Applying Active Space Principles to Active Classrooms

Chad Peiper, David Warden, Ellick Chan, Roy Campbell, Sam Kamin, †Tim Wentling
University of Illinois at Urbana-Champaign
Department of Computer Science
†Graduate School of Library and Information Science
<peiper, warden, emchan, rhc, kamin, wentling>@uiuc.edu

Abstract

Recent developments in pervasive computing have enabled new features for collaboration and instrumentation in educational technology systems. An infrastructure for the integration of components which support interactive pervasive education aids the development of these features. The e-Fuzion system contains the necessary components and management facilities to dynamically compose an adaptive learning system. e-Fuzion provides a set of distributed objects which allow for sharing of multimedia, recording of learning activity, and availability of real-time feedback information for instructors. Using the Gaia infrastructure, the system supports context-aware components and storage. e-Fuzion provides the extensibility necessary for the development of features to be integrated into educational technology systems.

1. Introduction

The rapidly evolving nature of education in the United States and elsewhere poses significant challenges to educators. Research has consistently shown that frequent communication between teacher and student can significantly enhance learning rates and overall student success [16]. Similarly, interactions between students can enhance peer to peer discovery, problem solving, and knowledge acquisition. However, increasing class sizes, particularly in technical fields, make it difficult to provide a maximal level of communication in the traditional lecture environment. Facilitating direct communication is even more difficult (though no less desirable) in “distance learning” settings.

Research in pervasive computing affords us an opportunity to investigate new paradigms for learning where dependencies from temporal and spatial factors are rendered obsolete. The integration of these technologies into an educational setting will result in a significant reduction in cost and a commensurate increase in the effectiveness of instruction [1].

e-Fuzion is an extensible architecture designed for the development of educational tools incorporating collaborative inking technology. Several pilot studies using an early version of e-Fuzion proved effective in enhancing learning outcomes [24]. Our centralized client-server system was not mature and extensible enough to support the integration of pervasive services. As a result, we created a proxy-based architecture to take advantage of the ubiquitous nature of smart devices [18]. This experience motivated us to envision the potential of a pervasive learning system.

Gaia is a distributed meta-operating system, designed to facilitate the creation and operation of active spaces. Among the services provided by Gaia are: component management, resource discovery, location discovery, and security. By utilizing a layered architecture, Gaia allows the abstraction of physical spaces and the various entities contained within it as a single programmable entity [21]. The integration of the Active Space architecture and e-Fuzion will foster the development of new tools and strategies designed for educational collaboration.

In the following section we describe the goals of our system. In the third section we describe our architecture for the integration of educational technology into a pervasive infrastructure. Subsequent sections discuss our work along with other research being done within the field of educational technology.

2. Goals of the System

The primary contribution of our work is the design of an architecture that facilitates the dynamic creation of adaptive learning environments through reusable components. Our efforts will leverage the Gaia infrastructure and context file system to integrate e-Fuzion’s tool-based system to facilitate the creation of a set of tools that allow students to interact via Tablet PC’s and other mobile devices.

A feature of our infrastructure made possible by pervasive computing is the ability to record and analyze student actions, while preserving a reasonable level of privacy. Gaia derives and correlates contextual information based on concrete metrics recorded by precision sensors. We apply the same data collection principles to the instructor; however, the instructor can set certain triggers to record pre-planned activities. This provides us with the flexibility to define policies that will allow us to analyze information while preserving privacy. Through the tuning of these policies, we can incrementally adapt our system to the needs of our learners.

3. Architecture Summary

3.1. Distributed Component Model

The goal of the e-Fuzion architecture is to provide the secondary infrastructure we consider essential to gain the benefits of pervasive technology in an educational computing system. In such a system, collaborative inking is a first class entity. All aspects of the system support the measurement of educational interaction dynamics with the purpose of providing feedback for both the student and instructor.

The e-Fuzion system architecture consists of a number of distributed objects integrated into the existing Gaia infrastructure [20]. These objects support e-Fuzion's set of tools for collaborative learning, recording and analysis of activity, and automation of course tasks. Inking is incorporated into all of the collaborative features such as note sharing and quizzing. All services support recording and playback of data, as well as provide introspective feedback. The pervasive computing infrastructure provides support for the use of educational context information as well as the ability for the distributed objects to communicate securely across heterogeneous devices.

The e-Fuzion architecture relies on a strict separation of the data model underlying each application module from its controller and/or display as well as from other modules [15]. Since components are specialized for their particular tasks and not coupled with unrelated features, this allows component re-use. Along with good object design, this provides for future extensibility. New features can be integrated with the system by composing components and services. Naturally, several models may be able to share a common display or controller component. The separation of the model and presentation components allows generic adaptation to Tablet PC and web devices.

The Gaia component model also enables e-Fuzion to become a first-class citizen in a pervasive environment. Each model may be distributed, moved, and adapted among a number of available execution nodes. The interconnection of modules in a hierarchy allows the system to scale to a large number of users. Different controllers and display components can be constructed and composed using well-defined functional interfaces to support different platforms. These features are particularly attractive to highly mobile users.

3.2. e-Fuzion Infrastructure

e-Fuzion employs the Gaia component structure and service architecture to facilitate the mobility, adaptation, and context-awareness which are then extended through e-Fuzion's modules. Gaia's space repository provides each participant with a list of available components in the system. Users are identified through the presence service, and Gaia's authentication and security services determine the set of allowable rights to components and services residing on client nodes. Gaia provides a unified context file system to all participants in the Active Space.

e-Fuzion integrates Gaia by creating one Active Space per "classroom." Students may log in to a class's space in several ways; through the security manager, RFID badges, fingerprints, and passwords. Fuzzy security policies associate a level of confidence given each authentication method [2]. For instance, taking an exam would require at least two of the three above means of identity verification.

Students typically access e-Fuzion on Tablet PCs. Once a student is logged onto a Tablet, three components are automatically generated. The first is an ink display module, which is responsible for rendering all graphical entities, including ink and figures on the screen. Students may annotate or modify the contents of the ink display though the tablet input controller, which reads ink strokes from the tablet's digitizer. Each tablet has a local ink model, which represents the set of ink strokes and annotations made to course material. Finally, a compositor component, described more fully in section 4.1, renders the ink model to a given device.

Gaia provides two primary support services for the classroom: A distributed file system, and contextual hints for activity-based modules. Contextual support services provide applications with the ability to communicate data which simplifies many tasks. For example, location data can be used to help locate nearby students exchange notes or a person to answer questions. e-Fuzion makes use of these services where ever possible to allow applications to exchange this information.

Students use the contextual file system by indexing and arranging files by common interests rather than by folders. The hierarchy changes in accordance with the current activity, and searching is made easier by the use of a simple predicate query language [11]. When students collaborate, the file system makes it easy to merge their shared document folders based on rules that create context-oriented folders for information.

3.3. Network Architecture

The network architecture of e-Fuzion follows the distributed model-view-controller pattern used in Gaia [20]. Peer to peer communication among students is supported, allowing group study sessions. To view each participant's slides, individual compositor modules on student's machines connect directly to remote ink models as shown in figure 1. The availability of note-sharing sessions is made available via the space repository service.

In a classroom setting, the instructor's notes are made available to students from authenticated sources found in the space repository. Students may also collaborate with their peers; adjacent pairs of students can be identified by the presence service. If most students access only instructor controlled sessions and that of a few peers within the classroom, network overload can be avoided. By providing shared note sessions on multiple servers, scalability can be achieved.

Persistently available course servers provide access to authoritative course materials such as quizzes. All traffic from the course server is encrypted via CORBA calls.

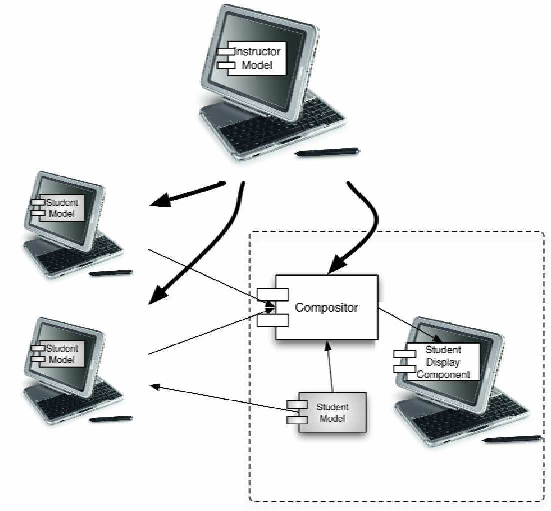


Figure 1: e-Fuzion architecture

In order to keep a classroom synchronized, the instructor node sends a message when the current page is updated. Different policies, defined on client machines determine what action to take. Similarly, for other events, students may specify policies that can aggregate or filter the event stream.

4. System Modules

4.1. Ink Layer Module

The ink layer module is responsible for the representation of visual data suitable for display, exporting, or printing. This support is used throughout e-Fuzion's feature set from note sharing, to examination scoring. The ink layer supports Tablet PC rich inking; it also supports a variety of graphical objects such as tables and images [13].

The ink layer provides the ability to write, draw, annotate, and highlight information. It allows many vector-based visual editing operations, such as the copying, movement and scaling of objects while maintaining the proper aspect ratio. This support is particularly sophisticated in the case of inking. Potential developments of inking in education show great promise [4]. By analyzing the inking to recognize words, gestures and capture meaning, annotations can be reformatted for display on different devices while still retaining their meta-information [5]. The inking is represented as an XML document to facilitate portability among different technologies [12]. Pages can be generated from importation of presentation slides and exported to common file formats through a compositor, including postscript and PDF.

The ink layer follows the typical model-view-controller design required by the Gaia Application Framework [21] and provides a generic representation of ink strokes and other objects on an ordered collection of pages. An ink layer controller typically has an integrated display in order to visualize operations in real-time as they occur on the e-Fuzion client. Clients that do not support rich inking may implement a reduced set of functionality, such as the standard drawing operations performed with a mouse. Similarly, resource constrained devices may display page layers flattened from a rich representation to raster images. The Gaia Context File System performs format conversion of rich ink into a suitable representation for mobile devices [11].

Ink layers may be shared amongst different controllers and displays on multiple client machines in order to facilitate note sharing and related activities. Policies are defined for resolving multiple accesses to the ink model, which coordinates a synchronous view to all subscribers. This is conceptually similar to how access is defined under a database view, where the view definition determines the scope of information access. Gaia uses Prolog-based policies to resolve access permissions for a space [20], these same policies are applied to individual displays on Tablet PCs. Using these policies, instructors can define an access control list, disable inking, or any combination of context-sensitive actions. For example, the shared instructor board may only allow annotations when the instructor, who is wearing a location tag, is in close proximity to the classroom.

Users wishing to interact with the shared session may elect to use the current page, insert a new shared page, or insert a collection of such pages. Shared layers contain a representation of the model on each execution node which serves as a bridge to the shared model allowing the data to be available in case connectivity is lost.

Multiple ink layers may need to simultaneously present objects such as when a student adds personal notes to shared slides provided by an instructor or when a grader annotates his or her comments on an electronically submitted homework assignment. To display information from multiple sources, we define a compositor architecture which allows a user to “skin” or define their own interface to optimally visualize the data. This allows us to define policies such as transparent layering, side-by-side views, and document re-flowing as shown in figure 2. The compositor allows us to adapt our visualization to next generation human interface devices. For example, the system can provide an instructor with a visualization of inking activity by composing the aggregate pen usage at any given moment.

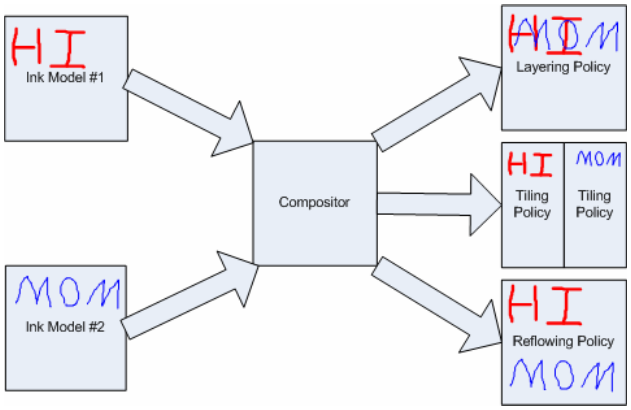


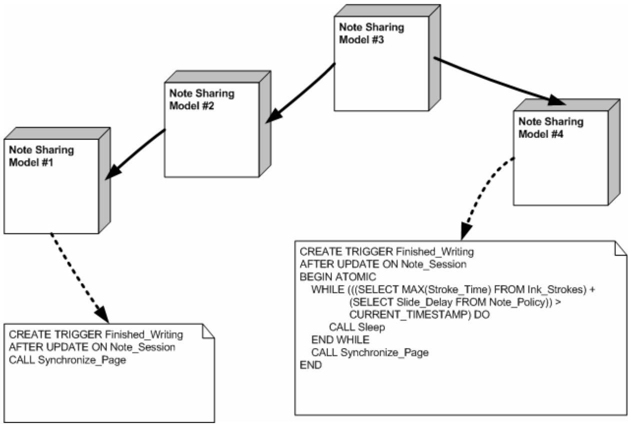
Figure 2: Composition of ink layers showing policy application

4.2. Note Sharing Module

The note-sharing module facilitates the creation of shared sessions in the space repository. Each session has several clients which access a set of ink layers in an organized manner. This module is responsible for selecting a particular page to be displayed from among the set of individual pages that comprise an ink layer. It allows for the current page to be simultaneously changed for a number of users that subscribe to a shared session. The sharing module is also responsible for synchronization among different layers, so that when a shared slide is advanced, a slide may be automatically advanced on the page of local notes.

The note sharing model determines which page layers should be displayed together on a given page. It contains references to remote shared session modules, which allow for the automatic synchronization among clients. To achieve scalability the note-sharing modules may be organized into a hierarchy. Since the modules have the same interface, a module may serve as a controller for other instances. The interconnection of modules is shown in figure 3.

A note sharing controller can be as simple as a single navigation button to advance the current page, and typically includes those controls found in Microsoft PowerPoint or other presentation applications. Policies specify the manner in which slides are synchronized. It is desirable to allow a user who may be creating his or her own notes to select whether or not to automatically synchronize with the current shared session or manually synchronize by explicitly pressing a button to “catch up.” Similar to a database system, we can define triggers that implement policy based on contextual information. Figure 3 illustrates how a trigger could be used to delay the advancing of a slide pending some condition.



```
CREATE TRIGGER Finished_Writing
AFTER UPDATE ON Note_Session
CALL Synchronize_Page
```

```
CREATE TRIGGER Finished_Writing
AFTER UPDATE ON Note_Session
BEGIN ATOMIC
   WHILE (((SELECT MAX(Stroke_Time) FROM Ink_Strokes) +
             (SELECT Slide_Delay FROM Note_Policy)) >
             CURRENT_TIMESTAMP) DO
      CALL Sleep
   END WHILE
   CALL Synchronize_Page
END
```

Figure 3: Distributed note sharing modules showing policy triggers

4.3. Quizzing Module

The quizzing module encompasses functionality to generate, and administer questions based on dynamic student profiles and current contextual information. Ink layers are used for the graphical display of the question as well as a scratch area for students to work out their answers. Quiz questions are organized into sections to facilitate selection of questions related to the current subject matter.

Classroom events such as questions asked and inking of notes trigger changes in the student profile to personalize the learning experience. Analysis of statistical information allows questions to be selected that the data indicates will provide reinforcement learning [14].

The principles of reinforcement learning can be illustrated through collaborative gaming. By defining policies to propagate quiz questions among students, the classroom becomes engaged in collective learning as students challenge one another to answer questions correctly. For example, if a student answers a question incorrectly then the next question will be sent to the team consisting of friends they have worked with in the class, while the correct answer results in the questions being sent to the other students.

4.4. Feedback Module

To make use of the recorded data as well as activity observed in real time, semantic information must be extracted and reported. This functionality is performed by feedback modules. Feedback modules may be application specific, but often perform a generic function such as counting which can be reused.

Applications locate feedback modules using the space repository. Modules interconnect via bridges which present the appropriate aggregated data, including data from other feedback modules. The data is then recorded or displayed to the instructor via an appropriate mechanism. Information displayed would typically include charts of data over time or quantities of categorical data as shown in figure 4. This information is also used to update context information.

Each module can provide valuable feedback through this mechanism. For example an audio/video module can provide information on how many students are speaking or gesturing over time. Survey or quiz results can indicate how well students understand the material. Inking activity can measure student engagement. The questions most often viewed on a class newsgroup may indicate an area of student concern.

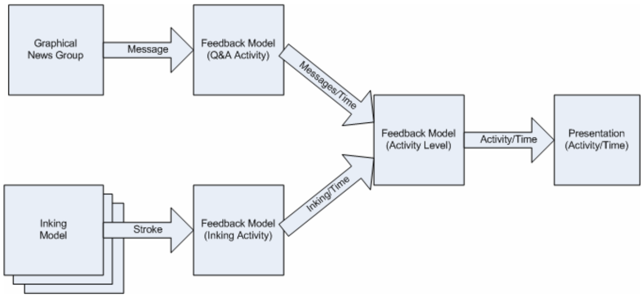

Graphical News Group
Message
Feedback Model (Q&A Activity)
Messages/Time
Feedback Model (Activity Level)
Activity/Time
Presentation (Activity/Time)
Inking Model
Stroke
Feedback Model (Inking Activity)
Inking/Time

Figure 4: Feedback functionality

5. Related Work

A number of educational Learning systems have been developed [6, 10, 3, 17, and 19]. Additional research has focused on utilizing pervasive technologies to facilitate collaboration outside a single setting [22]. [23] incorporates the use of personal and shared spaces to support face-to-face collaboration and Silicon Chalk is a commercial system introduced by [8], which helps build software that enables pervasive learning.

6. Conclusion

In [7] the designers of eClass reflect on their experiences and provide insight into how ubiquitous computing can help solve some of the problems encountered when assessing their work, automated capture and access in the classroom. In order to analyze and validate educational technologies, we must take a holistic approach through pervasive computing.

e-Fuzion differs from all of aforementioned systems in that it is constructed from a distributed component architecture based on a formal model. Ultimately we enable the creation and validation of a pervasive educational system. The traditional pen and paper paradigm is reinforced by our system while taking advantage of pervasive technologies.

7. References

[1] Abowd, G. D. and Mynatt, E. D., Charting Past, Present and Future Research in Ubiquitous Computing, ACM Transactions on Computer-Human Interaction, Vol. 7, No. 1, March 2000, Pages 29 –58.

[2] Al-Muhtadi, J., Ranganathan, A., Campbell, R., and Mickunas, D., Cerberus: A Context-Aware Security Scheme for Smart Spaces. In IEEE International Conference on Pervasive Computing and Communications (PerCom 2003), Dallas-Fort Worth, Texas, March 23-26, 2003.

[3] Anderson, R. J., Anderson R., Simon, B., Wolfman, S. A., VanDeGrift T., Yashuhara, K., Experiences with a Tablet PC Based Lecture Presentation System in Computer Science Courses, Proceedings of the 35th SIGCSE technical symposium on Computer science education, p. 56-60, 2004.

[4] Anderson, R. J., Hoyer, C., Wolfman, S. A., Anderson, R., A study of digital ink in lecture presentation, Proceedings of the 2004 conference on Human factors in computing systems, p.567-574, 2004.

[5] Bargeron D. and Moscovich, T., Reflowing Digital Ink Annotation, Proceedings of the conference on Human factors in computing systems, p. 385-393, 2003.

[6] Brotherton, J. A. 2001, eClass: Building, Observing and Understanding the Impact of Capture and access in an Educational Setting, Ph.D. Thesis, College of Computing, Georgia Institute of Technology, Atlanta, GA, December, 2001.

[7] Brotherton, J. A. and G. D. Abowd, Lessons learned from eClass: Assessing automated capture and access in the

classroom, ACM Transactions on Computer-Human Interaction (TOCHI), Vol. 11 Issue 2, p. 121-155, June, 2004.

[8] Coatta, T., & Kaufmann, R., A New Platform for Pervasive Learning, World Conference on Educational Multimedia, Hypermedia and Telecommunications, p. 742-745, 2003.

[9] W. G. Griswold, R. Boyer, S. W. Brown, T. M. Truong, E. Bhasker, G. R. Jay, and R. B. Shapiro, ActiveCampus - Sustaining Educational Communities through Mobile Technology, Technical Report CS2002-0714, Computer Science and Engineering, UC San Diego, July 2002.

[10] W. G. Griswold, R. Boyer, S. W. Brown, and T. M. Truong, ``A Component Architecture for an Extensible, Highly Integrated Context-Aware Computing Infrastructure", 2003 International Conference on Software Engineering (ICSE 2003), May 2003.

[11] Hess, Christopher K., The Design and Implementation of a Context-Aware File System for Ubiquitous Computing Applications, Ph.D. Thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 2003.

[12] http://www.w3.org/TR/InkML.

[13] Jarrett, R., Su, P., Building Tablet PC Applications, Microsoft Press, 2003.

[14] Kaelbling, L. P., Littman, M. L., Reinforcement Learning: A Survey, Journal of Artificial Intelligence Research 4, p. 237-285, 1996.

[15] Krasner, Glenn E. Stephen Pope, A cookbook for using the model-view controller user interface paradigm in Smalltalk-80, Journal of Object Oriented Programming, 1(3):25-49, 1988.

[16] McKeachie, W. J., McKeachie's Teaching Tips, Houghton-Mifflin Corporation, 1999.

[17] Ning, Hai, Williams, J. R., Slocum, A. H., Sanchez, A., Tablet PC Enabled Design-Oriented Learning, Seventh International Conference on Computers and Advanced Technology in Education, Kauai, Hawaii, USA, August 2004.

[18] Peiper C., Chan, E., Campbell, R., Bresler, J., Al-Muhtadi., J., Expanding Education through Active Space Collaboration, PerCom Workshops 2004, p. 236-240.

[19] http://www.cs.brown.edu/research/graphics/research/ReMarkableTexts.

[20] Román, Manuel, An Application Framework for Active Space Applications, Ph.D. Thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 2003.

[21] Román, M, Hess C. K., Cerqueira, R, Ranganathan, A., Campbell, R. H., and Nahrstedt, K., Gaia: A Middleware Infrastructure to Enable Active Spaces, IEEE Pervasive, vol. 1, pp. 74-82, 2002.

[22] Schraefel, M. C., Hughes, G. V., Mills, H. R., Smith, G., Payne, T. R., Frey, J., Breaking the book: translating the chemistry lab book into a pervasive computing lab environment, Proceedings of the 2004 conference on Human factors in computing systems, Vienna, Austria, p. 25-32, 2004.

[23] Sugimoto, M., Hosoi,K., Hashizume, H., Caretta: A System for Supporting Face-to-Face Collaboration by Integrating Personal and Shared Spaces, In Proceedings of CHI2004, Vienna, Austria, pp.41-48, 2004.

[24] Tim Wentling, C. Peiper, P. Bristow, and J.H. Park, Validation of e-Fuzion: a Wireless Classroom Technology, Proceedings of the World Conference on E-Learning in Corp., Govt., Health & Higher Ed., Vol. 2003, Issue 1, pp. 343-350, 2003.