System Architecture for Developing Mobile Cinema

David Crow
Media Laboratory Europe
Sugar House Lane
Bellevue, Dublin 8, Ireland
crow@mit.edu

Pengkai Pan
Room E15-351 Media Laboratory
Massachusetts Institute of Technology
20 Ames Street, MA, 02139, USA
ppk@media.mit.edu

Glorianna Davenport
Room E15-368c Media Laboratory
Massachusetts Institute of Technology
20 Ames Street, MA, 02139, USA
gid@media.mit.edu

ABSTRACT

Mobile Cinema is embodied in temporally and spatially discontinuous narrative segments that can be delivered on wireless PDAs as users navigate physical locations and interact with the environment. Mobile Cinema takes as its starting point the truism that "every story is a journey" and bends this idea into a new form in which the narrative is augmented by physical surroundings, social engagement, and contextual awareness.

To better handle participatory, context-aware, and evolving mobile story experiences, Mobile Cinema requires methodologies and flexible technology for allowing story makers to create coherent, immersive, and appealing content. The system architecture for developing Mobile Cinema presented in this paper offers several key features: an indoor location detection system for wireless networks, a universal messaging framework designed to support story scripting and logic, and a heuristic engine for user profiling. These features enable story makers to rapidly prototype, produce, and evaluate Mobile Cinema narratives. Our current Mobile Cinema project, *MIT in Pocket*, offers users a sense of campus life by telling a day-in-the-life story of four undergraduate students.

Keywords

mobile cinema, context-aware systems, application development platform, participatory media, wireless indoor location detection

1 INTRODUCTION

As personal mobile devices proliferate and wireless networks become ubiquitous, mobile multimedia becomes an increasingly curious anomaly because the genre does not distinguish between the creativity, intent or human impact of different experiences. Today we are on the verge of being able to capture, receive and transmit the full range of media elements (text, image, graphics) and sensed signals (haptic, bio, environmental) anytime, anywhere, to anyone or any group over a mix of wireless and wired networks. This is a grand technological achievement. However, the fact that we can make and share media fragments via email, web posting, IR or other proximity transmission begs the question of how our notion of content and meaning is transformed by this emergent e-mobility.

We are interested in applications that create perceived coherency and meaning through the delivery of multiple media fragments in an environment that is, by nature, intermittent, discontinuous and multimodal. For the purposes of this paper, we reuse Brooks' definition of narrative as a meaningful collection of media fragments about events/things/people that contain correlated temporal dependencies. With reference to Chatman and Branigan, Brooks points out: "To 'tell a story' means to choose a medium for the telling and to construct it by imparting some amount of editorial power over all the possible material which could be used." [4] [5] [6]

Figure 1: A mobile cinematic story, *MIT in Pocket*

Mobile Cinema is an interactive cinematic experience that provides its audience with relevant media sequences based on narrative constraints and close to real-time insight into audience context.

Conference '03, November 2-8, 2003, Berkeley, CA, USA

Mobile Cinema is designed for delivery to the subscriber's PDAs or cell phone as s/he moves from place to place according to day-to-day goals, agenda, and serendipitous encounters—many of which will have nothing to do with the underlying story experience.

Mobile Cinema takes as its starting point the truism that "every story is a journey" and bends this idea into a new form in which the narrative is augmented by physical surroundings, social engagement, and contextual awareness. As the participant navigates physical space, s/he triggers distinct media elements; these media elements can be considered as embedded in the locale, often explicitly depicting events at the location where they appear. Unlike previous mobile tour guide systems that emphasize the delivery of local information, Mobile Cinema immerses the participant in a coherent narrative progression. The individual media segments are acquired at discrete times and places, with allowances for the serendipitous augmentation of the whole experience through instant messaging, real-time scripting and logic, and deliberate interactions between viewers.

During this research project, we have developed two productions. In *Another Alice* [17], a comedic mystery, the story creator provides a single starting point for all viewers; once this segment has been viewed, each subscriber completes the plot according to his or her own decisions about which path to follow and which actions to take within the given amount of time. In our most recent production, *MIT in Pocket*, any subscriber can enjoy the story as they walk around the MIT campus. Structured around a day in the life of four undergraduate students, the production includes over forty story events. While our system delivers these segments based on scripted contextual cues including location and time of day, the viewers are also able to send messages and share their acquired video clips with others. Figure 1 shows an example of a viewer receiving a clip—a chance encounter between two musicians at the MIT Media Lab. This mobile story tapestry includes both fictional and non-fictional components. 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. As this paper progresses, we will refer back to the *MIT in Pocket* story as an evaluation of our system.

One of our hypotheses regarding story development for Mobile Cinema is that sequences should visually reflect more or less exactly the ambient location in which they are to be retrieved. [13]. Thus, a scene that takes place at the MIT Media Lab should be videoed at the Media Lab and will be delivered when the audience is in the vicinity of the Media Lab. Likewise, scenes that take place in the "infinite corridor" are delivered to the subscriber while they are traversing that corridor. This exact matching enables a perceptual merging of real and story space. This context is easily augmented by other causal/temporal linkages that are relevant to the story, such as environmental conditions (weather) or the state of story receiver (heart rate).

Context-aware mobile stories offer advantages of complexity, personalization and game-like engagement, however their construction is not as straightforward as that of traditional movies. What kinds of technology could be invented to enable the creation of coherent, immersive, and appealing mobile stories? In 2001, we initiated the M-Views Project, to investigate some critical issues of multimedia research on mobile, context-aware, and ubiquitous platforms. The basic objectives of M-Views include (1) architecting an M-Views application development platform (*M-Views ADP*), (2) creating an authoring tool, *M-Studio* [16], (3) developing Mobile Cinema applications for the entertainment and leisure space (4) understanding the implications and methods for the creation of Mobile Cinema and communicating these to groups of would-be content developers.

This paper focuses on the overall functionality of the M-Views ADP, the ways in which it supports the creative process and aesthetic intent of *MIT in Pocket*, and highlights how we can adapt this system for other proposed Mobile Cinema experiences. In particular, we describe our approach to story scripts, flags, media streaming, messaging, personalization, and various modes of interaction. We also discuss our MapAgent algorithm for location discovery and methods for allowing clients to cross 802.11 wireless network boundaries smoothly. Figure 2 illustrates the high-level modules of the M-Views ADP.

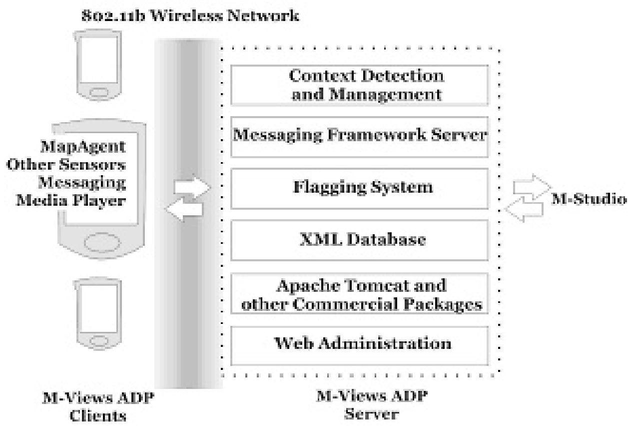

Figure 2: High-level modules of M-Views ADP

This paper proceeds as follows: In Section Two we discuss related mobile systems and similar development platforms. In Section Three, we discuss design principles that are employed in the architecture of the M-Views ADP. In Section Four, we illustrate the specific implementation of the M-Views ADP, including the client-server model, scripting capabilities, and the location detection algorithm. In Section Five, we discuss our story production experience as an evaluation of the M-Views ADP. We conclude in Section Six with a reflection on significant issues for Mobile Cinema developments and a discussion of future work.

2 RELATED RESEARCH

The basic concept of the M-Views ADP is similar to other projects that involve distributed application development. The Cyberdesk project [8] at Georgia Institute of Technology integrates diverse software behaviors across multiple computing platforms. The MIT Oxygen project [14] concentrates on eight ubiquitous computing technologies, such as the H21 handheld device, which relies on software to automatically detect and self-configure as a cell phone, pager, or network adapter. Other Oxygen projects are aimed at enhancing speech, collaboration, knowledge access, and everyday tasks. These projects are good examples of mobile devices that function as portals. Like Oxygen, the PIMA project [1] at IBM and the Portolano Project [15] at the University of Washington focus on new infrastructure developments for ubiquitous, context-aware, and mobile computing. Their research goals include application integration, system self-configuration, and middleware for diverse computing platforms. The M-Views ADP is designed to support the development of mobile multimedia applications, and further progress in mobile systems research is quite likely to affect the future evolution of the M-Views ADP.

Many experimental context-aware guide systems have been developed in the past decade and have influenced the design of the M-Views ADP. The Cyberguide system [9], the Guide system [7], the Hippie project [19], and the HyperAudio project [18] are all aimed at providing context-aware experiences for visitors, city tourists, or museum tourists. All these systems adopt client-server architectures similar to the M-Views ADP, but differ in that they do not focus on the narrative aspect. In addition, few systems are actual development platforms for mobile applications. Both HyperAudio and the Guide system support multimedia forms and adaptive hypermedia to provide context-aware information. However, these applications are stand-alone projects. In other words, none of this past research has focused on the development of cinematic narrative and little effort has been made to purposely support multiple mobile applications using these architectures. The authors of the Guide system have planned to "increase the diversity of context-sensitive interactive services." This proposal supports our belief that one context-aware computing platform should support multiple mobile applications.

As mobile information systems proliferate, our devices must become more intelligent in order to provide us with the information we need, when and where we need it—without constant user input. Flavia Sparacino has explored "Sto(ry)chastics," a predictive approach to delivering content for a museum guide system [20] [21]. In this system, observed visitor behavior is classified a priori. Her Museum Wearable system uses a sensor fusion approach to track the movement of actual visitors through an exhibition and classify them. Based on this classification, the system personalizes the delivery of content. The current version of the M-Views ADP has an adaptation/personalization module designed to profile users; however, it currently makes use of only very simple machine learning techniques. Sparacino's predictive approach research indicates an important direction for future work.

Mobile gaming has been on the rise for several years. Nokia has released the N-Gage mobile game deck [11], which touts a capability for both local and remote multiplayer games via Bluetooth and WAP over GPRS. *It's Alive*, a Swedish game company, has two location-based games: *Supafly* and *Botfighters* [10]. Both products rely on the location detection features built into newer mobile phones. *GoGame* [12] is a mobile scavenger hunt game based in San Francisco. Via phone, game participators are assigned "missions," which range from challenging a stranger, snapping pictures of passersby, or competing with other players. Game preparation is demanding, taking an average of two weeks and involving various tasks, such as writing questions, mapping routes, and planting clues. The I-Ball project [3] and the Meme Tags project [2] at the MIT Media Lab explore viral effects of face-to-face communication for community building. Although the goals of mobile games vary, many support multiple players and enhance many-to-many communication. These two features have also been adopted by the M-Views ADP.

3 M-VIEWS ADP DESIGN

The design approach to creating this application development platform has been bottom-up, case-based, and evolving. The focus has been on how to create methods and tools that encourage and support the entire creative process, from quick prototyping to debugging and release of mobile entertainment narratives. Many needs, requirements, and constraints have been discovered through our experience of authoring two mobile stories. In addition, we have incorporated our knowledge of established computer game design and considered emergent cultural behavior, particularly concerning the use of mobile phones, to make the ADP as flexible as possible.

By producing content in parallel with platform development, we tailored the ADP and authoring tools for three salient aspects of context-aware story authoring:

- The nature of participatory media: creators provide a story environment; the audience re-constructs and negotiates meaning by direct interaction—walking, sending and receiving messages, and dealing with the larger ambient environment. The mobile story maker is unable to know all possible ways that the audience might experience the story.

- The nature of context-aware narrative: in addition to inherent difficulties of sensing context, personalization of content using context-sensing techniques requires the author to build a multidimensional narrative. We found that prototyping and simulation tools can help authors pre-visualize and debug their content.

- The nature of evolving stories: given a client-server approach, the Mobile Cinema creator is able to iterate, change, and improve a mobile narrative over time. It is also conceivable to connect one mobile cinematic story with other mobile applications and/or cinematic stories.

The design of the M-Views ADP grew out of earlier prototypes for shareable and mobile media. Our experience suggested particular features that would be critical to a generalized application development package. These include:

- Support for connecting multiple mobile clients to a centralized server over a wireless network using an account/subscription model.

- Support for detecting and managing contextual data, as well as the ability to incorporate new types of sensors and computational processing.

- Facility to allow story developers to acquire and utilize context information about the subscriber, his or her personal profile, and the current environment.

We classify contextual data into three categories: physical context, social context, and story context. The physical context indicates an individual user's situation, such as the person's location, time/date, ambient light, weather, surrounding sound level, and some possible biological data (heart rate, for example). The social context describes relational data between users or the actions of the user in a social setting. For example, the M-Views ADP can be aware of how close two users are; it can compare one user's profile to others' data or be notified when a client interacts with a particular object. Social contextual data is very useful for game-like mechanisms. The story context is essential for narrative construction and experience. Unlike many mobile tour guides, in which location information is the primary trigger, Mobile Cinema upholds the continuity of discontinuous events (media clips). In the M-Views ADP, each event is associated with a set of flag requirements (logic statements with variables) to help determine, track, and maintain story context. For example, every event uses flag variables to know which clips have been previously seen and which clips might to be played later on. Furthermore, a mobile story might require that Clip A can only be viewed around 10:00 am, while Clip C cannot be viewed unless Clip B has already appeared.

The M-Views ADP provides the capability for rapid story prototyping and functional story authoring. Our experiences producing *Another Alice* and *MIT in Pocket* have taught us that Mobile Cinema is a challenging task for the creator for three reasons: (1) most creators don't understand how they can incorporate contextual information into their story or game applications, (2) both the multi-threaded narrative and the multi-user engagement add a level of complexity to scripting story arcs that are difficult to visualize without experience, (3) given undirected movements by the audience, it is nearly impossible for creators to predict all the different ways a story or game might unfold as the user moves through space and time. With these concerns in mind, enabling rapid prototyping at the early stages of application development becomes critical.

Currently, the M-Views ADP and M-Studio support the M-Views flagging system, story scripts, and several visual tools to accelerate the development of application prototypes. After creating a story script in M-Studio, all event data, including flag configurations and context requirements, is exported as an XML data file. The M-Views server then loads this file in order to deploy the story, game, or mobile application. Thus, M-Studio serves as the development environment for the M-Views ADP and a comfortable starting point for all authors.

Finally, the M-Views ADP gives story developers a convenient means to examine and evaluate the mobile story experience. As one creator of *MIT in Pocket* commented, it is very time-consuming to physically test a mobile story, especially one intended to unfold over the course of an entire day. Both the M-Views ADP and M-Studio provide some tools for simulation and examination. The M-Views ADP also allows the story creator to manipulate and test story scripts in order to examine the mobile experience prior to and during story deployment.

In summary, mobile cinematic experiences, which are mediated by context-rich sensory data as well as by social interaction, need to be flexible and responsive in their design. Our system achieves this through a combination of modular context sensing, flagging logic, user profiling, authoring tools, and evaluation utilities.

4 M-VIEWS ADP IMPLEMENTATION

The M-Views client-server architecture consists of multiple clients (handheld computers) connecting to a centralized server over a wireless (802.11) network. It makes use of an account/subscription service model, allowing users to subscribe to multiple story services at the same time. The server contains modules and information used for specific behavior, such as particular types of context monitoring or flag operations. These include the location detection engine and custom classes, which analyze sensor data sent by the client. Story scripts are also maintained on the server and dictate the content and media to be returned. M-Views applications are entirely defined by story scripts, custom classes, and location maps.

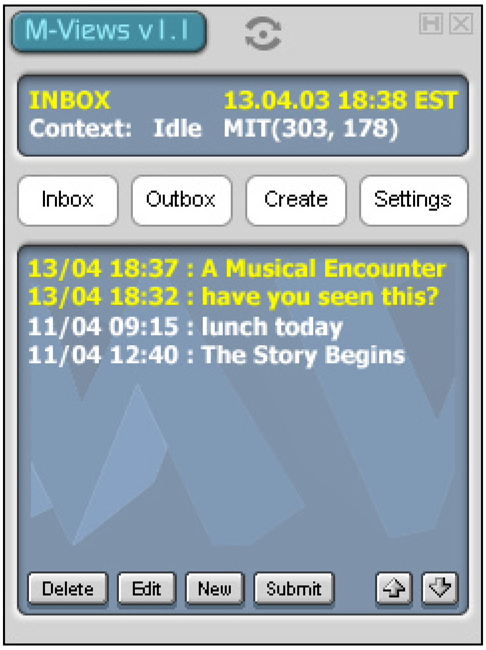

Figure 3: Client Interface on Pocket PC

The M-Views client operates on the Windows CE operating system (Pocket PC) and uses the CE Game API to provide a customized look and feel, dropping each new event into a message queue, which is visibly represented as the user's inbox. In addition to the message manager interface shown in Figure 3, the client also features a map viewer/editor tool. This permits users to see their server-calculated positions and those of others. It also allows administrators to calibrate map coordinates using the standard client. The software is modular and can be augmented for new functionality and sensors, while also using third party programs to stream media over the network. When a message arrives with an associated media URL, the streaming media player is launched. The user interface is generalized; parsed messages from the server are displayed and edited in the GUI, while sensor data is polled and sent to the server along with the pertinent user input. The complete information flow is diagrammed in Figure 4. Nearly all computation is subsequently performed by the server, thereby saving client resources.

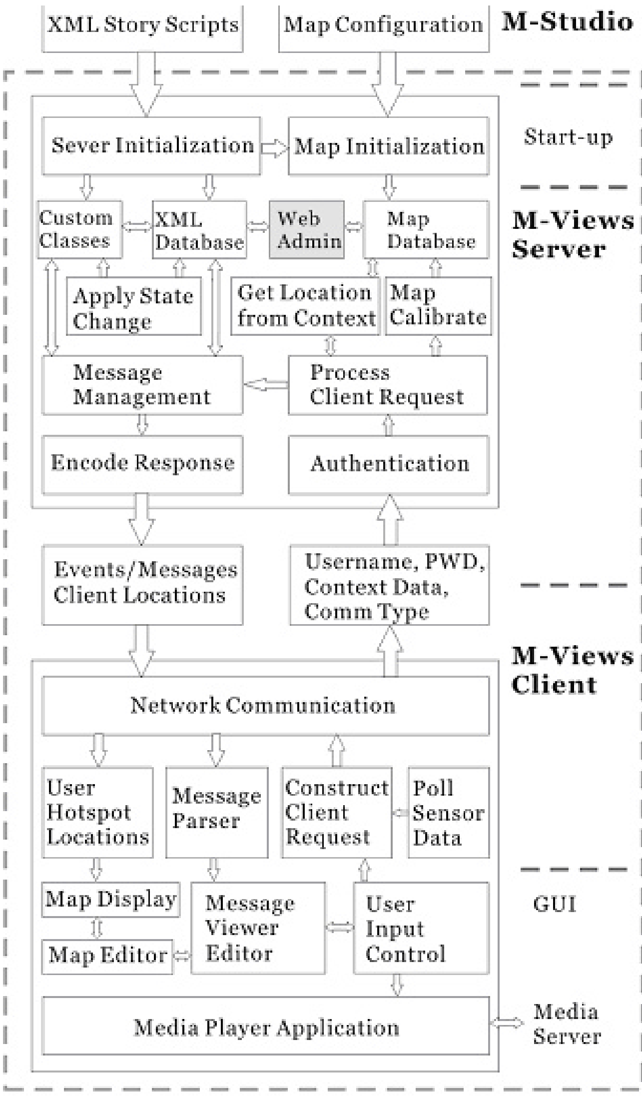

Figure 4: M-Views ADP Information Flow and Architecture

Communication between the client and server is carried out via HTTP POST requests. Every update cycle (approximately once per second), the client transmits authentication information, communication settings, and sensor data to the server, which then validates the information and sends back messages, story events, and location estimates. This communication scheme eliminates the need for a logon/logoff mechanism, and it is also fault-tolerant. If the connection is interrupted (perhaps due to losing wireless network coverage), the client will keep trying to send the last request until a connection is made or the program is terminated. To allow for roaming between wireless networks, the client attempts to reinitialize its wireless network card and DHCP address after any connection timeout or interruption. In practice, it takes an average of 30 seconds to reacquire a new network connection after the previous one has been lost.

The M-Views server is written in Java and runs as a servlet with the appropriate container software, such as Apache Tomcat. After initialization, the server maintains all story, message, and user information as memory-resident XML data. XML management is done using the Apache Xerces 2 package.

```
<ID5 Context="OFFICE" Date="" From="" Repeatable="true"
  Story="Tour" Subject="How to Install M-Views" To="*">
  <Requires>
    <Started expr="==" value="1" />
    <StepWFT expr="!=" value="1" />
    <StepWFT expr="!=" value="2" />
    <SeenINS expr="==" value="0" />
  </Requires>
  <Message>
    <Text>Want to install M-Views yourself? View this
      clip to find out how easy it is!</Text>
    <MediaURL>http://mvserver/INS.mpg</MediaURL>
  </Message>
  <Results>
    <SeenINS oper="+" value="1" />
  </Results>
  <Heuristics>
    <TechInterest value="8" />
  </Heuristics>
</ID5>
```

Figure 5: Story Event XML

The M-Views ADP features a generic messaging framework, which is specifically designed to support narrative structures but flexible enough to be used for a full range of applications. 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 mechanism. All messages and events are stored in either a story script or the general message forum (to which all users are subscribed and where client-to-client messages are created). Additionally, all messages, even those sent by clients, can be made context-dependant 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.

Story scripts contain a collection of messages (events). These XML elements include event information, requirements for the client's context and flag variables, state change information (applied to a user's profile when he or she receives the message), heuristics that describe the content, and an associated Media URL. Figure 5 shows event XML taken from an M-Views interactive tour application. The flagging system is used to specify story behavior based on user activity, and each event element contains flag requirements and results. If current variable values (maintained in the account data of each user) meet flag requirements, the event is considered encountered, and the variables are changed according to any update rules that may also be defined for that event. In addition to the flagging system, story scripts can use the heuristic profiling engine, which may be employed to enhance user experience. After receiving an event, any associated heuristic information is averaged into the current values stored in the user's account data. This means, for example, that encountering a "romantic" event could make the user more prone to receiving future story events that involve romance. Therefore, the more a user pursues favorite story themes, the more he or she will be rewarded with desired content.

MapAgent is the default location detection engine written for the M-Views ADP. M-Views clients monitor the Received Signal Strength Indicator (RSSI) for all 802.11 wireless access points in range. These measurements are averaged over a small time window and transmitted to instances of MapAgent running on the server. For each subscribed map, the associated MapAgent compares the RSSI averages to measurements recorded previously by an administrator at known locations, which are called hotspots. Hotspots have a threshold, and they are represented on the map with translucent circles, as in Figure 6. The MapAgent algorithm uses a combination of nearest neighbor matching, triangulation, and trajectory estimation to determine client locations. MapAgent also tracks client movement, allowing applications to incorporate a location-based social component.

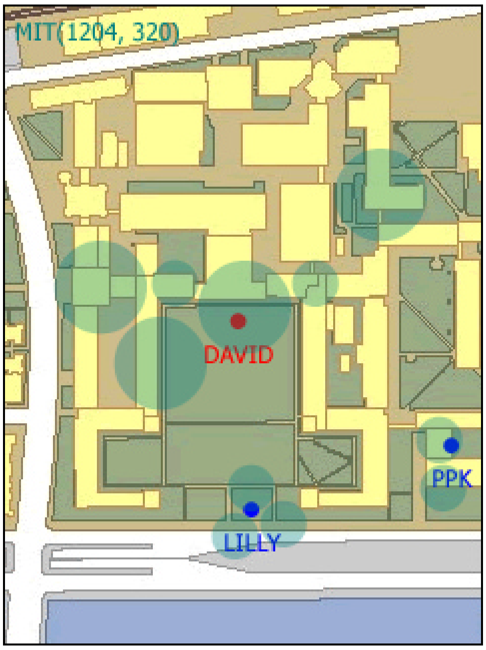

Figure 6: Map Monitor on the Client

Web-based administration provides access to all server-maintained XML documents. It provides statistics on message delivery and displays current log files. A map monitor and editing tool is also available over the web, which allows administrators to track the locations of users in real-time. Administrators may also edit map variables and calibration data via this interface.

5 EVALUATION

The merits of any development platform are judged by the applications built with that technology. To evaluate the M-Views ADP, we created *MIT in Pocket*, an ongoing Mobile Cinema production. The story was designed to be very non-linear, with no single conflict or climax. Instead, we leave audiences to find their own path and meaning in the dramatic tapestry. Viewers are given a day to explore the lives of four main characters. These characters are students at MIT who face humorous challenges, bizarre surprises, and difficult personal decisions, during what can only be described as a "typical day at the Institute." Indeed, many scenes are based on actual events. We made this production realistic for the same reason it is open-ended: our target audience—the larger MIT community including visitors, prospective students and area residents—would be better able to identify with the characters. The challenge for this production has been to develop fictional content that will convey the spirit of student life, while still remaining coherent regardless of who the viewer is and where (in time and space) he or she enters the story. This means that there will be no single physical starting point for the narrative, and that there will be no prescribed paths through the campus, since each visitor is likely to have a different agenda as well.

The development of *MIT in Pocket* has included the following steps: (1) identifying the audience (end users); (2) distinguishing relevant context types; (3) conceiving main story plots, characters, and events; (4) producing and post-producing stories; (5) scripting and simulating mobile experiences with M-Studio; (6) installing story scripts on the M-Views server; and (7) testing *MIT in Pocket*. By evaluating the production, we can better understand the balance required between visitors' navigational goals, their journeys through physical space and their emotional journey through the narrative space. Our evaluation has involved testing and discussions with the production crew. Through working closely with the story creators of *MIT in Pocket*, we have observed how the M-Views ADP supports Mobile Cinema developments in terms of *flexibility*, *evolvability*, and *continuity*.

Flexibility

The story creators initially identified both the MIT community and visitors as the end users. However, they did not realize that most MIT visitors spend less than half a day on campus. By contrast, *MIT in Pocket* was originally intended for people who would dedicate up to eight hours for the experience. Consequently, at the start of the evaluation phase, the creators were able to make easy adjustments to the story flag requirements in order to create a compressed, three-hour version of *MIT in Pocket*. There was no need to restructure or rewrite the story.

Evolvability

To date, the production contains forty video events. Now that we have moved into the final testing phase, our first results indicate that the story requires more event density. To improve this, the production team is inserting media fragments that provide a textured but less narrative sense of MIT (such as historical, documentary, and text-only clips). It is hoped that this textured approach will provide a greater level of immersion currently lacking in the first production release. Furthermore, to augment the narrative aspects of *MIT in Pocket*, the production team has recruited additional students and professors to create new plot events. Because Mobile Cinema is discontinuous, these productions can be carried out in a discontinuous and unrestricted manner. The medium and technology allow people to add new content to existing stories indefinitely.

Continuity

The visual storyboard and the flagging utilities help the story creators establish, simulate and examine plot structures. Nevertheless, not all aspects of the physical experience can be simulated using M-Studio. 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.

This evaluation process is still ongoing. There will be more story productions in the future, and they will incorporate elements from multiplayer gaming and additional sensor technologies. Additionally, further evaluation of multiple user interaction will be performed to investigate the effect of role-playing and collaboration on the story experience.

6 CONCLUSION AND FUTURE WORK

Since the earliest days of storytelling, the audience has experienced, responded to, and communally negotiated an interpretation of narrative landscapes. Technology has allowed that interpretation to be negotiated with others in new and powerful ways, thus expanding the capacity for novel participation and social dialogue. Incorporating mobility into the storytelling process is an innovation that creates both complexity and immediacy. While one could argue that it might be inappropriate to watch movie fragments while traveling, one can equally argue that navigating space is accomplished with many pauses, discoveries, and serendipitous meetings that could accommodate media augmentation. An immersive narrative can provide augmentation to the everyday situation, as well as time to negotiate meaning with a larger audience.

However, creating a new storytelling form using a new network channel is not enough: identifying how and why people embrace mobile services is also crucial to the design of Mobile Cinema applications because mobile services pose not only technical, but also behavioral, social, and economic challenges. The average Japanese citizen spends about seventy minutes every day traveling on trains between the office and home. Therefore, many mobile services in Japan are specifically designed for commuters. The MIT campus was chosen as the first test bed because MIT provides all kinds of human resources, as well as pervasive wireless networks. There are a number of active theater, storymaking, and gaming communities, as well as plenty of technology-savvy users who are willing to try out Mobile Cinema. To successfully introduce mobile, context-aware services, we believe that it is vital to understand and participate in the target community.

Prototyping is also not sufficient; deployment, self-support, and evaluation are also necessary to this research. To properly understand the potential of the M-Views ADP and M-Studio, a long-term study is required. Ideally, we would like to grow a community of self-motivated developers who would create their own Mobile Cinema applications on campus or elsewhere without significant support from our development team. Will Mobile Cinema be created, viewed, and discussed among students? Will mobile applications provide new methods for team building, training, and collaboration? How will other communities use the same technological infrastructure?

With these questions in mind, two additional research tasks have been introduced:

- Experimenting with social context: we would like to understand how the social proximity itself enhances scalability.

- Investigating how end users create their own content: in the evolution of *MIT in Pocket* we plan to explore ways in which users will be able to input command information as well as audio and video content.

In summary, the work of designing the M-Views ADP in parallel with the creation and evaluation of *MIT in Pocket* has provided us with an appreciation for the future potential of Mobile Cinema. Our current research involves inviting and evaluating how a new generation of authors imagines and realizes content for this new experimental cinema.

7 ACKNOWLEDGEMENTS

The authors wish to acknowledge the contribution of all M-Views project team members: Carly Kastner, Lilly Kam, Debora Lui, Chris Toepel, and Dan Bersak. Special thanks goes to our Interactive Cinema colleagues: Barbara Barry, Paul Nemirovsky, Aisling Kelliher, and Ali Mazalek. We also thank Prof. Alan Brody, Prof. Bill Mitchell, Prof. Donald Sadoway, and Prof. Ted Selker for excellent acting, advice, and support. We gratefully acknowledge Thomas Gardos from Intel, Taka Sueyoshi from Sony, Steve Whittaker from BT, and Franklin Reynolds from Nokia 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.

8 REFERENCES

[1]. Banavar, G., Beck, J., Gtuzberg, E., Munson, J., Sussman, J. and Zukowski, D., Challenges: An Application Model for Pervasive Computing. *7th International Conference on Mobile Computing and Networking*, 2000.

[2]. Borovoy, R., Martin, F., Vemuri, S., Resnick, M., Silverman, B., and Hancock, C., Meme Tags and Community Mirrors: Moving from Conferences to Collaboration. *Computer Supported Collaborative Work conference*, Seattle, pp. 159-168. 1998.

[3]. Borovoy, R., Silverman, B., Gorton, T., Klann, J, Notowidigdo, M., Knep, B., and Resnick, M., 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 2001.

[4]. Branigan, E., *Narrative Comprehension and Film*. New York, New York: Routledge, 1992.

[5]. Brooks, Kevin, Do Story Agents Use Rocking Chairs? The Theory and Implementation of One Model for Computational Narrative *Proceedings, ACM Multimedia 96,* November 1996, p.317-28.

[6]. Chatman, S., *Story and Discourse - Narrative Structure in Fiction and Film*. Ithaca and London Cornell University Press, 1978.

[7]. Cheverst, K., N. Davies, K. Mitchell, A. Friday, and C. Efstratiou., Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences. *Proc. of CHI'2000,* Netherlands, pp. 17-2, April 2000.

[8]. Dey, A.K., Abowd, G.D., Wood, A., CyberDesk: A Framework for Providing Self–Integrating Context–Aware Services *Knowledge Based Systems*, Vol. 11 (1), September 1998, pp. 3-13.

[9]. Gregory D. Abowd, Christopher G. Atkeson, Jason Hong, Sue Long, Rob Kooper, and Mike Pinkerton. Cyberguide: A mobile context-aware tour guide. *Wireless Networks*, 3(5):421-433, October 1997.

[10]. http://www.itsalive.com/

[11]. http://www.n-gage.com/n-gage/home.html

[12]. http://www.thegogame.com/

[13]. Jason Spingarn-Koff. Museum Tour: Walk This Way. http://www.wired.com/news/culture/0,1284,42152,00.html

[14]. M. Dertouzos. The Oxygen Project. *Scientific American*, 281(2):52-63, August 1999.

[15]. M. Esler, J. Hightower, T. Anderson, and G. Borriello., Next Century Challenges: Data-Centric Networking for Invisible Computing. The Portolano Project at the University of Washington. *In Proceedings of the Fifth ACM/IEEE International Conference on Mobile Networking and Computing*, pages 256-262, August 1999.

[16]. Pengkai Pan, Carly Kastner, David Crow, Glorianna Davenport, M-Studio: an Authoring Application for Context-aware Multimedia, *ACM Multimedia 2002,* Juan-les-Pins, France, 2002.

[17]. Pengkai Pan, Christina Chen, Glorianna Davenport, The Birth of "Another Alice". *COMPUTERS AND FUN 4*, King's Manor, University of York, UK, 2001.

[18]. Petrelli,Daniela, Not,Elena, Sarini,Marcello, O. Stock, a. trapparava and M. Zancanarov. HyperAudio: Location-Awareness + Adaptivity. *In Proceedings of CHI'99, Conference on Human Factors in Computing Systems*, 1999, pp 21-2.

[19]. Reinhard, M. Specht and I. Jaceniak. Hippie: A Nomadic Information System. *In Proceedings of 1st International Symposium on Handheld and Ubiquitous Computing (HUC '99)*, 1999, pp 330-3.

[20]. Sparacino, F. Storychastics: a Bayesian network architecture for combined user modeling, sensor fusion and computational storytelling for interactive spaces. *PhD thesis*, MIT, 2001.

[21]. Sparacino, F. Technologies and methods for interactive exhibit design: from wireless object and body tracking to wearable computers. In *Proceeding of the International Conference on Hypertext and Interactive Museums*, ICHIM 99, Washington DC, Sep 22-26.