes"
       Subject="Welcome to the next 15 Minutes" To="">
      <Requires />
    - <Message>
        <Text>Press the lower right button on the iPAQ to view video clips attached to
           each M-Views message. Have Fun!</Text>
        <MediaURL>:http://keaton.media.mit.edu:8080/Media/00Intro.mpg</MediaURL
           >
      </Message>
    - <Results>
        <David_value oper="=" value="0" />
        <Michael_value oper="=" value="0" />
        <Eve_value oper="=" value="0" />
        <Fight oper="=" value="0" />
      </Results>
      <Heuristics />
    </IDX>
  - <ID6 Context="STEVEN-SPIELBER ML(525,700,40)" Date="" From="M-Views" Repeatable="false"
       Story="FifteenMinutes" Subject="A Favor" To="">
    - <Requires>
        <Fight expr="==" value="0" />
        <Michael_value expr="==" value="3" />
        <_ expr="?LS" value="ID20" />
      </Requires>
    - <Message>
        <Text>Eve needs a favor... Will you oblige?</Text>
      </Message>
      <MediaURL>:http://keaton.media.mit.edu:8080/Media/07EveLeaves.mpg</MediaURL>
    - <Results>
        <Michael_value oper="=" value="4" />
        <Eve_value oper="=" value="3" />
      </Results>
      <Heuristics />
    </ID6>
  - <ID22 Context="STEVEN-SPIELBER ML(525,700,40)" Date="" From="M-Views"
       Repeatable="false" Story="FifteenMinutes" Subject="Something is printing..." To="">
    - <Requires>
        <Eve_value expr="=" value="3" />
        <Michael_value expr="==" value="4" />
        <Fight expr="==" value="0" />
        <_ expr="?LS" value="ID6" />
        <_ expr="?RT" value="ID6,1,-1" />
      </Requires>
    - <Message>
        <Text>What does the printout say?</Text>
        <MediaURL>:http://keaton.media.mit.edu:8080/Media/15Copier.mpg</MediaU
           RL>
      </Message>
    - <Results>
        <Michael_value oper="=" value="4" />
        <Eve_value oper="=" value="3" />
      </Results>
      <Heuristics />
    </ID22>
  - <ID10 Context="STEVEN-SPIELBER ML(525,700,40)" Date="" From="M-Views"
       Repeatable="false" Story="FifteenMinutes" Subject="Eve returns" To="">
    - <Requires>
        <Fight expr="==" value="0" />
        <Michael_value expr="==" value="4" />
        <Eve_value expr="==" value="3" />
        <_ expr="?RT" value="ID22,1,-1" />
        <_ expr="?LS" value="ID22" />
      </Requires>
    - <Message>
        <Text>Eve is grateful for your help...</Text>
      </Message>
      <MediaURL>:http://keaton.media.mit.edu:8080/Media/11EveCard.mpg</MediaURL>
    - <Results>
        <Eve_value oper="=" value="4" />
      </Results>
      <Heuristics />
    </ID10>
```

```
- <ID23 Context="WONG ML(720,820,25)" Date="" From="M-Views" Repeatable="false"
      Story="FifteenMinutes" Subject="Just in time?" To="">
  - <Requires>
      <Michael_value expr="==" value="4" />
      <Eve_value expr="==" value="5" />
      <Fight expr="==" value="1" />
      <_ expr="?RT" value="ID12,0,4" />
    </Requires>
  - <Message>
      <Text>Just in time?</Text>
      <MediaURL>:http://keaton.media.mit.edu:8080/Media/20MysteryEnding.mpg<
          /MediaURL>
    </Message>
  - <Results>
      <Fight oper="=" value="3" />
    </Results>
    <Heuristics />
  </ID23>
- <ID8 Context="DZIGA-VERTOV ML(480,770,25)" Date="" From="M-Views" Repeatable="false"
      Story="FifteenMinutes" Subject="You may know too much..." To="">
  - <Requires>
      <David_value expr="==" value="2" />
      <Fight expr="==" value="0" />
    </Requires>
  - <Message>
      <Text>You may know too much for your own good...</Text>
      <MediaURL>:http://keaton.media.mit.edu:8080/Media/09DavidWarning.mpg<
          /MediaURL>
    </Message>
  - <Results>
      <David_value oper="=" value="3" />
    </Results>
    <Heuristics />
  </ID8>
- <ID16 Context="DZIGA-VERTOV ML(480,770,25)" Date="" From="M-Views" Repeatable="false"
      Story="FifteenMinutes" Subject="The End?" To="">
  - <Requires>
      <Eve_value expr="#LT" value="4" />
      <Fight expr="==" value="2" />
      <_ expr="?RT" value="ID13,3,-1" />
    </Requires>
  - <Message>
      <Text>Something doesn't add up...</Text>
    </Message>
    <MediaURL>:http://keaton.media.mit.edu:8080/Media/18BadEnding.mpg</MediaURL>
  - <Results>
      <Fight oper="=" value="3" />
    </Results>
    <Heuristics />
  </ID16>
- <ID18 Context="DZIGA-VERTOV ML(480,770,25)" Date="" From="M-Views" Repeatable="false"
      Story="FifteenMinutes" Subject="Too late..." To="">
  - <Requires>
      <Fight expr="==" value="1" />
      <_ expr="?RT" value="ID12,4,-1" />
    </Requires>
  - <Message>
      <Text>Why didn't you try to stop Michael?</Text>
    </Message>
    <MediaURL>:http://keaton.media.mit.edu:8080/Media/16BadEnding.mpg</MediaURL>
  - <Results>
      <Fight oper="=" value="3" />
    </Results>
    <Heuristics />
  </ID18>
- <ID3 Context="DZIGA-VERTOV ML(480,770,25)" Date="" From="M-Views" Repeatable="false"
      Story="FifteenMinutes" Subject="A Memory..." To="">
  - <Requires>
      <David_value expr="==" value="1" />
      <Fight expr="==" value="0" />
      <_ expr="?RT" value="ID0,1,-1" />
    </Requires>
  - <Message>
      <Text>David recalls a friend.</Text>
    </Message>
    <MediaURL>:http://keaton.media.mit.edu:8080/Media/04DavidWatch.mpg</MediaURL
      >
  - <Results>
      <David_value oper="=" value="1" />
```

```
      </Results>
      <Heuristics />
    </ID3>
  - <ID7 Context="STEVEN-SPIELBER ML(525,700,40)" Date="" From="M-Views" Repeatable="false"
        Story="FifteenMinutes" Subject="Eve suspects" To="">
    - <Requires>
        <Eve_value expr="==" value="1" />
        <Fight expr="==" value="0" />
        <_ expr="?RT" value="ID2,1,-1" />
      </Requires>
    - <Message>
        <Text>Eve senses trouble.</Text>
      </Message>
      <MediaURL>:http://keaton.media.mit.edu:8080/Media/08EveCameras.mpg</MediaURL
          >
    - <Results>
        <David_value oper="=" value="2" />
        <Eve_value oper="=" value="2" />
      </Results>
      <Heuristics />
    </ID7>
  - <ID12 Context="DZIGA-VERTOV ML(480,770,25)" Date="" From="M-Views" Repeatable="false"
        Story="FifteenMinutes" Subject="Watch out!" To="">
    - <Requires>
        <_ expr="?RT" value="IDX,5,-1" />
        <Fight expr="==" value="0" />
        <Michael_value expr="!=" value="4" />
        <_ expr="!?LS" value="ID0" />
        <_ expr="!?LS" value="ID3" />
        <_ expr="!?LS" value="ID8" />
      </Requires>
    - <Message>
        <Text>David gets a visit from an old friend...</Text>
      </Message>
      <MediaURL>:http://keaton.media.mit.edu:8080/Media/13Fight.mpg</MediaURL>
    - <Results>
        <Fight oper="=" value="1" />
      </Results>
      <Heuristics />
    </ID12>
  - <ID17 Context="DZIGA-VERTOV ML(480,770,25)" Date="" From="M-Views" Repeatable="false"
        Story="FifteenMinutes" Subject="Congratulations..." To="">
    - <Requires>
        <Eve_value expr="==" value="4" />
        <Fight expr="==" value="2" />
        <_ expr="?RT" value="ID13,3,-1" />
      </Requires>
    - <Message>
        <Text>You've done well...</Text>
      </Message>
      <MediaURL>:http://keaton.media.mit.edu:8080/Media/19EveEnding.mpg</MediaURL>
    - <Results>
        <Fight oper="=" value="3" />
      </Results>
      <Heuristics />
    </ID17>
  - <ID2 Context="STEVEN-SPIELBER ML(525,700,40)" Date="" From="M-Views" Repeatable="false"
        Story="FifteenMinutes" Subject="Hello, Stranger" To="">
    - <Requires>
        <_ expr="?RT" value="IDX,1,-1" />
        <Fight expr="==" value="0" />
        <Eve_value expr="==" value="0" />
      </Requires>
    - <Message>
        <Text>Meet one of Infosafe's finest...</Text>
      </Message>
      <MediaURL>:http://keaton.media.mit.edu:8080/Media/03EveIntro.mpg</MediaURL>
    - <Results>
        <Eve_value oper="=" value="1" />
      </Results>
      <Heuristics />
    </ID2>
  - <ID13 Context="DZIGA-VERTOV ML(480,770,25)" Date="" From="M-Views" Repeatable="false"
        Story="FifteenMinutes" Subject="Look out!" To="">
    - <Requires>
        <Fight expr="==" value="0" />
        <Michael_value expr="==" value="4" />
      </Requires>
    - <Message>
```

```
        <Text>David is visited by an old friend...</Text>
      </Message>
      <MediaURL>:http://keaton.media.mit.edu:8080/Media/14Fight.mpg</MediaURL>
    - <Results>
        <Fight oper="=" value="2" />
      </Results>
      <Heuristics />
  </ID13>
- <ID9 Context="WONG ML(720,820,25)" Date="" From="M-Views" Repeatable="false"
      Story="FifteenMinutes" Subject="Michael indisposed" To="">
    - <Requires>
        <Fight expr="==" value="0" />
        <Eve_value expr="#GTE" value="3" />
      </Requires>
    - <Message>
        <Text>Michael's not too pleased...</Text>
      </Message>
      <MediaURL>:http://keaton.media.mit.edu:8080/Media/10MichaelFloor.mpg</MediaUR
          L>
    - <Results>
        <Michael_value oper="=" value="4" />
        <Eve_value oper="=" value="3" />
      </Results>
      <Heuristics />
  </ID9>
- <ID21 Context="STEVEN-SPIELBER ML(525,700,40)" Date="" From="M-Views"
      Repeatable="false" Story="FifteenMinutes" Subject="Eve knew it." To="">
    - <Requires>
        <Fight expr="==" value="0" />
        <David_value expr="==" value="3" />
        <Eve_value expr="#GT" value="0" />
        <_ expr="?LS" value="ID8" />
      </Requires>
    - <Message>
        <Text>Eve knew something was going on...</Text>
        <MediaURL>:http://keaton.media.mit.edu:8080/Media/12EveToldYa.mpg</Med
            iaURL>
      </Message>
    - <Results>
        <Michael_value oper="=" value="4" />
        <Eve_value oper="=" value="5" />
      </Results>
      <Heuristics />
  </ID21>
- <ID1 Context="WONG ML(720,820,25)" Date="" From="M-Views" Repeatable="false"
      Story="FifteenMinutes" Subject="Meet the Boss" To="">
    - <Requires>
        <Fight expr="==" value="0" />
        <Michael_value expr="==" value="0" />
      </Requires>
    - <Message>
        <Text>Meet the CEO of Infosafe.</Text>
      </Message>
      <MediaURL>:http://keaton.media.mit.edu:8080/Media/02MeetMichael.mpg</MediaURL
          >
    - <Results>
        <Michael_value oper="=" value="1" />
      </Results>
      <Heuristics />
  </ID1>
- <ID20 Context="WONG ML(720,820,25)" Date="" From="M-Views" Repeatable="false"
      Story="FifteenMinutes" Subject="Michael awaits" To="">
    - <Requires>
        <Fight expr="==" value="0" />
        <David_value expr="#GTE" value="1" />
      </Requires>
    - <Message>
        <Text>What good is power if you can't use it?</Text>
      </Message>
      <MediaURL>:http://keaton.media.mit.edu:8080/Media/05MichaelPower.mpg</MediaU
          RL>
    - <Results>
        <Michael_value oper="=" value="3" />
      </Results>
      <Heuristics />
  </ID20>
- <ID14 Context="WONG ML(720,820,25)" Date="" From="M-Views" Repeatable="false"
      Story="FifteenMinutes" Subject="Just in time..." To="">
    - <Requires>
```

```
        <Michael_value expr="#LTE" value="3" />
        <Eve_value expr="!=" value="5" />
        <Fight expr="==" value="1" />
        <_ expr="?RT" value="ID12,0,4" />
      </Requires>
    - <Message>
        <Text>You're just in time...</Text>
      </Message>
      <MediaURL>:http://keaton.media.mit.edu:8080/Media/17GoodEnding.mpg</MediaURL
         >
    - <Results>
        <Fight oper="=" value="3" />
      </Results>
      <Heuristics />
    </ID14>
  - <ID5 Context="WONG ML(720,820,25)" Date="" From="M-Views" Repeatable="false"
       Story="FifteenMinutes" Subject="Michael suspects" To="">
    - <Requires>
        <Fight expr="==" value="0" />
        <Michael_value expr="==" value="1" />
        <David_value expr="==" value="0" />
        <_ expr="?RT" value="ID1,1,-1" />
      </Requires>
    - <Message>
        <Text>Michael suspects something may happen tonight...</Text>
      </Message>
      <MediaURL>:http://keaton.media.mit.edu:8080/Media/06MichaelSuspicious.mpg</Me
         diaURL>
    - <Results>
        <Michael_value oper="=" value="2" />
        <David_value oper="=" value="2" />
      </Results>
      <Heuristics />
    </ID5>
  - <ID0 Context="DZIGA-VERTOV ML(480,770,25)" Date="" From="M-Views" Repeatable="false"
       Story="FifteenMinutes" Subject="Elevator's broken..." To="">
    - <Requires>
        <Fight expr="==" value="0" />
        <David_value expr="==" value="0" />
        <_ expr="?RT" value="IDX,1,-1" />
      </Requires>
    - <Message>
        <Text>The elevator repair man is here.</Text>
      </Message>
      <MediaURL>:http://keaton.media.mit.edu:8080/Media/01DavidEnters.mpg</MediaURL
         >
    - <Results>
        <Michael_value oper="=" value="1" />
        <David_value oper="=" value="1" />
      </Results>
      <Heuristics />
    </ID0>
  </FifteenMinutes>
```


Table 24 XML script of 15 Minutes

APPENDIX THREE: CREDITS

M-Views System 0.1 and *Another Alice*:
Research Supervisor: *Glorianna Davenport* ***Investigator:*** *Pengkai Pan* ***Research Team:*** *David Crow, Isaac Rosmarin, Steven Chan* ***Production Crew:*** *Christina Chen, David Crow, Daniel Mcanulty, Isaac Rosmarin*

M-Views System 0.2 and *MIT in Pocket*:
Research Supervisor: *Glorianna Davenport* ***Principal Investigator:*** *Pengkai Pan* ***System Development Team****: David Crow and Carly Kastner* ***Production Team****: David Crow, Lilly Kam, Debora Lui, and Chris Toepel.* ***Leading Characters****: David Crow, Lilly Kam, Debora Lui, and Chris Toepel.* ***Supporting Characters****: Dan Bersak, Donna Tversky, Alan Brody, Bill Mitchell, Dan Katz, Tim Sutherland, Welkin Pope, Jumaane Jeffries, Bao-Yi Chang, Chris Avrich, Sayre Neufield C, Rachel Kline, Elizabeth Jochum S, Imani Ivery S, et al.*

15 Minutes:
Research Supervisor: *Glorianna Davenport* ***Director and Writer:*** *David Crow* ***Production Crew:*** *David Crow, Lilly Kam, Pengkai Pan, Welkin Pope G, Surj Patel, et al.*

BIBLIOGRAPHY

Abowd *et al.* 1997	Abowd, G., Atkeson, C., Hong, J., Long, S., Kooper, R. and Pinkerton, M. (1997). “Cyberguide: A mobile context-aware tour guide.” Wireless Networks, 3(5):421-433.
Agamanolis 2001	Agamanolis, S. (2001). Isis, Cabbage, and Viper: New Tools and Strategies for Designing Responsive Media, Ph.D. Thesis, MIT.
Aoki *et al.* 2002	Aoki, P. M., Grinter, R. E., Hurst, A., Szymanski, M. H., Thornton, J. D. and A. Woodruff. (2002). “Sotto Voce: Exploring the Interplay of Conversation and Mobile Audio Spaces.” Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems, Minneapolis, MN, Apr. 2002, 431-438.
Baecker *et al.* 1995	Baecker, R., Grudin, J., Buxto, W., and Greenberg, S. (1995). A historical and intellectual perspective, Human-computer interaction: toward the year 2000, Morgan Kaufmann Publishers Inc., San Francisco.
Bahl and Padmanabhan 2000	Bahl, P., and Padmanabhan, V. N. (2000). “RADAR: An inbuilding RF-based user location and tracking system.” In Proceedings of the IEEE Infocom, Tel Aviv, Israel.
Baird 1998	Baird, F. (1998). The Un/Real Duet: Intimacy & Agency through Interaction with a Virtual Character, Master’s Thesis, MIT.
Balabanovi'c and Shoham 1997	Balabanovi'c, B., and Shoham, Y. (1997). “Fab: Content-based, collaborative recommendation.” Communications of the ACM, 40(3): 66-72, March 1997.
Benford *et al.* 2003	Benford, S., Anastasi, R., Flintham, M., Drozd, A., Crabtree, A., Greenhalgh, C., Tandavanitj, N., Adams, M., and Row-Farr, J. (2003). “Coping with Uncertainty in a Location-Based Game.” Pervasive Computing. July-September.
Borovoy *et al.* 2001	Borovoy, R., Silverman, B., Gorton, T., Klann, J, Notowidigdo, M., Knep, B., and Resnick, M. (2001). “Folk Computing: Revisiting Oral Tradition as a Scaffold for Co-Present Communities.” Proceedings of the CHI 2001 Conference on Human Factors in Computing Systems. New York: ACM.
Branigan 1992	Branigan, E. (1992). Narrative Comprehension and Film. New York: Routledge.
Brondmo and Davenport 1989	Brondmo, H. and Davenport, G. (1989). “Creating and viewing the elastic Charles—A Hypermedia Journal.” Hypertext II Conference Proceedings, Chapter 5, pp. 43-51.
Brooks 1996	Brooks, K. (1996). “Do Story Agents Use Rocking Chairs? The Theory and Implementation of One Model for Computational Narrative.” ACM Multimedia ’96. Boston, MA.
Brooks 2003	Brooks, K. (2003). "There is nothing virtual about immersion: Narrative immersion for VR and other interface." (Unpublished manuscript).
Bruner 1990	Bruner, J. (1990). Acts of Meaning. Harvard University Press.
Byun and Cheverst 2003	Byun, H. E., and Cheverst, K. (2003). “Supporting Proactive ‘Intelligent’ Behaviour: the Problem of Uncertainty.” Workshop on User Modeling for Ubiquitous Computing in conjunction with User Modeling 2003, Pittsburgh, PA, USA.

Capkun *et al.* 2002	Capkun, S., Hamdi, M., and Hubaux, J.P. (2002). "GPS-free Positioning in Mobile Ad Hoc Networks." Cluster Computing.
Chatman 1978	Chatman, S. (1978). Story and Discourse - Narrative Structure in Fiction and Film. 1st ed. Cornell University Press, Ithaca and London.
Cheverst *et al.* 2000	Cheverst, K., Davies, N., Mitchell, K., Friday, A. and Efstratiou, C. (2000) "Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences." Proc. of CHI'2000, Netherlands.
Costikyan 2001	Costikyan, G. (2001). "Where stories end and games begin" http://www.costik.com/gamnstry.html
Crow *et al.* 2003	Crow, D., Pan, P., Kam, L., and Davenport, G. (2003). "M-Views: A System for Location-Based Storytelling." ACM UbiComp 2003, Seattle, WA.
Davenport 2002	Davenport, G. (2002). "Art and other anxieties: Steerable stories." A keynote at the conference of Story telling in the Digital Age, National Institute of Design, Ahmedabad, India.
Davenport and Friedlander 1995	Davenport, G. and Friedlander, L. (1995). "Interactive transformational environments: wheel of life." (Chapter 1, pp. 1-25). Contextual Media: Multimedia and Interpretation. MIT Press. Cambridge, MA.
Davenport and Murtaugh 1995	Davenport, G. and Murtaugh, M. (1995). "ConText: Towards the evolving documentary." ACM Multimedia '95.
Davenport and Murtaugh 1997	Davenport, G. and Murtaugh, M. (1997). "Automatist storyteller systems and the shifting sands of story." IBM Systems Journal, vol.36, no.3, pp.446-56. Publisher: IBM, USA.
Davenport *et al.* 2000	Davenport, G., Bradley, B., Agamanolis, S., Barry, B., and Brooks, K. (2000). "Synergistic storyscapes and constructionist cinematic sharing." IBM Systems Journal, vol.39, no.3-4, pp.456-69.
Delany and Landow 1991	Delany, P. (Editor), and Landow, G. (Contributor) (1991). Hypermedia and Literary Studies. MIT Press.
Dix 1995	Dix, A. J. (1995). "Cooperation without (reliable) Communication: Interfaces for Mobile Applications." Distributed Systems Engineering, 2(3): 171--181.
Domike *et al.* 2003	Domike, S., Mateas, M., and Vanouse, P. (2003). "The recombinant history apparatus presents: Terminal time." In M. Mateas & P. Sengers (Eds.), Narrative Intelligence. Amsterdam: John Bengamins Publishing.
Horvitz 1999	Horvitz, E. (1999). "Principles of Mixed-Initiative User Interfaces." ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '99), pages 159--166, Pittsburgh, PA.
Horvitz 2002	Horvitz, E. (2002). "Uncertainty, intelligence, and interaction." Proceedings of the 7th international conference on Intelligent user interfaces. San Francisco, California, USA.
Horvitz *et al.* 1998	Horvitz, E., Breese, J., Heckerman, D., Hovel D., and Rommelse, K. (1998) "The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users." Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pp. 256-265, Madison, Wisconsin, 1998.

Houbart 1994	Houbart, G. (1994). Viewpoints on Demand: Tailoring the Presentation of Opinions in Video. Master's Thesis, MIT.
James and Sayrs, 1999	James, J., and Sayrs, B. (1999). "Uncertainty Management: Keeping Battlespace Visualization Honest" ATIRP Conference.
Jenkins 2002	Jenkins, H. (2002). "Interactive audiences?" The 'collective intelligence' of media fans" in Dan Harries (Ed.), The New Media Book. London: British Film Institute.
Joachims *et al.* 1997	Joachims, T. Freitag, D. and Mitchell, T. 1997. Webwatcher: A tour guide for the World Wide Web. In Proc. IJCAI-97.
Kastner 2002	Kastner, C. (2002). M-Studio: An Authoring Tool for Context-Aware Mobile Storytelling. Master's Thesis, MIT, Cambridge, MA.
Konstan *et al.* 1997	Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl, J. (1997). "GroupLens: Applying Collaborative Filtering to Usenet News." Communications of the ACM 40,3 77-87.
Krizanc 1997	Krizanc, J. (1997). Tamara: A Play. Stoddart Publisher.
Kroeber 1992	Kroeber, K. (1992). Retelling/Rereading: The Fate of Storytelling in Modern Times. Rutgers University Press.
Lieberman 1997	Lieberman, H. (1997). "Autonomous Interface Agents." Proceedings of the ACM Conference on Computers and Human Interface, CHI-97.
Lynch 1960	Lynch, K. (1960). The Image of the City. MIT Press.
Maes 1995	Maes, P. (1995). "Intelligent Software." Scientific American, Vol. 273, No. 3, pp. 84-86, September 1995.
Maglio *et al.* 2000	Maglio, P., Matlock P. T., Campbell, C. S., Zhai, S., and Smith, B. A.. (2000) "Gaze and Speech in Attentive User Interfaces." Proceedings of the International Conference on Multimodal Interfaces.
Marmasse and Schmandt 2000	Marmasse, N. and Schmandt, C. (2000). "Location--aware information delivery with ComMotion." In HUC, pages 157—171.
Martin 1996	Martin, R. (1996). "Between teller and listener: The reciprocity of story telling." In Birch, C. L. and Heckler, M. A. (Eds.), Who Says? - Essays on Pivotal Issues in Contemporary Storytelling (pp. 141-154). Little Rock, Arkansas: August House.
Mazalek 2001	Mazalek, A. (2001). Tangible Interfaces for Interactive Point-of-View Narratives, Master's Thesis, MIT.
Minsky 1988	Minsky, M. (1988). Society of Mind. Simon & Schuster.
Morgenroth 1995.	Morgenroth, L. (1995). Movies, Talkies, Thinkies: An Experimental Form of Interactive Cinema, Master's Thesis, MIT.
Murtaugh 1996	Murtaugh, M. (1996). The Automatist Story telling System: Putting the Editor's Knowledge in Software, MIT Master's Thesis, MIT.

Myers 1988	Myers, B.A. (1988). "A taxonomy of user interfaces for window managers." IEEE Computer Graphics and Applications, pp. 65-84.
Nack and Parkes, 1997	Nack, F., and Parkes, A. (1997). "The application of video semantics and theme representation in automated video editing." Multimedia tools and Applications, vol. 4, no. 1.
Norman 1999	Norman, D. (1999). The Invisible Computer. MIT Press.
Norman and Draper 1986	Norman, D. and Draper, S. (1986). User Centered System Design. L. Erlbaum & Associates, Hillsdale, NJ.
Pan *et al.* 2002	Pan, P., Kastner, C., Crow, D., and Davenport, G. (2002). "M-Studio: An authoring application for context-aware multimedia." ACM Multimedia 2002, Juan-les-Pins, France.
Pearl 1988	Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, San Mateo, CA.
Petrelli *et al.* 1999	Petrelli, D., Elena, N., Marcello, S., Strapparava, C., Stock, O., Zancanaro, M. (1999). "HyperAudio: Location-Awareness + Adaptivity." In Proceedings of CHI'99, Conference on Human Factors in Computing Systems.
Pinhanez *et al.* 2000	Pinhanez, C. S., Davis, J. W., Intille, S., Johnson, M. P., Wilson, A. D., Bobick, A. F. and Blumberg, B (2000). "Physically interactive story environments." IBM Systems Journal Vol. 39, No. 3 & 4,
Reinhard *et al.* 1999	Reinhard, O., Specht, M., and Jaceniak, I. (1999). "Hippie: A Nomadic Information System." In Proceedings of 1st International Symposium on Handheld and Ubiquitous Computing (HUC '99).
Rheingold 2002	Rheingold, H. (2002). Smart Mobs: The Next Social Revolution, Perseus Publishing
Rhodes 2000	Rhodes, B. (2000). Just-In-Time Information Retrieval. Ph.D. Thesis, MIT Media Laboratory, Cambridge, MA, June 2000.
Robertson *et al.* 1977	Robertson, G., Newell, A., and Ramakrishna, K. (1977). "ZOG: A Man-Machine Communication Philosophy." Carnegie Mellon University Technical Report.
Rodden *et al.* 1998	Rodden, T., Chervest, K., and Davies, N. (1998). "Exploiting Context in HCI Design for Mobile Systems." First Workshop on Human Computer Interaction with Mobile Devices
Salber *et al.* 1999	Salber, D., Dey, A., Abowd, G. (1999). "The Context Toolkit: Aiding the development of context-enabled applications." In: Proceedings of CHI'99. 434—441.
Schank 1990	Schank, R. (1990). Tell me a story: A new look at real and artificial memory. New York, NY: Charles Scribner.
Schechner 1994	Schechner, R. (1994). Environmental Theater. New York, NY: Applause Books.
Shardananda and Maes 1995	Shardananda, U., and Maes, P. (1995). "Social information filtering: Algorithms for automating 'word of mouth.'" In Proc. of the International Conference on Computer-Human Interfaces, 1995.

Shneiderman 1983	Shneiderman, B. (1983). "Direct manipulation: A step beyond programming languages." IEEE Computer (1983), pp. 57-69.
Smith *et al.* 2000	Smith, B., Bender, W., Endter, I., Driscoll, J., Turpeinen, M., and Quan, D. (2000). Silver Stringers and Junior Journalists: Active Information Producers. IBM Systems Journal, Vol. 39.
Sparacino 2001	Sparacino, F. (2001). Storychastics: a Bayesian network architecture for combined user modeling, sensor fusion and computational story telling for interactive spaces. Ph.D. Thesis, MIT.
Sparacino 2003	Sparacino, F. (2003). "Sto(ry)chastics: A Bayesian Network Architecture for User Modeling and Compuatational Storytelling for Interactive Spaces." UbiComp 2003, Seattle, WA, USA.
Sparacino *et al.* 2000	Sparacino, F., Davenport, G., and Pentland, A. (2000). "Media in performance: Interactive spaces for dance, theater, circus, and museum exhibits." IBM Systems Journal Vol 39, Numbers 3 & 4.
Weiser 1991	Weiser, M. (1991). "The computer for the 21st century." Scientific American, 265(3):94—104.
Winkler 2000	Winkler, T. (2000). "Audience participation and response in movement-sensing installations." In Proceedings of the 2000 International Computer Music Conference, International Symposium on Electronic Art (ISEA), Paris, France.