Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pervasive and Mobile Computing 2 (2006) 1–36
pervasive and mobile computing
www.elsevier.com/locate/pmc

# Embedding expression: Pervasive computing architecture for art and entertainment

Jeff Burke[a,*], Jonathan Friedman[b], Eitan Mendelowitz[a], Heemin Park[b], Mani B. Srivastava[b]

[a] *HyperMedia Studio, School of Theater, Film and Television, University of California, Los Angeles, 102 East Melnitz Hall, Los Angeles, CA 90095-1622, United States*
[b] *Networked and Embedded Systems Laboratory (NESL), Electrical Engineering Department, University of California, Los Angeles, 6731 Boelter Hall, Los Angeles, 90095-1594, United States*

Received 15 March 2005; received in revised form 30 June 2005; accepted 1 July 2005
Available online 6 September 2005

**Abstract**

Pervasive computing's historical driver applications include environmental monitoring, safety and security, home and office productivity and guided experience of cultural activities. It also suggests and offers the means to achieve new types of expression in art and entertainment, which has not been a significant area of research despite its cultural and socio-economic importance and unique requirements. This paper discusses motivations and requirements for a pervasive computing architecture for expression, and it presents a specific approach being developed at UCLA in collaboration among research groups in engineering, theater, film and television.
© 2005 Published by Elsevier B.V.

*Keywords:* Entertainment; Art; Pervasive computing; Expression

## 1. Motivation

> "A valid order accommodates the circumstantial contradictions of a complex reality. It accommodates as well as imposes. It thereby admits 'control and

* Corresponding author. Tel.: +1 310 794 5358; fax: +1 310 825 3383.
*E-mail addresses:* jburke@hypermedia.ucla.edu (J. Burke), jf@ee.ucla.edu (J. Friedman), eitan@hypermedia.ucla.edu (E. Mendelowitz), hmpark@ee.ucla.edu (H. Park), mbs@ee.ucla.edu (M.B. Srivastava).

1574-1192/$ - see front matter © 2005 Published by Elsevier B.V.
doi:10.1016/j.pmcj.2005.07.002

> spontaneity', 'correctness and ease'—improvisation within the whole. It tolerates qualifications and compromise."—Robert Venturi, *Complexity and Contradiction in Architecture* [76].

The networking of distributed, embedded computational devices—pervasive computing—has attracted wide attention and considerable research effort in the past decade. Percom has been envisioned as transforming environmental monitoring, home/office productivity, and health and security applications, as well as supporting experience of museums and public spaces. In various articulations, it aims to be *ubiquitous, pervasive, unremarkable, ambient, calm,* both *tangible* and *invisible*—it is always intended to be *functional*. Yet what of *expressive*? Venturi's text is about buildings, not technological systems, but the thought is apropos to computing approaches that, like architecture, must serve both functional and expressive purposes. In each of its traditional application domains, Percom is designed to be aware of its real-world operating context; in art and entertainment, it is employed explicitly to help *create* that context. Therefore, relevant Percom architectures must both accommodate and impose (achieve and suggest) both new function and new expression. Here, we present one vision for that opportunity and initial results from collaboration between the Networked and Embedded Systems Laboratory (NESL) and the HyperMedia Studio at UCLA.

A shift of similar scope occurred in tandem with the rise in the use of personal computers, which enabled increasingly sophisticated digital representations of image, sound and video. This transformed our capacity for the capture, processing, transport and output of media. But the impact of these new digital media did not lie solely in the new functionality they offered to manipulate and distribute information. They provided new 'languages', palettes and opportunities for creative expression stemming from their unique qualities and, tellingly, the tools available for their manipulation [50]. Later, the internet's growth in affluent nations, and recently in those less so, vastly expanded our capabilities for creation, publication, re-use, criticism and distribution, affecting participation in cultural experiences as much as it has driven evolving definitions of productivity in industry.

By mixing qualities of media, physical material and machine in inexpensive, modular components, pervasive computing offers another broad leap in capacity: an expansive, far-reaching shift in the role of computing in artistic expression. Percom's nature—and its encompassing scope—suggests that an architectural metaphor is fitting. We can design pervasive computing architectures to fulfill only functional goals, and like buildings, they will both be interpreted in expressive terms and re-used for expressive purposes. (One need only read [48] or [84] to see this.) Or, we could design architectures that both express and support expression—as one might design a theater. They will not be neutral; they will both *suggest* and *impose*. As with a theater, users of such an architecture will appropriate its functionality, exploit its flexibility, incorporate its particular nature, bring in other necessary elements, and, within it, create something unanticipated, complex and uniquely expressive.

For the Percom field itself, supporting expression in art and entertainment as an 'application domain' prompts new research as well. For example, it requires rapid, incremental deployment of Percom systems by end users, and the ability to operate them while incompletely deployed in prototypes, maquettes and rehearsals. (Early work

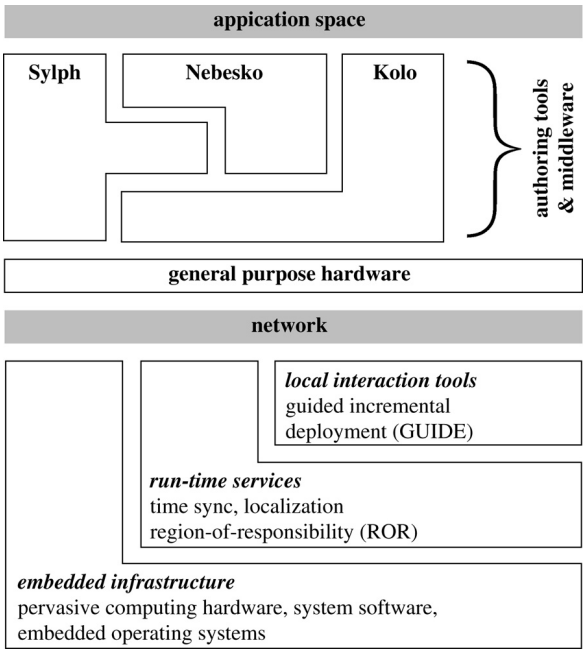



Fig. 1. NESL/HyperMedia Studio pervasive computing architecture.

in this area is described in Section 5.3.1.) It drives performance requirements: NESL's development of scalable high sample rate multilateration for object and personnel localization (Section 5.2.2) is partially motivated by this application. It suggests new needs: Beyond localization, deployment and authoring requirements drive devices to have more internal knowledge of the regions of responsibility of their sensors and actuators (Section 5.2.3), which is also required for coverage models in incremental deployment (Section 5.3.1). Middleware needs are unique, requiring both task-oriented service abstractions as well as access to nuanced control of individual devices. Below, we describe a set of middleware approaches that provide a service-based scheme with declarative interface (Sylph, Section 5.4.2), distributed state management (Nebesko, Section 5.4.3) and simplified interconnection of devices for expressive systems (Kolo, Section 5.4.1). Our current architecture incorporating these components is shown in Fig. 1 and will be discussed in detail in Section 5.

## 2. Roles of pervasive computing

Instead of targeting one monolithic 'cinema of the future' [39] or Piscator's theatrical 'play-machine, a wonder-world, an arena for battling ideas' [59], an appropriate architectural approach will evolve into a judicious (if eclectic) set of modular components, protocols and techniques. While specialized Percom systems will be deployed for expressive purposes, it is also likely that existing networks could be supplemented or

dynamically provisioned to support expressive purposes. By harnessing the modularity envisioned for modern pervasive systems, and providing flexible approaches to their deployment and configuration, Percom technologies can play different roles: they can augment traditional control systems by serving as intelligent **bridges** to typically unconnected devices. They can tune the health and performance of production systems as **proprioceptors** or, as **assistants**, extend the reach, precision and responsiveness of those systems and their human operators. Finally, they can act directly as intrinsic **elements** responsible for new audience–work–artist relationships.

### *2.1. Bridging*

Percom's **bridging** role addresses the question of 'how does one get (system or phenomenon) A to influence (system or phenomenon) B', which is often the first one asked by students working with us to explore expressive uses of technology. Creating interfaces to particular production, building and industrial systems of interest typically poses only implementation challenges and, though out of reach of many artists, does not directly prompt new Percom research. However, existing software architectures for interconnecting disparate control systems in the pervasive computing context have inappropriate abstractions for the highly designed physical environments of interest; aesthetically relevant parameters of the controlled system must be exposed with simple semantics and robust performance [13,57]. These and other limitations have also resulted in too few interfaces between Percom systems and typical arts authoring tools like Macromedia Flash or Director [49] and Cycling 74's Max/MSP [22]. Instead, simple sensor interfaces, such as the EZIO [28], have become pervasive in artistic experimentation because they provide bridges to such software, even as they only scratch the surface of the potential available in current Percom research platforms.

### *2.2. Proprioceptive*

Percom's **proprioceptive** role directly extends monitoring, a key historical motivation of sensor networks, to support system health and equilibrium and ensure precise, repeatable performance in a distributed control system. For the animated feature film *Polar Express* (2004) directed by Robert Zemeckis, the complex performance capture environment (64 infrared cameras in a Vicon motion capture system [77]) required manual calibration three times a day *in a fixed installation* [37]. Percom systems' ability to self-tune will encourage their widespread use in expression, while the domains' use of rehearsals suggests new research is required with respect to online and interactive calibration. This role is not necessarily 'in-the-loop' of control generating an audience's experience; rather, it attempts to ensure the stability and performance of control systems that do directly affect experience.

### *2.3. Assistive*

In an **assistive** role, Percom technologies provide task-oriented *human–network interfaces* to help people manage artistic production processes. They also need not be in the loop of the final experience and therefore may operate post facto, involved in the analysis

of what has already happened, or a priori, planning what will occur, as well as providing real-time decision support. An example from traditional production is the infrared cameras used by a crew to view actors on a dark stage; an analogous Percom system might report on their localization to the various crew members, while also comparing positions to a previous set of rehearsals. Such roles often have unique visualization and user interface challenges because of the dynamic, time-constrained nature of production environments; they have much in common with augmented reality research [6,43,71]. They also can form the precursors to systems enabling truly unique expression.

### *2.4. Elemental*

In an **elemental** role, Percom is directly responsible for both *suggesting* and *achieving* a unique manifestation of expression experienced by audiences or participants. It is actively in-the-loop of 'end-user' experience, causing something to happen or permitting something to be experienced—whether image projection, physical actuation, sound, website update or another phenomenon. These are the most provocative, important and least well-defined roles; they arise from the synthesis of expressive goals with system capabilities. Artist David Rokeby notes, "While engineers strive to maintain the illusion of transparency in the design and refinement of media technologies, artists explore the meaning of the interface itself, using the various transformations of the media as their palette" [62]. To consider Percom devices as tools and materials—as one would a new fabric, a carpenter's fine handtool or the Éclair NPR, the film camera created in the 1960's that sparked *cinema verite* filmmaking—would further turn the field's expressive promise into reality. Beyond the technical requirements described in Section 4, an 'unreasonable care' must be given to exposing clearly what new tools offer while not embedding too many assumptions about the ways in which they will be used [57].

## 3. Application milieu

In all of its roles, Percom helps to fuse the digital and physical realms. Because of digital media, this intersection is already integral to both popular and avant garde art, whether live performance (Section 3.1), themed entertainment and architecture (Section 3.2), new genres (Section 3.3) or film (Section 3.4). An appropriate Percom architecture will take into account current approaches, assist with existing challenges and enable previously unachievable expression.

### *3.1. Live performance*

The first application area we consider, live performance, includes theater, dance, music and many other forms. Works are continually repeated and tuned by their trained participants, sometimes for years, while often seen only once by a given audience. Ranging from the wholly improvisational to the rigidly structured, they offer rich, demanding roles for pervasive computing. Theater and dance artists, as well as musicians, already infuse media into their work on physical stages; they also use digital systems for playback, control and manipulation of sound, lighting, motor and projection effects. They also

have experimented with sensing in several contexts. In the eminent choreographer Merce Cunningham's piece *Biped* (1999), he used motion capture quite differently than in Robert Zemeckis's more familiar movie *Polar Express* (2004). To striking effect, Cunningham juxtaposed live dancers with the projection of non-photorealistic renderings generated from captured performance data [26,46]. Work like his has expanded interest in the performance world in markerless tracking and other long-term computer vision research, which will be supported by pervasive computing infrastructures and have been explored by Sparacino [70] and others.

At UCLA, we have used sensing systems to create otherwise unachievable rehearsal documentation; an example is the diagrammatic summary of performer movement [11]. This study informed further work in the creation of dynamic performance spaces that bridge sensing with the extensive technical infrastructure of modern theaters and highlight an exciting capability provided by Percom. Large productions may use hundreds of individually dimmed lighting fixtures, additional automated lighting with RGB color mixing, intensity, pan and tilt control, as well as many speakers, wireless microphones, projection and motor systems. When such production equipment is interconnected and linked to Percom-based observing systems using an appropriate architectural approach, new possibilities for dynamic stage design arise.

Figs. 2 and 3 show a commercial localization device embedded in a witch's staff from our Department of Theater's production of *Macbett* in 2001. This piece experimented with localizing performers and their props, using movement data to influence design elements controlled via bridged large-scale production equipment in 'real-time' [10]. We developed tools to create simple, elemental relationships among sensed action and the stage environment. For example, sound could be selected, amplified, processed and routed to outputs based on actor movement or gestures with an instrumented prop like the staff. Similarly, theatrical-scale lighting could be influenced by performers' proximity to one another during a scene or by short-time features of their movement.

Though promising in their linkage to the physical world, traditional Percom architectures and their associated programming tools are not yet designed to support such detailed device control or stage designers' authoring approaches. These early experiments suggest additions to the palettes of directors and designers offered by Percom's capability to enable the complex technical infrastructures of modern theaters to be more responsive to the people performing within them. In the future, Percom-enabled production environments could go beyond existing concepts of performance and be explicitly constructed to evolve from one production to the next based on what is observed, or to incorporate audience demographics and activity [14].

### *3.2. Themed entertainment, museums, and architecture*

Museum exhibit design already incorporates media, websites, databases, and artifacts within physically navigable structures. Many of Percom's interesting elemental roles in this domain are like those suggested for live performance, but with visiting participants instead of trained professionals (actors) interacting via Percom with dynamic, media-rich, physical spaces. The HyperMedia Studio's past work in this area is described in [78]. Others have explored immersive computer graphics based experiences for museums [68], while



Fig. 2. Staff used by a witch in the UCLA Department of Theater's production of *Macbett* in 2001.




Fig. 3. Embedded device for localization of a witch's staff used in *Macbett* (see Fig. 2).

assistive functions of related technologies have been discussed in [30,69] and elsewhere. Museums, like theme parks and most architecture, are usually experienced through self-directed navigation with various degrees of assistance; even the most controlled and monitored situations, theme parks, are designed for informal and exploratory participation. Museum goals for social interaction and repeat visits suggest a key new capability provided by Percom will be the recording of individuals' history of action, and the subsequent coordination of experience based on this data as well as explicit navigation choices. A body of Percom work relating to museum experience, (e.g., [30]) has centered on handheld devices as interfaces to customized content. RFID is already being deployed in theme parks for logistical uses [25] and is sure to expand to customization of experience in the future. The ability to customize media based on crowd flow, direct interaction with artifacts, and demographic information offers unique new expressive possibilities for information presentation in these contexts. Also, the emergence of systems that can record distributed, collective participation and re-integrate it with physical architecture and artifacts through media suggest that new museum experiences will emerge. They will be built up of both contributed and curated components, paralleling a provocative shift proposed to the museum community: to move from being 'about something' to being 'for someone' [81].

Similar in nature but larger in scale, themed entertainment and architecture present a clear need for scalability, reliability and robustness. Like museums, they serve as venues for individual artistic works that may incorporate Percom, as well as being expressive works in their own right. They may last into the decades under continuous use, and have functional requirements that vary greatly for 'inhabitants' versus 'visitors'. These environments, including amusement parks and other location-based entertainment, already use sophisticated combinations of media and control systems synchronized to live performance and sometimes responsive to direct participation or online exploration. More broadly, in architecture, Percom will already reach into the sophisticated, digitally controlled mechanical and electrical systems of modern buildings, even as architects already experiment with the incorporation of media and other computer-controlled design elements. For example, see the work of Diller+Scofidio architects [7] and Art+COM in Berlin [4].

### *3.3. New genres and gaming*

'Media' and 'net' artists already create artworks not fitting easily into traditional categories using surveillance technologies, sensing, embedded computing, novel displays, interfaces and building systems, often from a perspective rooted in the visual and sculptural arts but also drawing from many other fields like theater, film and television [48,84]. The typically smaller scale of this work (in comparison with that of the previous section) provides immense freedom for exploration in how new interfaces and embedded, distributed computing will affect society and expand its capabilities for expression. However, the correspondingly tight resources of those working in this field require that appropriate architectures exist to support exploration with a low cost of entry. Along with the unique capabilities of digital media and the internet, artists are exploring what is usable now—typically, simple digital interfaces to sensors and actuators—to explore

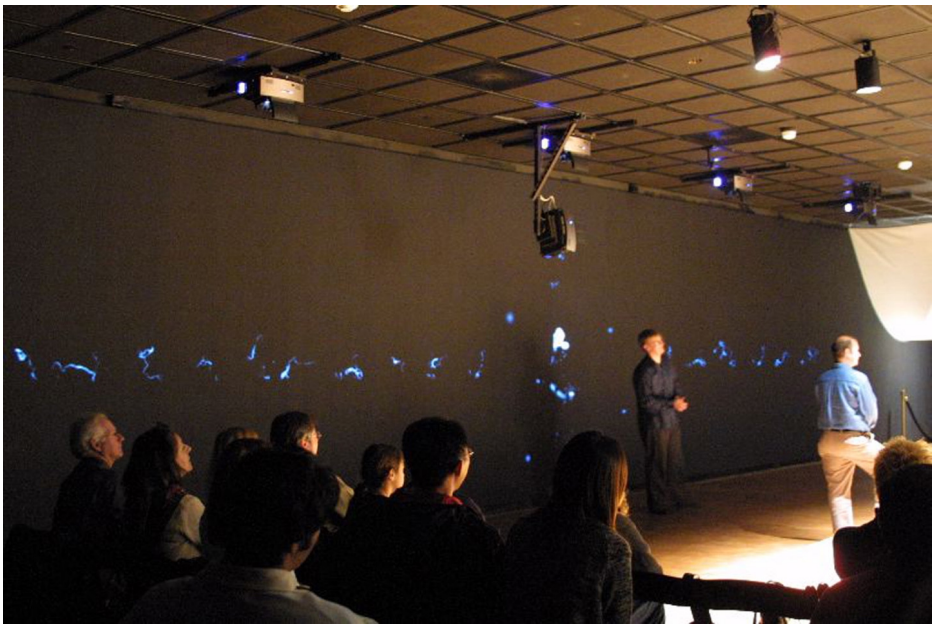


Fig. 4. View of the immersive installation work *Ecce Homology* at its premiere in 2003 at the Fowler Museum of Cultural History.

new ‘audience’ roles, to create works that evolve based on environmental phenomena, and to interweave media with physical objects, often in ‘site-specific’ works. Work by Fabian Wagmister and others at the UCLA HyperMedia Studio [78] greatly influenced the architectural approach developed here. The most recent test of the Kolo middleware (Section 5.4.1) was in the media artwork *Ecce Homology* (Fig. 4) [83]. This artwork enabled gallery visitors to interact with a large (42′ wide) display wall showing a visualization of human and rice genomic data. It integrated a five display OpenGL rendering system with computer vision for user interaction and animation of the BLAST algorithm used in bioinformatics (see Fig. 4).

Gaming, a related area that does have immense technical production capacity, would also benefit from Percom. This growing industry, whose revenues in 2003 eclipsed box-office ticket sales in United States [41], anticipates many future elements of the broader entertainment industry. It incorporates novel interfaces, multi-user experiences and film/media segments, and draws from the traditional arts to create interactive experiences sharing much with the new genres described above. Film director Ridley Scott (*Alien, Blade Runner, Gladiator*) told the *New York Times*, “[t]he idea that a world, the characters that inhabit it, and the stories those characters share can evolve with the audience’s participation and, perhaps, exist in a perpetual universe is indeed very exciting to me” [41]. Games are composite user-driven experiences that can incorporate simulation, time-based media, multi-user interaction, physical spaces and interfaces, and mobile devices. As computing blends into the environment and gaming simultaneously seeks new audiences,

game creators will likely apply elements of both Percom and production systems to create experience in the physical environment. An early example of such a system explored at UCLA is the robotic gaming project, Ragobot, described briefly in Section 5.1.3.

### *3.4. Film, television, and 'mobile media'*

Unlike the forms described above, the conceptualization, deployment of equipment, and recording or generation of action for media production all focus on creating a final unassailable piece of media *seen from one perspective*—typically, a film. We see Percom's clearest initial function in production as assisting in the generation of that fixed form rather than playing a new elemental role in its experience. Research into the assistive roles of Percom will provide the foundational platforms and experience to better develop the unique elemental roles it will enable in the future. In UCLA's *Advanced Technology for Cinematography* (ATC) project [12], we are researching systems to assist the on-set production of live action films as a way of motivating Percom research sensitive to filmmaking processes and to uncover these elemental roles. ATC initially focuses on distributed observation and decision support [51] and methods to record the conditions in which shots are created [72] and assist with recreating them.

We envision that, in the future, Percom will play a unique enabling role by providing metadata supporting media production for *multiple perspective experiences*, in which media assets are dynamically 'edited' together on-the-fly as experienced by viewer-participants. Precursors to such experiences can already be found in new DVDs, games, new genre works, themed entertainment, and certain types of performance. Today, modern feature films, which have only one final perspective, have no broad-ranging system to assist filmmaking teams in managing the physical conditions of production and embedding records of those conditions within the media, despite their large scope, complexity and budget. Because of the prevalence of unconnected manually operated systems in film production, the more technically complex the shot, the less flexible the shooting process, allowing fewer different shots to be attempted in a given period and requiring more repetition to coordinate performances and technical elements. (Fig. 5, from a UCLA student film shoot, illustrates how many people may be involved in one relatively simple shot.) Human beings must control each piece of equipment carefully as a scene progresses, while recording its state and that of each prop and actor in case the scene must be recreated for a later shot or manipulated in post-production. When tight coupling of live action and computer graphics (CG) is required, even more precision from the cast and crew may be necessary to ensure spatial and temporal coordination with CG elements.

An example of such coordination in live action is when, during a shot, an actor must 'hit' a succession of marks on the ground to be in focus while many elements move around them. For complex shots, movement through space between such marks drives so many different manually executed processes that it is often impossible to adapt them to variations in performance or directed improvisation. In cases where automation is used, equipment is typically controlled by key frame or other stored-preset approaches. Percom systems could, for example, bridge personnel localization with interfaces to this equipment, allowing the automation to respond flexibly to changing conditions and on-the-fly creative choices. In a complex tracking shot, the cinematographer could tell the system to shift



Fig. 5. Film location shoot by students in the UCLA Department of Film, Television and Digital Media. (Courtesy of Adam Martin.)

camera angle dynamically based on actors' actual positions, in order to maintain a certain type of framing as the camera and actors move according to the director's instructions. Alternately, a Percom-based decision support system could enable a camera operator to manually achieve even more nuanced control. Such decision support might be provided in real time, or for review after shots have been taken, as with the Augmented Recording System (ARS), described below.

### *3.5. Example tool — Augmented Recording System*

The Augmented Recording System (ARS) is an example of an *assistive* tool built in collaboration between NESL and the HyperMedia Studio to support media production using the components described in Section 5. (Architecturally, it functions at the topmost layer of Fig. 1.) It is designed to create 'augmented footage' that combines a synchronized data record with a film camera's privileged visual record [27,31]. The capabilities that it explores—for example, synchronization of sensor networks with traditional media capture and playback systems—form a foundation for more elemental roles in next generation media production. For now, we have developed it as part of the ATC project's initial goal of supporting traditional production. In that project, Percom devices are deployed to gather incident light measurements, sound levels, or localization and orientation information about actors, props and equipment. The basic elements of ARS include (1) these wireless nodes that have sensors to sample interesting physical phenomena, a processor and a radio

to transmit the sampled data, (2) a timecode source that generates a reference timestamp for each film or video frame, and (3) a video camera or film package that can record a scene and audio track (used for timecode) simultaneously. The current version of ARS collects data from mote-class devices, synchronizes with video frames, and can record and represent augmented footage. For data gathering, the Berkeley MICA1 mote was used in our first implementation; ARS is now ported to the MICA2 platform to take advantage of the more stable radio and faster processor. In a base station computer, a Java-based serial port server runs, collecting data from gateway motes and interpolating for missing frames, then forwarding the data to the *Sylph* middleware [19], which handles registration, multiplexing and high-level query translation. The current implementation [51] uses *Kolo* to distribute timecode through the local area network as described in Section 5.2.1 and to distribute sensor data from the gateway to other subsystems that also require it.

ARS is motivated by the observation that sensor data recorded during media production are typically most useful if they can be analyzed with knowledge of the corresponding film (or video) frame in the recording. This requires time synchronization of the Percom devices gathering the data with the timecode typically used as a global clock in media production. (This signal is often sourced by an external timecode generator, a device that generates a reference time clock that is distributed as an analog signal among devices including camera and video tape recorders.) For experimentation, we have used a standard SMPTE timecode generator providing a 29.97 frames per second signal although any frame rate can be integrated into our system. Data acquisition from each sensor must also be performed at least as fast as the refresh rate of the application's media. This allows each data point to be measured during the exposure time for the corresponding frame, so that the visual media record will be in synchronization with the measurement.

New techniques are required to achieve the necessary performance on mote-class devices, which have the appropriate form factor for our application. The data-sampling period for typical frame rates is less than 33.37 ms (i.e., the 29.97 picture per second NTSC standard for television); this is relatively short compared to other sensor network monitoring applications. (Film typically uses twenty-four frames per second, though special effects photography may use much faster or slower rates.) As it is impractical to synchronize each device every 33.37 ms due to network latency, ARS uses local timers in each mote whose periods are equal to the media frame period. Each device sends packets which contain data for a pre-defined number of frames to a base station. Then, the base station synchronizes the sensed data with video frames by mapping local timer values in the packets and external timecode signals in a post facto fashion.

To synchronize devices, we adopted a simple broadcast-based synchronization method in the initial implementation of ARS. This method was easy to implement on a single hop network, and we believed that it would be possible to synchronize at an accuracy of the video frame rate. Selection of the scheme was based on two factors—that there exists an upper bound on the time drift for the motes and that broadcast messages are received at approximately the same time at all receivers. Based on the upper bound on time drift between motes gathered experimentally, we set the update period to ensure that the synchronization error does not exceed our selected period of one video frame. A gateway mote broadcasts time synchronization commands to neighboring motes every synchronization period.

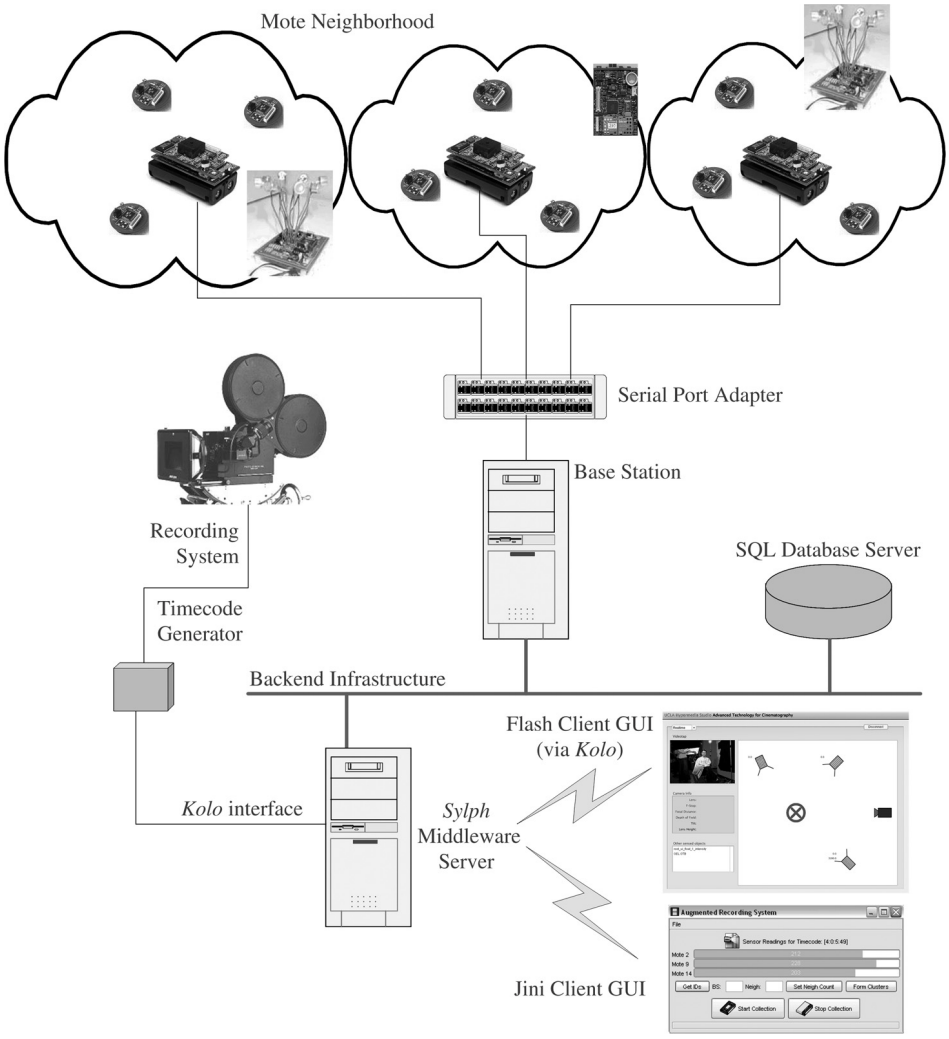



Fig. 6. Augmented recording system (ARS) system diagram.

As we aim to make ARS feasible to use on sets and to generalize it for use in live performance and other environments, we have developed graphical user interfaces to start and stop recording of the data as well as to search and browse the augmented footage. These applications provide the synchronized display of the data corresponding to each frame with the media from a shot (video proxies in the case of film) during playback. Currently, we have developed two different interfaces.

A Java client adhering to Jini standards, one window of which is pictured in Fig. 6, supports both end-user and developer functionality, allowing system developers to create clusters of sensor nodes, find their neighbors and to start and stop collection of data.

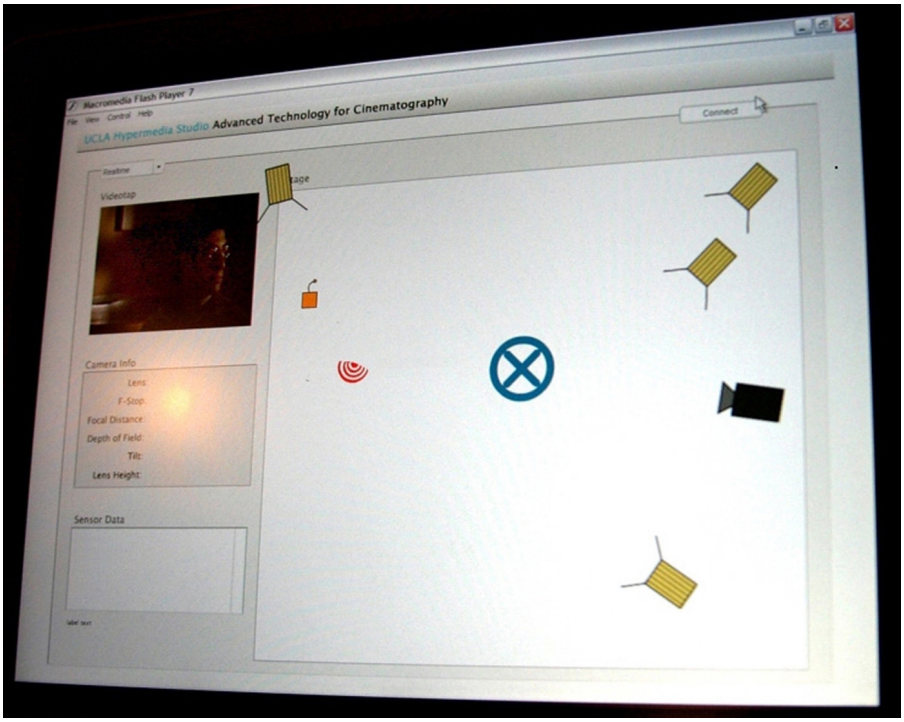



Fig. 7. Macromedia Flash interface to display of ARS measurements of localization and light information delivered on a tablet PC; upper left hand corner shows streaming video of either live or recorded video matching the displayed data.

A Macromedia Flash based user interface for a tablet PC (shown in Fig. 7) was created as a prototype for end users on the film set. With the help of the Sylph middleware, ARS can respond to many expressive queries that Sylph supports. Parsing and processing of the declarative queries are done by the proxy core of the Sylph middleware. Kolo provides data conduits between system components (e.g. Mica mote gateways, timecode generator, GUI and database).

There are still several platform issues to be addressed in order for the ARS to support daily use in media production. With resource constrained nodes like Mica motes, at most up to 10 nodes can be deployed before experiencing data frame losses. We are seeking solutions to increase the number of deployable nodes in various ways: adopting new hardware platforms such as MicaZ motes [21], which have higher radio bandwidth, networking via multi-hop communication and hierarchical network architecture, and exploiting in-network processing such as compression techniques. Additionally, since packet propagation delay may vary and thus the broadcast-based synchronization is not as stable as we require, we plan to adopt TPSN (discussed in Section 5.2.2) for time synchronization. Finally, as lighting is crucial information for cinematography, the ATC project team will replace the light sensors supplied with the MICA2 sensor board with higher precision and dynamic range sensor modules.

## 4. Requirements

The general requirements presented below orient our current joint architectural research. They are based on our experience in collaborating on the development of ARS and other components of the *Advanced Technology for Cinematography* project, and our two groups' past experience in engineering research and artistic production. Each application domain—each art form—balances expressive and functional goals with a focus on 'end-user' experience. For Percom systems to successfully play assistive, proprioceptive, bridging and elemental roles in creating that experience in these demanding domains, we feel they need to address the following issues. Underlying them is the premise that Percom must both *achieve* old goals more effectively and *suggest* completely new possibilities. In the following sections, we have attempted to focus on system performance goals only where necessary as an enabler of broader requirements and as a point of departure for spontaneity, improvisation and unexpected juxtaposition—what Venturi terms 'messy vitality' in the context of another type of architecture.

In expression, the concepts and control of rhythm and scale, whether temporal, visual, or otherwise, are fundamental; Percom systems need to have both *a 'sense of space'* and *a 'sense of time'*. They must be *easily integrated with media*. They must embrace their role in *fusing eclectic sets of components* to create behaviors that cannot be motivated in task-oriented frameworks. (While ARS supports the 'tasks' of production, its descendant applications that will be in the loop of control, will surely be tasked according to the creative goals of a shot, not the administrative reality.) However, they should also support assistive tasks with appropriate semantic abstractions, while *allowing abstraction layers to be set aside* when fine-tuning of control is required. As tools, they must make their capabilities clear in their interfaces and design, and eventually provide sufficient tracing functions to allow discovery of the stimuli generating a particular system behavior. For now, at a minimum, they must *guide their own deployment* to make it easier to assemble robust systems that have these basic capabilities. We consider these requirements at design-time and discuss them below. They complement existing requirements for historical applications of Percom, such as power management and service discovery. (While there are new tradeoffs and constraints to explore, we do not address these here due to the breadth of existing and ongoing work.) In the future, Percom systems should become better able to *operate while incompletely deployed*, *learn by example* and *adapt to repetitive action given certain performance goals*; that is, they will need to mature in place. These run-time requirements are mentioned here as topics of our future work motivated by the application areas discussed above. The nature of rehearsals and design development in these domains requires such an ability to robustly support the iterative design of physical systems.

### *4.1. Sense of 'time'*

The capacity for rhythm and coordinated action begins with synchronization. Time synchronization across all devices in a Percom system already provides synchronization to external clocks and a host of other familiar low-level services: distributed signal processing, time-of-arrival calculations, beamforming, time stamped sampling, time

division multiple access (TDMA), and duty cycling, among many others. Existing models of time synchronization can be placed into four categories: internal, external and hybrid synchronization, and virtual clocks [33]. For example, ARS, described in Section 3.5, requires time synchronization between an external timecode generator (clock signal) and the sensor nodes so that sensor data can be synchronized with recorded film or video frames. ARS exploits both external and hybrid approaches to achieve post facto synchronization of data acquisition in mote devices at high sample rates for that platform.

We believe that a 'sense of time' should build upon synchronization. An architecture for expressive applications could go further, and provide information on system confidence that 'rhythm' is kept (synchronization is maintained). For example, to support media production's demanding physical circumstances, Percom systems' future proprioceptive role will require monitoring confidence in system time synchronization. (ARS would also benefit from this capability.) Devices will need to maintain calibration and global synchronization during shifts in the environment or adjustments to the deployment. Where possible, they will have to anticipate impending failure or conditions that might cause the loss of synchronization.

At the application level, coordinated action requires that the physical outputs controlled by Percom devices in an environment can be reliably synchronized; feedback or better internal device models may be necessary to coordinate synchronized *experience* in this way. For example, in theater, a stage manager might initiate a sound cue a half-second after a lighting cue to make them seem in synchronization, because of incandescent lighting's slower rise time. This is a function of each cue's programmed fade time, the number of lights controlled and the sound cue's volume (thus the perceptible difference in intensity at each step of the fade), as well as other environmental conditions. While the stage manager may already have effective time synchronization with each of her nodes via a broadcast protocol—instructions spoken over an intercom to human operators—her sense of time must be much more nuanced to create the experience desired; often it is adjusted heuristically on-the-fly based on actors' performances. Such nuanced but quantifiable sensibilities commonly understood in expressive domains should inspire the future design of Percom systems and be implemented (even incompletely) not just to *support* what is already done but *suggest* what might be possible.

### *4.2. Sense of 'space'*

Similarly, humans' sense of space—and thus their consideration of space in artistic expression—exceeds simply positional knowledge. While we consider precise high sample rate localization of objects, devices and people to be an enabling capability for many expressive roles and early applications for Percom, there is much more to be done in addition to this difficult known problem. Robotics researchers quickly discovered that single sensor, single modality information is insufficient to obtain a truly functional understanding of an environment [8]. Likewise, Percom systems must use multiple modalities to ensure proper deployment and provide proprioceptive function. For example, a light sensor on a film set should be able to distinguish the difference between being placed face-down and being placed in a truly dark room, or between spatial graduation among peer devices and calibration mismatch.

More broadly, devices should understand not only their deployment location, but the points in space for which events are visible to them or their actuators can influence. We term the volumes in physical space corresponding to possible fields of view of a device's sensors and the 'reach' of its actuators and outputs, the device's *region of responsibility* (RoR). Several areas of work at NESL incorporate and support this approach [45]. Sensor beam-forming uses predictive statistical techniques to determine when spatially diverse sensors are likely to have observed the same event. Correlation of common events leads to the construction of RoR coverage models. Further work examines distributed calibration by pair-wise differentials to common events. The final step in RoR delivery is the notion of detection given a multisensor, multimodality environment. NESL is currently developing 4D cross-correlation models for optical events. Once developed, such device-level coverage models enable guided deployment (described in Section 4.5) as well as proposed proprioceptive roles. Further work will be required for typical output devices, such as video projectors, lights and speakers, which will be Percom bridged and act over a distance with typically well-known performance characteristics. For such devices, embedding RoR models would allow them to be easily addressed or queried based on both their location and the coverage region of their perceived output. Handling these models at the device level would allow in-network collaboration to take advantage of this information directly, allowing neighborhoods to be created based on spatial capability and responsibility rather than simply location. Tasking of two different display types, for example, a projector and a monitor, could be done in a manner consistent with the location of their output image rather than their electronics.

In achieving this overall 'sense of space', localization remains a central capability. Our experience [10,12,13,51,78] and goals for Percom suggest that expressive applications incorporating physical spaces will place extremely high demands on localization system performance, especially with respect to accuracy. (See Section 5.2.2 for further discussion.) Localization supporting expressive applications must seek to (1) minimize power consumption without trading off very high accuracy and sampling rates, (2) have small physical size, allowing the technology to be hidden if desired, (3) provide immunity from external forces imposed by the movement of performers or objects tracked by the device, (4) resolve positions with great enough resolution that error cannot be visually perceived *by the intended audience*, (5) achieve stability (repeatability) in the face of both repetitive action and stasis, (6) operate quietly while performing well in what are often infrared-saturated and loud environments, (7) scale to many localized objects operating near each other, and (8) scale in overall coverage to support environments with many occlusions.

### *4.3. Interface flexibility: Fusing eclectic sets of components*

Our work at the HyperMedia Studio has required us to create interfaces to a wide variety of devices, ranging from large scale production systems [13] to arrays of simple sensors and actuators [78]. *Kolo*, described in Section 5.4.1, was created to address the lack of middleware available to support this interconnectivity in a simple way. While we expect that systems programmers will generally need to construct low-level interfaces on behalf of larger collaborative teams, the incorporation of atypical or unexpected devices

into a Percom framework should be well-supported by its overall approach. This is essentially a restatement of the bridging role of Percom (Section 2.1) as a *requirement* for the other roles to be successful. For example, more so than in traditional genres, bridging is elemental in the vital experimentation occurring within new art genres. The implementation of unexpected connections between devices and phenomena form a starting point for much experimentation. For situations with small technology development budgets, as is common in many artistic endeavors, these experiences are often created, rehearsed, tested and maintained by non-engineers. By creating Percom architectures allowing easy implementation of unanticipated simple network-based interconnections, the research community could prompt vast experimentation in this area that would feed back into new application ideas [57].

### *4.4. Drill-down: Allowing abstraction layers to be set aside*

In Section 3.1, we suggested that current abstractions also do not support nuanced control of bridged systems, such as theatrical lighting, once an interface has been created. This suggests a complementary requirement to the one above. Traditionally, to make Percom software problems tractable, domain-specific tasks and abstractions are often assumed, limiting future use of the intrinsic capabilities of devices. While the ability to query 'is the light on in the theater' is necessary for certain assistive and management functions (and is supported by *Sylph*), the design of lighting in a particular scene requires individual adjustment of many hundreds of lighting levels, which is supported by *Kolo*. No matter how much assistance may be provided, fine control will still be required in certain circumstances and may not need high-level abstraction to be useful. Designers using traditional entertainment systems are very familiar with using many individual I/O points that they must address with very simple control metaphors (e.g., timed interpolation between presets). An appropriate Percom approach will allow both access to the individual light intensities, while also allowing aggregate queries and manipulation at higher levels of abstraction. Though distributed systems pose greater challenges, traditional multimedia authoring tools like Macromedia Flash and Director partially achieve this in single software packages by layering graphical authoring metaphors on top of more general scripting languages.

### *4.5. Guided deployment*

Currently, even simple Percom systems take a long time to deploy and configure. Equally problematic for expressive applications, they often cannot be redeployed without systems engineers, even after they are designed. Deployment must become less time-consuming if the Percom applications described in Section 3 are to be explored broadly. Art and entertainment have rehearsal and prototype processes that accommodate set-up time and experimentation but need capabilities for incremental function and low-complexity re-deployment. This indicates they would benefit greatly from real-time assistance in deployment of devices, providing verification of appropriate RoR and necessary wireless coverage, for example.

Film production, live performance and new genres will typically require regular rapid deployment, with longer times possible in museum exhibit design and the nearly

permanent installations imagined in architecture. Solutions for guiding deployment that allow tradeoffs between deployment time and quality of service across this broad spectrum will eventually be required if a common platform is to be developed. Once such solutions exist, the consideration of guided deployment at other points in the set-up time/QOS space could be explored. For example, Percom technologies could aid rapid deployment and verification of systems in touring performances. These events move regularly between venues and often have very little time to calibrate technically complex systems, while having immense creative and financial pressures demanding their flawless performance. In both proprioceptive and explicitly assistive ways, Percom systems integrated with production equipment would allow such environments to be more effectively deployed and rapidly tuned, but they themselves must be easily deployed.

## 5. Approach

Using components created at UCLA, we are developing a flexible and modular system architecture that addresses the design requirements of Section 4 and will provide an avenue to explore the run-time requirements envisioned for future work in Section 6. The architecture is illustrated in Fig. 1 and supports different types of configurations required by expressive applications. Connectivity in the diagram is vertical between blocks. The many paths from hardware to services and application layers directly address the flexibility requirements of Sections 4.3 and 4.4. The lower part of the figure shows the layers of the Percom infrastructure, which in addition to a set of software services currently includes hardware in three broad categories: mote class (MICA motes), super mote class (iBadge and MK2; see Section 5.1), and micro server class (e.g., Intel Stargate). The embedded hardware platforms run Sensor OS (SOS), TinyOS, or in the case of microservers, the Linux operating system. SOS is an operating system for mote class and super-mote class wireless sensor networks developed by NESL [67]. Because of its core support for dynamically loadable firmware modules and their wireless distribution and verification, SOS is well-suited to handle the incremental deployment and rapid retasking of nodes required in Percom systems for expression. The ability to install new features, replace defective ones and modify others that already exist, without having to physically interact with the devices, or be concerned with their exact locations and quantity, supports proprioceptive management envisioned in Sections 2 and 3.

Basic *run-time services*, such as time synchronization and localization, are built on top of these operating systems and address the ‘senses’ of space and time (Sections 4.1 and 4.2). *Local interaction tools* use these services at the device or Percom infrastructure level rather than via a general purpose machine elsewhere in the network. An example of this is NESL’s guided incremental deployment (GUIDE) tool, which is installed on each mote or super-mote device and does not require connection to higher-level middleware. It operates in a distributed fashion using only previously deployed nodes as its infrastructure to guide the deployment of each additional node, and is motivated by the requirements of Section 4.5 and described in more detail in Section 5.3.1.

The upper sections of Fig. 1 show the suite of middleware, which offer various paths from application to embedded infrastructure; they are discussed in Section 5.4. Together,

they provide the varying levels of abstraction and basic services required for Percom's different roles. Accessing devices via Sylph, for example, offers a greatly simplified data-aggregation interface that facilitates management and monitoring, while applications primarily in need of bridging can use Kolo to manage the data flow between devices. Where coordinated high level control over many distributed Kolo-connected processes are required, Nebesko provides an augmented FSM-based scripting language. This provides a platform to study the more nuanced application-level synchronization discussed in Section 4.1.

### *5.1. Percom infrastructure (hardware platforms)*

Numerous pervasive computing hardware platforms exist, most notable in the United States are Berkeley/Crossbow's MPR (Mote) family of platforms [21]. Expressive applications are generally well-served by typical design goals of light weight, small size, and low power hardware platforms with various sensing capabilities. NESL and the Center for Embedded Networked Sensing (CENS) have developed two platforms, the iBadge and MK2, which include the same general sensing application model as Motes, but incorporate additional capabilities that are proving useful in expressive applications. An additional platform, the Nerve, is being developed with art and entertainment applications considered explicitly.

#### *5.1.1. iBadge*

The iBadge was developed in NSF's Smart Kindergarten (SmartKG) project, which explored uses of Percom devices for assessment in early childhood education [47]. (See Fig. 8.) The goal of SmartKG was to assist parents and teachers in comprehensively assessing student learning and social interaction. Design requirements for the classroom have much in common with those for the art and entertainment applications described above. The device is multi-modal, used by non-professionals, physically robust and easily deployed. The iBadge houses a sensor suite capable of measuring light, acceleration, orientation, position (via ultrasonic techniques), pressure, temperature, sound, and magnetic fields with varying degrees of fidelity. It is small enough and light enough that it can be worn comfortably. It has both a low-power Mote-class radio and a Bluetooth transceiver. Its microcontroller is supplemented by a DSP, allowing it to perform tasks such as speech-recognition [58]. This platform is being used for localization of people and large equipment in the Advanced Technology for Cinematography project; the next generation being designed (Section 5.1.3) will incorporate additional support for the requirements above based on the HyperMedia Studio's experience with localization systems [13] and discrete sensors [78].

#### *5.1.2. MK2*

The MK2 platform (see Fig. 9) is designed to act as fixed infrastructure, and does not have the extensive integrated sensor suite of the iBadge. Prior small, low-power platforms like motes [60] lacked sufficient processing capability for complex position estimation and data manipulation. The MK2 supplements a limited microcontroller with a more advanced 40 MHz ARM Thumb CPU [65]. Networks can use slightly larger, lower-cost devices such as the MK2 as beacons and base stations without the size, weight and power constraints

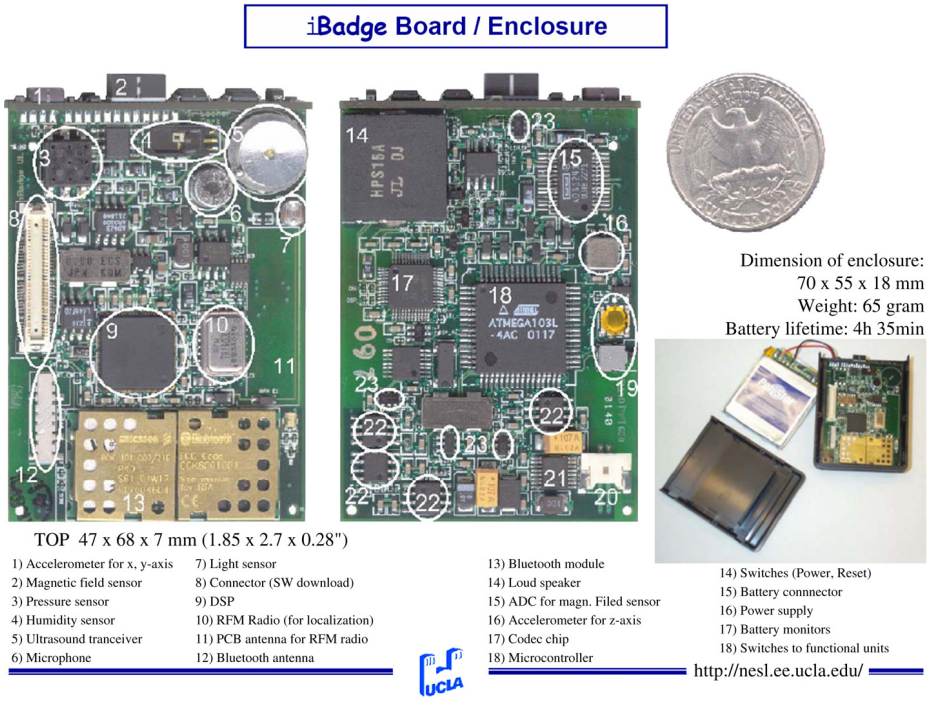



Fig. 8. The UCLA NESL iBadge platform.



Fig. 9. The UCLA NESL MK2 platform.

of the other platforms. The additional in-system mathematical power that is available in the MK2 increases precision of calculations contributing to the outcome consistency and reproducibility. For expressive applications, such devices act as human–network interfaces and ‘responsiveness’ is desired as a baseline capability in all phases: deployment, rehearsal and training, and run-time.

#### 5.1.3. Nerve/Ragobot

Our exploration of expressive applications using the MK2, iBadge, Mote, and other platforms has uncovered hardware requirements that are not addressed in either typical sensor networks or broader pervasive computing research. Both consider long operational life per battery charge cycle, robustness in the face of limited connectivity, and reliability in the face of single-point failures. They less consistently address common constraints for expressive applications including visual subtlety, operability in harsh thermal environments, fast sampling, low latency and high data bandwidth, tight accuracy and calibration, tight time-synchronization, user simplicity, varying infrastructure, and rapid deployment requirements. This combination of constraints, driving future generations of the iBadge and MK2, has also prompted NESL/CENS to develop two new platforms: the Nerve and the Ragobot. The Nerve's development includes a direct collaboration with the Hypermedia Studio and is designed specifically with the bridging, assistive, and proprioceptive roles in mind. The system is capable of high volume data acquisition, transport, and processing to fulfill the technical requirements of Section 4. Ultimately, it is hoped that this system will come to represent an elemental component of new expressive media. It is equipped with facilities for digital signal processing/filtering, low-latency data acquisition, and wireless communications capable of high sustained data rates. The DSP capability supports localization; traditional Percom systems often lack the processing power to perform the complex volumetric analysis required for localization and are forced to round to less precision in calculations, decreasing positional granularity. Therefore, high accuracy Percom systems must have hardware facilities for performing advanced computations.

Nerve is part of a modular controlled-mobility (self-propelled) pervasive computing system called the Ragobot being developed at NESL/CENS [61]. This emerging mobile sensor network platform is designed to enable a real-world actuated gaming environment in which users task robots to achieve goals in a media-rich and physically challenging terrain. Exploring an application space related to both gaming and themed entertainment, the system integrates external actuation, mobile autonomous robots, and media projection. In addition to other Percom considerations including fine-grained energy awareness, the robots must coordinate proprioceptive tasks (e.g., battery charging), be bridged to communicate and coordinate with other systems in the environment and maintain an elemental presence as the central 'actors' of the experience even as they are controlled by user-participants.

### 5.2. Run-time services

Running on these hardware platforms are a set of basic *run-time services* (see Fig. 1), including time synchronization, localization, and region of responsibility (coverage) modeling. These services address the 'senses' of space and time. They support application-level authoring on general purpose machines via the middleware described in Section 5.4, while enabling local interaction tools that provide important user services within the Percom network itself.

#### 5.2.1. Time synchronization

Forming the initial basis of the 'sense of time' envisioned in Section 4, time synchronization is a run-time service required by many other components, including

most localization approaches. Complex and computationally intensive protocols like the Network Time Protocol (NTP) [23] can be used in general-purpose networks to synchronize the clocks of computers forming the leaves of the network. (In ARS, servers synchronize using NTP, and distribute timecode synchronization information among them as an offset of the global time clock of the workstations.) However, NTP is not suitable for energy constrained pervasive computing environments. Ganeriwal et al. proposed the Timing-sync Protocol for Sensor Networks (TPSN) [35], which is simple, flexible, and scalable for energy constrained pervasive computing nodes. TPSN's future use in ARS is described at the end of Section 3.5. The central idea of TPSN is to synchronize the sender's clock after exchanging message between a pair of sensor nodes. Assuming that the clock drift and propagation delay does not change during two-way message exchange, the sender can calculate relative clock skew between the two nodes. To reduce indeterminism of synchronization errors, TPSN time stamps the packets in the Medium Access Control layer. Network-wide time synchronization can be achieved by traversing a directed cyclic graph formed by the nodes' pairwise synchronizations. TPSN has been implemented on Berkeley Mica motes and has an accuracy of 17 and 8 μs on Mica1 and Mica2 motes, respectively [33,35].

*5.2.2. Localization*

The basis of a 'sense of space', localization is being explored extensively at NESL in new development. For example, the new Nerve platform (Section 5.1.3) will incorporate support for localization addressing the requirements outlined in Section 4.2. Previous experience in *Macbett* and the ATC project [51] suggests personnel tracking in many expressive applications will require at least 25 positional-fixes-per-second at cubic centimeter precision or better. In choosing our localization approach, other known techniques were analyzed in light of typical requirements of the expressive domain.

For example, correlating the strength of a received radio signal (RSSI) to the distance from the transmitter is one technique explored by many [2,44]. Unfortunately, RSSI techniques do not approach our requirements for precision as the correlation process is quite difficult due to variances among devices in manufacturing tolerances, battery level, and antenna orientation. Current work in infrared (IR) imaging, target tracking, and localization has emphasized either low hardware complexity at the cost of low accuracy [5,15] or high accuracy at the cost of intense mathematical complexity, calibration, maintenance, and expense [38,52]. The former systems do not provide the required positional accuracy, while the latter do not scale well to a high number of tracked objects, are too physically large, and are generally cumbersome to deploy. Additionally, infrared saturation can render many IR approaches useless and is common in many environments of interest, including the outdoors, as well as theaters and sound stages, which often use thousands of watts of incandescent light.

Our approach uses ultrasonic time difference of arrival (TDOA) ranging [63]. The basic simplicity and accuracy of this approach, the availability of inexpensive hardware components, the immunity of the time measurement to signal strength variations, and the relatively light computational requirements (for distance acquisition) make this type of ranging very well-suited to pervasive computing systems. Preceded by the Medusa node [63], an ultrasonic TDOA approach now in common use is MIT/Crossbow's

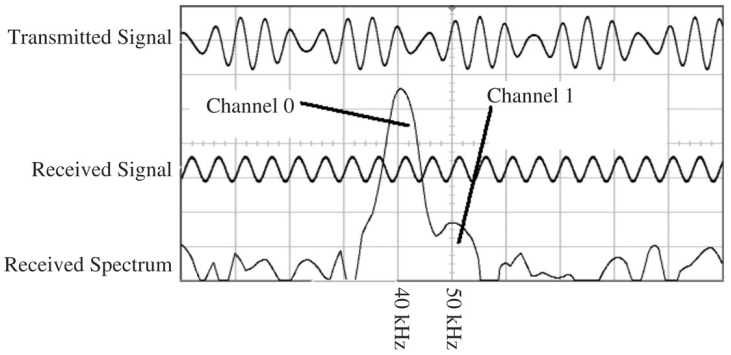



Fig. 10. At the transmitter (top), the two ultrasonic channels are well isolated. At the receiver (bottom) the channels are weaker, but still actionable. The spectrum of the received signal is shown in the background.

Cricket [66], which can provide fine-grained ranging information at an accuracy of approximately one cubic centimeter [22]. TDOA over an ultra-wideband (UWB) radio system has only recently become commercially available [74]. For expressive applications, current UWB systems lack sufficient positioning accuracy, with most commercial systems attaining locations to only within a cubic meter [31]. In the future, it is likely that a combination of ultrasonic and UWB approaches will be very useful in expressive applications, providing a valuable set of tradeoffs among accuracy, coverage and power consumption.

NESL's new ultrasonic TDOA system, demonstrated at Sensys [51], included a specifically engineered fail-safe TDMA ring beaconing protocol (RBP) [32] for sharing the ultrasonic channel. Beacons are the (usually fixed) nodes that form infrastructure for localization of other mobile nodes. Design goals for RBP included (1) high receiver sensitivity, (2) channel isolation, (3) beacon arbitration to allow multiple beacon access given shared spectrum (RBP will be extended to handle multiple channels in addition to its current support for multiple beacons per channel), (4) beacon mobility. NESL's new Nerve platform (Section 5.1.3) will relax the reliance on static infrastructure. In typical systems, beacons must not be moved once deployed. However, many requirements for expressive Percom suggest beacon mobility needs to be supported; for example, the GUIDE tool (Section 5.3.1) envisions a deployment-adjustment loop requiring re-positioning of nodes. Nerve hardware will be able to detect when it has been moved and reacquire its position from the remaining immobile beacons; this is an initial 'sense' of its spatial situation.

Using RBP and the same transducers deployed in the MK2 (Section 5.1.2) and Cricket [66] platforms, we have demonstrated the feasibility of an approach that involves multiple simultaneous ultrasonic channels (Fig. 10). This approach allows the system to be immune to beacon deployment order, able to accommodate dynamically inserted and removed beacons, and capable of functioning given a partial deployment of beacons—features required to support GUIDE (Section 5.3.1). It is important to note that our current implementation of the protocol requires that all beacons be within broadcast range of each other. Ongoing work seeks to relax this requirement. Localization methods used are based on $n$-hop multilateration [63,64] using Minimum Mean Square Estimates

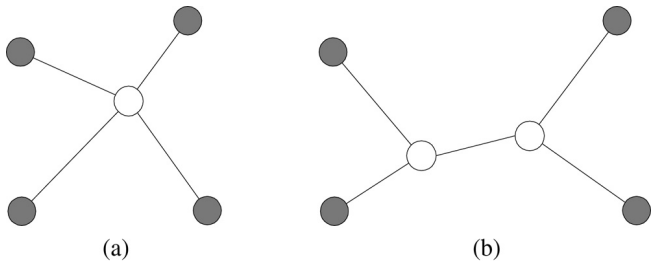



Fig. 11. Multilateration cases: Solid nodes represent beacon nodes who know their position. Open circles are nodes nodes that do not know their position and estimate it.

(MMSE) to minimize the differences in noisy distance measurement environments (See Fig. 11 for examples of one-hop and two-hop multilateration). The ATC project team was able to deploy and maintain an early system using this approach with 10 Hz ranging rate (10 distances/second, which corresponds to 3.33 positions/second) for eight tracked objects. Bandwidth limitations prevented transport of all data out of the basestations, but the new Nerve radio will provide the required additional bandwidth. While significantly faster than earlier work with the platform, the distributed rate of localization demonstrated [51] with these techniques was still significantly slower than expressive media applications might require, as discussed. (The Nerve platform will also provide additional range and accuracy; the test system for the RBP protocol provided +/− 2 cm accuracy at 2.5 m range between beacons and tracked devices.)

The development of the localization prototypes used in [20] reveals another critical constraint for ultrasonic systems that we are focusing on—noise performance. Expressive applications often involve eclectic equipment, including machinery, which can severely interfere with ultrasonic ranging systems. Fig. 12 shows the results of measurements conducted on a computerized motion-controlled camera system located at ImageG Studios, Inc. in Burbank, California, which is pictured in Fig. 13. Strong noise harmonics are visible throughout the entire ultrasonic spectrum. As indicated by Fig. 12, in performance environments with large machinery existing ultrasonic platforms, may be incapable of operating without large errors.

*5.2.3. Region of responsibility*

The concept of providing internal coverage models to Percom devices so that they are able to estimate their regions of responsibility (RoR) for different sensors and actuators arose in examining the requirements of guided deployment. As it also contributes to a spatial context-awareness that can be expressed as three-dimensional regions or volumes, it begins to address our design requirement of a ‘spatial sense’. There are two main considerations regarding the RoR of Percom devices: (1) whether, for a particular modality, the Percom infrastructure covers a given region, or, if not, how to determine the maximal breach path in the region, and (2) where to place Percom infrastructure to ensure that user-defined goals for RoR can be met. As the fiducial point of each region is the actual position of Percom devices, localization is required to calculate RoR. Given knowledge of device locations and a coverage model, the first issue can be resolved by existing techniques

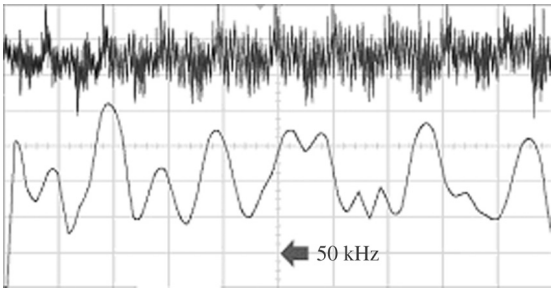



Fig. 12. The ultrasonic noise (top) of a large motion-controlled camera system while translating. The spectral representation of this noise appears below. (Horizontal axis is 10 kHz/division.) The amplitudes of these peaks are large relative to MK2 normal transmit power. These signals were recorded on a sound stage at *ImageG* in Burbank, California.



Fig. 13. Motion control camera system at *ImageG* in Burbank, California.

like [53], where a disk sensing model is assumed and the authors also present methods to find maximal breach paths based on Voronoi diagrams that can be calculated with the information of each sensor. The RoR concept can be further generalized by techniques such as those discussed in [54], which incorporates general non-disk sensing models and total exposure for representing the probability of moving object detection.

As expressive Percom should enable an iterative design process and support operation during incomplete deployment, users should be able to achieve their coverage goals by incrementally deploying Percom infrastructure rather than waiting to see if a satisfactory

set of regions for each modality is obtained at the end of a complete deployment. Much research has been done on planning of RoR for networks of distributed devices before or after deployment [1,9,16,24,40,79,80], but all need knowledge of location. As expressive Percom systems must operate during incomplete deployment, we cannot rely on knowledge of location always being available. In [16,26], the authors formulate coverage problems as combinatorial optimization problems and present methods to find a minimum number of sensors and their locations such that coverage constraints are met and sensor cost is minimized. These methods represent sensor field as a grid of points to reduce the solution search space, which is not practical in a real pervasive computing environment. Bulusu et al. presented adaptive beacon placement methods that incrementally improve initial random deployment [9]; however, the authors assume that the robot or human can determine its geographic position by some means (e.g. GPS). Another approach achieves local improvement through each sensor iteratively seeking (and being moved to) better positions after an initial random deployment [1,40,79,80]. This method requires sensor mobility as well as knowledge of sensor location. As position information is not available, some local interaction tools must exist to help planning RoR during deployment.

### *5.3. Local interaction tools*

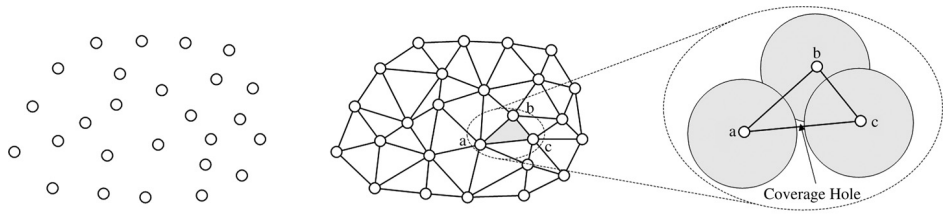
*Local interaction tools* (see Fig. 1) support deployment and bootstrapping. Using run-time services and the resulting 'senses' of time and space, they act on the Percom devices themselves without requiring tasking (authoring) by middleware. One such tool is GUIDE.

#### *5.3.1. Guided-incremental deployment (GUIDE)*

To support rapid, incremental deployment of nodes while meeting coverage and localizability requirements, we are developing a guided incremental deployment tool (GUIDE). For proper performance of many sensor networks of interest, sufficient localizability and coverage must be achieved by their deployment. Currently, performance metrics, such as coverage area (or volume), are only available after sensor deployment is completed. GUIDE aims to break the time consuming deploy-and-evaluate loop thus required by enabling devices to provide incremental feedback during their deployment.

In order to calculate nodes' locations, their deployment topology must satisfy localizability conditions and some nodes (i.e. beacons) must know their locations a priori. One of the simplest ways to determine if a node is localizable is to check if the node can measure its distance to at least three beacons and that those beacons are not collinear. Coverage verification is typically considered to be solved by dense and random deployment (e.g. over-the-air deployment) [1,34,18,85]. This approach is not necessarily realistic for architecture (Section 3.2), where aesthetic constraints and conflicts my limit the placement of beacons, nor is it guaranteed in new genres (Section 3.3) due to budget and equipment constraints. This assumption is especially problematic for sensor deployment in film and media production (Section 3.4), where random and dense deployment is limited because of the difficulty for crews to deploy and re-collect many sensors on the set in a short time while incorporating visibility and equipment coordination constraints.

GUIDE provides feedback to users via simple interfaces on nodes themselves that indicate when their location and orientation allow the network to satisfy minimum




(a) Deployment of Sensors (b) Planar Mesh Construction (c) Coverage Hole Detection

Fig. 14. Three major steps of guided deployment. (a) Each node discovers neighbors based on radio connectivity and distance measurement. (b) Planar mesh is constructed using triangulation technique that divides the region into non-overlapping triangles. (c) Each triangle area becomes the region of responsibility of three nodes and each triangle determines if the performance requirements (e.g., coverage) are met.

performance requirements. Examples of such performance requirements include coverage (RoR) of sensors and actuators, localizability, and connectivity. A typical scenario for the use of GUIDE is as follows: Users deploying devices employ their heuristic knowledge of necessary final coverage to begin deployment, perhaps eventually with assistance from simulation tools, and feedback from each device indicates the best location for it to be placed as it is being added to the network. Percom devices have coverage models (a 'sense' of their own RoR) and ad hoc connectivity to nodes within radio range. Each node estimates the quality of its current deployment based on its local knowledge and indicates this estimation to the person via a simple interface on the device itself, such as an LED display. Based on the feedback, the person adjusts positions of deployed sensors and adds or removes devices for better deployment results. Nodes can be heterogeneous in coverage, communication range and computing power. For complex systems or those with very stringent requirements, traditional performance analysis methods could certainly also be used.

Fig. 14 shows basic procedures to construct an overall RoR for a group of sensors and verify performance, using a disk model of coverage. We assume that distance measurements between neighbors are available by means of some existing ranging technology, such as those discussed in Section 5.2.2. GUIDE can be used with either a incremental or complete sensor deployment, as shown in Fig. 14(a). Based on the distances measured between nodes, a planar mesh is constructed in a distributed fashion. The sensing field is divided into non-overlapping triangular regions of responsibility (RoR). For example, the shaded triangle is the joint RoR of sensors a, b and c. Each node checks if the performance requirements are met. For instance, Fig. 14(c) shows coverage hole detection for the RoR of sensors a, b and c. In Fig. 14(c), shaded circles represent sensing ranges. In this case, the existence of the coverage hole can be easily detected by the fact that the distance between sensors a and c is greater than sum of sensing ranges of sensors a and c. By enabling incremental and interactive sensor deployment and allowing humans to accommodate environmental constraints, this local interaction tool guides users for better deployment quality.

Research challenges arise because each device must rely on the connectivity and distance measurement to its direct neighbors and must assume the network may be

partitioned into several clusters and not be connected to a central computing resource. Howard et al. investigated an incremental self-deployment method [42], and presented an algorithm for robots to discover an unknown environment, maximizing the covered area while maintaining visibility (connectivity) among robots. The main difference from our approach is that the robots are assumed to have a capability of global localization by some means (e.g. GPS) and their coverage computations are done centrally. Our goal is a guidance method for incremental deployment that does not assume that sensor nodes know their locations a priori and can operate in a distributed fashion. Goldenberg et al. proposed a *smart deployment* method for beacon deployment based on a node-localizability test [36]. However, it considers only beacon placement for localizability. For guided deployment to improve sensing coverage, we have developed distributed methods to detect the existence of coverage holes in the region of deployed sensors. We are now formulating coverage (RoR) models for light sensing beyond simple disk models, which will be more useful in media production. Work related to the incorporation of such models in guided deployment is referenced in Section 5.2.3.

### *5.4. Middleware and authoring tools*

For application design, all of the expressive domains described in Section 3 require system interconnection and configuration by non-programmers, the complexity of which we have previously identified as a major stumbling block to experimentation [13]. We use the term *authoring* for this process, to distinguish it from the *systems programming* required to create the components, infrastructure, and the middleware *API programming* for device interface and software component construction. Authoring tools must support the creation of works by individuals and collaborative, multi-disciplinary teams in iterative and modular design processes. Our architecture combines three different middleware components: *Kolo*, a Java-based framework to handle low level control and interconnection (wiring) between eclectic sets of components; *Sylph*, a declarative interface for querying sensor networks using a SQL-like query language, and *Nebesko*, a scripting language developed to add interactive control to Kolo and support distributed state management.

#### *5.4.1. Kolo*

Kolo [56] seeks to address challenges faced by both novice authors and experienced developers in bridging devices for expressive purposes. The Kolo framework is built on a series of simple assumptions: (1) Network access to low latency and high sample rate sensor data (e.g., position information from a localization system) and corresponding control (e.g., of lighting) is baseline functionality for many new artworks; (2) control cannot only be hidden behind abstractions that do not provide access to aesthetically relevant parameters; (3) a useful control interface to most devices used in the expressive domain can be articulated as a collection of named data objects whose primitive data values (numbers, strings, lists, etc.) can reflect device inputs or drive the state of device outputs (this arose from experience in implementing behavior based systems used successfully by designers in 3D artworks [55]); (4) most practical distributed control can be supported in this common namespace by allowing objects in the namespace to have their values read and set in one-time requests or via a publish/subscribe mechanism; (5) by

allowing persistent functional relationships among these distributed objects, many types of interaction, especially for prototyping and experimentation, can be supported this direct 'wiring' within the framework.

Unlike other proposed frameworks for expressive Percom, like ARIA [3], Kolo is intended for the control of multimedia and not content delivery. We have found that *control* over remote playback devices (or processes on a general purpose machine), not media transport, is often a key concern in current development of expressive works, and this is what Kolo is designed to simplify and support. Through a compact Java-based API, developers can extend Kolo to support new sensors or media elements. Kolo then acts as a bridge allowing the interconnection between all supported devices. To promote quick adoption by developers, Kolo's API is designed with relatively few core abstractions, especially when compared to such traditional distributed frameworks such as CORBA [29]. Kolo only requires the understanding of six concepts: knobs, values, subscriptions, groups, relationships, and arbitrators.

The Kolo Network Object (*Knob*) is Kolo's basic building block. Organized into trees, every knob has a name and can be uniquely identified by its path, an ordered list of ancestors' names. Knobs can be accessed from any process on the Kolo network as if they were local and can also be created on a remote Kolo process. Knobs may have *values*, which may reflect and/or affect the internal state of device. One difficulty bridging heterogeneous devices over a network, is data type agreement. This led us to develop a single polymorphic value type for Kolo. When a value is accessed from the Java API the developer must request it in a primitive type, and the framework casts values to the requested type transparently. To expose the functionality of a new sensor or media device to the Kolo network, developers extend existing knob classes.

*Subscriptions* are the only way knob values can be set in Kolo. Knobs receive their value from one or more subscriptions. A knob may subscribe to another knob's value regardless of its location on the network. For example, to control the intensity of a light based on the distance of a tracked actor from the audience, the intensity of the light can be subscribed to the actor's distance knob. To minimize network traffic and/or energy consumption, subscriptions may be time based (periodic), change triggered, or a hybrid of the two.

Elemental uses of Percom often wish to drive collections of individual devices in synchrony, e.g., a number of lights in different physical locations. A Kolo *group* is a knob that maintains a list of members, as a child knob. A group's value is a list that aggregates the values of all its members. Any operation that can be performed on a knob can be performed on a group and the group applies the operation to all of its members. Any operation applied to a child of a group is applied to the children of its members. Thus the intensity child of a group of lights can be controlled with a single subscription.

A knob whose value is defined as a function of its subscriptions is called a *relationship*. For example, a light's intensity attribute can be made equal to the minimum of two actors' distances from their audience by creating a relationship that returns the minimum of its subscriptions and subscribing the light to that relationship. Developers create relationships though the same API as for standard knobs. Like all knobs, relationships can be created remotely, allowing for load distribution. In complex environments, multiple relationships may vie for control of a single attribute [19]. *Arbitrators* are privileged relationships that examine competing subscriptions and arrive at a single value for a knob. For example, the

'Min' (minimum) relationship can be used as an arbitrator. The use of arbitrators allows for a consistent and well-defined manner for dealing with conflicting relationships.

A peer-to-peer network of *KnobManagers* handles Kolo's network functionality. A KnobManager acts as a daemon for each Java process and supports the (1) creation, referencing, and destruction of knobs, relationships, and groups, (2) the creation and termination of subscriptions, and (3) the assignment and de-assignment of arbitrators. Every Kolo process must have at least one KnobManager and each KnobManager must be uniquely named. When a knob is created, its presence is broadcast to all KnobManagers, which are collectively responsible for maintaining a consistent namespace. KnobManagers allow developers to work with remotely managed knobs as if they were local objects. Additionally, KnobManagers attempt to minimize network traffic by aggregating subscription data when possible.

Building on interface ideas explored in non-networked applications by Phidgets [83], we are also developing graphical tools to monitor, simulate, and manipulate the Kolo network. Working prototypes include the Kolo Network Browser, which allows users to interactively browse the Kolo network and make subscriptions. Additionally we are using Kolo's support for Max, a visual programming language, to prototype visual approaches to authoring expressive Percom systems.

#### *5.4.2. Sylph*

Kolo is ideally suited to facilitate experimental and elemental roles of Percom. Both larger scale experiences and assistive applications can make use of higher level data query support and the 'service' based approach of *Sylph*, the middleware developed for NSF's SmartKG project described above in Section 5.1.1. Sylph provides a database capability for recording, archiving, searching and playing back events in a media rich environment. This includes a bootstrapping process in which configuration files and/or dynamically discovered sensor resources are registered in a discovery module (currently Jini). 'Information services' are described in the Jini registry and a uniform interface is provided to control and obtain data from a wide variety of sensors. A GUI allows users to define new recording sessions and/or browse stored sessions and control playback. Facilities exist for playback of audio-visual media coordinated with flexible display of other sensor data, e.g., an object's location superimposed on a floor plan of the space. The query language provides special constructs for controlling sensors, including sample rate, filtering, thresholding, duration, etc. Through access to Kolo's name space, Sylph can be integrated with devices to which Kolo is interconnecting.

Sylph uses three types of components according to each abstraction layer: *sensor modules*, *service discovery modules*, and *a proxy core* [17]. Sensor modules for each sensing infrastructure (e.g. bluetooth and Mica motes) are responsible for communicating with the sensor hardware directly. Service discovery modules advertise available sensors through lookup service to client applications. The proxy core processes application queries and manages data between sensor modules and service discovery modules. With this modular design, client applications and application designers can use sensor hardware through a simple interface with high-level queries.

A query language was developed for the Sylph middleware to support retrieval of the stream of sensor data [80]. With the query language, user and application

can specify sensor data value to return, transformations to apply, sampling rates, and duration. A basic form of a query is: "READ ⟨data tuple⟩ [WHERE ⟨predicate clause⟩] [EVERY ⟨time interval⟩] [FOR ⟨duration⟩]". An example query of returning temperature and pressure value from environmental sensors every minute for an hour is: "READ temperature, pressure EVERY 1 min FOR 1 h".

The Sylph system handles registration, multiplexing and higher-level query translation for pervasive computing. Sylph is particularly well suited for expressive Percom because of it supports multiple types of sensor nodes and allows one to write minimal new Jini code for each new type of sensors, due to its modular nature. At a low level, Sylph is based on using SET and GET operators on attributes. Once one defines an appropriate Sensor Module for their sensors and its attributes, it is easy for a Jini client to change or get these attribute values. For example, in the Augmented Recording Sensor Module, we define attributes such as light, period and command. To begin sending data, the user interface (i.e. Jini client) queries a service discovery domain for sensors and then issues queries to each of them of the form "READ LIGHT EVERY 30 SECONDS". Such a query gets translated to SET and GET requests—"SET PERIOD 30 SECONDS" and "SET COMMAND=STARTSENDING". Thus, Sylph provides a common method for user interfaces to interact with sensor nodes.

#### *5.4.3. Nebesko*

Iterative creation cycles, which are required for expressive Percom, are traditionally well-supported by scripting languages. Nebesko is a high-level scripting language built on Kolo that complements Sylph and commercial multimedia authoring packages, including Max/MSP and Director (see Section 2.1), for which Kolo also has interfaces. Nebesko is an interpreted language based on an augmented finite state machine that allows runtime creation and manipulation of Kolo interconnections [56]. While Kolo already provides interfaces to existing tools used by artists, Director via an 'xtra' and Max/MSP via collection of 'objects', neither of these platforms was designed for authoring expressive works using Percom systems. Significantly, they do not support the considerable amount of distributed state management required. Nebesko fills that gap.

Authors using Percom to build expressive experiences encounter many of the same problems faced by game developers when creating interaction with a large number of objects distributed in a virtual environment; challenges which other gaming-specific scripting languages attempt to solve: "The major complication in C/C++ based AI and game logic programming lies in dealing with events that take a certain amount of game time to complete, and with events which are dependent on aspects of the object's state. In C/C++, this results in spaghetti-code that is hard to write, comprehend, maintain, and debug" [73]. By its similar management of state across the eclectic and distributed networks enabled by Percom, Nebesko aims to support elemental expressive uses of Percom.

## 6. Conclusion

We have motivated a set of requirements for pervasive computing that are sensitive to the goals and processes involved in several art and entertainment domains. Also, we have presented a Percom architecture that begins to incorporate these sensitivities and

address their unique research challenges, and discussed a sample application, ARS, using services available in the architecture. We have discussed both related and resulting work in hardware platforms, localization, deployment, middleware and authoring tools, as well as other elements of the architecture.

Throughout, we have suggested future research driven by the same overall requirements. If anything, these requirements may seem mundane in light of the application domain. In their thoughtful paper on 'unremarkable computing', Tolmie et al. quote Weiser [75], "For thirty years most interface design, and most computer design, has been headed down the path of the 'dramatic' machine. Its highest ideal is to make a computer so exciting, so wonderful, so interesting, that we never want to be without it. A less-traveled path I call the 'invisible'; its highest ideal is to make a computer so embedded, so fitting, so natural, that we use it without even thinking about it" [82]. A third way, another expressive role of digital technology, can be found in the same collections of invisible, 'unremarkable' machines coordinated not simply to engender productivity, comfort or safety as they disappear, but to create remarkable and sublime expressive experience. To fade into the background has never been enough; in a letter about color filmmaking in 1948, Sergei Eistenstein wrote

> "this viewpoint holds that good music in a film is that which you do not hear; that good camera work is that which is unobtrusive; and that good direction is that which you do not notice. As regards colour, this viewpoint holds that in a good colour film you are not conscious of colour. To my mind, this viewpoint, raised to the level of a principle, is a reflection of creative impotence, of an inability to master the complex of cinematic expressive means needed to make an organic film. . . I repeat that master here means ability to develop each element of the expressive means to the utmost, at the same time orchestrating, balancing the whole so as to prevent any particular, individual element from undermining the unity of the *ensemble*, the unity of the compositional whole" [27].

## Acknowledgements

The authors thank Ryan Dorn and Jen Roach of the UCLA HyperMedia Studio, who assisted in manuscript preparation, Maija Burnett for graciously reviewing several drafts, and Kaitlin Drisko for much discussion about the relationship between function and expression. The Smart Kindergarten, Augmented Recording System, and their constituent components, as well as Kolo, are based on research funded in part by the National Science Foundation (NSF) and the Center for Embedded Networked Sensing at UCLA. Nebesko's development is partially supported by the National Endowment for the Arts, and the Advanced Technology for Cinematography Project is partially supported by the Intel Research Council.

## References

[1] S. Adlakha, Designing sensor networks — a density perspective, Master's Thesis, Department of Electrical Engineering, University of California, Los Angeles, 2002.

[2] C. Alippi, G. Vanini, Wireless sensor networks and radio localization: a metrological analysis of the MICA2 received signal strength indicator, in: 29th Annual IEEE International Conference on Local Computer Networks, November 2004, pp. 579–580.
[3] ARIA Research, http://aria.asu.edu/research.htm.
[4] Art+COM website, URL: http://www.artcom.de.
[5] T. Aytac, B. Barshan, Differentiation and localization of target primitives using infrared sensors, IEEE/RSJ International Conference on Intelligent Robots and System, vol. 1, 2002, pp. 105–110.
[6] R.T. Azuma, A survey of augmented reality, Presence: Teleoperators and Virtual Environments 6 (4) (1997) 355–385.
[7] A. Betsky, K. Michael Hays, L. Anderson, Scanning: The Aberrant Architectures of Diller + Scofidio, Whitney Museum, New York, 2003.
[8] R.A. Brooks, A Robust Layered Control System for a Mobile Robot, IEEE Journal of Robotics and Automation 2 (1) (1986) 14–23. Also MIT AI Memo 864, September 1985.
[9] N. Bulusu, J. Heidemann, D. Estrin, Adaptive beacon placement, in: Twenty-first International Conference on Distributed Computing Systems, ICDCS-21, April 2001, Phoenix, AZ, 2001, pp. 489–498.
[10] J. Burke, Dynamic performance spaces for theater production, Theatre Design & Technology, U.S. Institute of Theatre Technology 38 (1) (2002) 26–35.
[11] J. Burke, Interactive performance environments and the visualization of actor movement, in: Visual Interaction and Design, Digital Creativity 13 (2002) 122–128 (special issue).
[12] J. Burke, W. McDonald et al., Advanced Technology for Cinematography project, University of California Los Angeles. http://hypermedia.ucla.edu/atc.
[13] J. Burke, E. Mendelowitz, J. Kim, R. Lorenzo, Networking with knobs and knats: Towards ubiquitous computing for artists, in: Ubiquitous Computing 2002, Concepts and Models Workshop, September 29–October 2, 2002, Gothenburg, Sweden, 2002.
[14] J. Burke, J. Stein, Theatre of context, in: New Visions of Performance: The Impact of Digital Technologies, 2004.
[15] M. Burton, S. Acton, Target tracking using the anisotropic diffusion pyramid, in: Proc. of the IEEE Southwest Symposium on Image Analysis and Interpretation, April 1996, pp. 112–116.
[16] K. Chakrabarty, S.S. Iyengar, H. Qi, E. Cho, Grid coverage for surveillance and target location in distributed sensor networks, IEEE Transactions on Computers (December) (2002).
[17] A. Chen, R.R. Muntz, S. Yuen, I. Locher, S.I. Park, M.B. Srivastava, A support infrastructure for the smart kindergarten, Pervasive Computing, IEEE 1 (2) (2002) 49–57.
[18] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, K.K. Saluja, Sensor deployment strategy for etection of targets traversing a region, Mobile Networking and Applications 8 (4) (2003) 453–461.
[19] M. Coen, B. Phillips, N. Warshawsky, L. Weisman, S. Peters, P. Finin, Meeting the computational needs of intelligent environments: The Metaglue system, in: Proc. of MANSE '99, 1999.
[20] A. Crabtree et al., Organizing a mixed reality game 'on the ground', in: Proc. of the 2004 Conf. on Human Factors in Computing Systems, Vienna, Austria, 2004, pp. 391–398.
[21] Crossbow, Inc., Cricket Project Schematics. http://nms.lcs.mit.edu/projects/cricket/software2.0/, 29 October, 2004.
[22] Cycling 74 website, URL: http://www.cycling74.com.
[23] D.L. Mills, Internet time synchronization: The network time protocol, in: Z. Yang, T. Anthony Marsland (Eds.), Global States and Time in Distributed Systems, IEEE Computer Society Press, 1994.
[24] S. Dhillon, K. Chakrabarty, S. Iyengar, Sensor placement for grid coverage under imprecise etections, in: Proceedings of International Conference on Information Fusion, 2002.
[25] L. Dignan, J. McCormick, Riding Radio Waves, in: Baseline Magazine, May 1, 2004.
[26] A. Dils, The ghost in the machine: Merce Cunningham and Bill T. Jones, Performing Arts Journal 70 (2002) 94–104.
[27] S. Eisenstein, Colour film, in: B. Nichols (Ed.), Movies and Methods, vol. 1, University of California Press, 1976.
[28] Ezio website, URL: http://www.ezio.com.
[29] C. Faver, G. Stein, Towards a digital stage architecture: a long-term research agenda in digitally enabled theater, IEEE MultiMedia (October–December) (2001) 6–9.

[30] M. Fleck, M. Frid, T. Kindberg, E. O'Brien-Strain, R. Rjani, M. Spasojevic, From informing to remembering: unbiquitous systems in interactive museums, IEEE Pervasive Computing (April–June) (2002) 14–21.
[31] R. Fontana, S. Gunderson, Ultra-wideband precision asset location system, in: IEEE Conference on Ultra-Wideband Systems and Technologies, Baltimore, MD, May 2002.
[32] J. Friedman, S. Han, Y.-C. Kuan, I. Locher, S. Park, A. Savvides, RBP Localization codebase, NESL CVS. Available http://cvs.nesl.ucla.edu/cvs/viewcvs.cgi/Localization_RBP/, November 2004.
[33] S. Ganeriwal, J. Elson, M.B. Srivastava, Time synchronization. Wireless sensor networks, Technical Report, November 2004.
[34] S. Ganeriwal, A. Kansal, M.B. Srivastava, Self aware actuation for fault repair in sensor networks, in: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'04, 2004.
[35] S. Ganeriwal, R. Kumar, M.B. Srivastava, Timing-sync protocol for sensor networks, in: ACM Conference on Embedded Networked Sensor Systems, SENSYS, 2003.
[36] D. Goldenberg, A. Krishnamurthy, W. Maness, Y.R. Yang, A. Young, A. Savvides, Network localization in partially localizable networks. To appear in INFCOM, 2005.
[37] M. Goldman, Performanced captured, in: Millimeter Magazine, 2004.
[38] D. Hallaway, T. Hollerer, S. Feiner, Coarse, inexpensive, infrared tracking for wearable computing, in: Seventh IEEE International Symposium on Wearable Computers, 21–23 October, 2003, p. 69.
[39] M. Heiling, The cinema of the future, Presence 1 (3) (1955) 279–294.
[40] N. Heo, P.K. Varshney, A distributed self spreading algorithm for mobile wireless sensor networks, in: Wireless Communications and Networking, WCNC, 2003.
[41] L.M. Holson, In land of Oscars, rich new category is Best Video Game, in: New York Times Business/Financial Section, April 10, 2004.
[42] A. Howard, M.J. Matari'c, G.S. Sukhatme, An incremental self deployment algorithm for mobile sensor networks, Autonomous. Robots 13 (2) (2002) 113–126.
[43] R. Hull, J. Reid, E. Geelhoed, Creating experiences with wearable computing, IEEE Pervasive Computing (October–December) (2002) 56–61.
[44] X. Ji, H. Zha, Robust sensor localization algorithm in wireless ad hoc sensor networks, in: Proc. Computer Communications and Networks, October 2003, pp. 527–532.
[45] A. Kansal, E. Yuen, W.J. Kaiser, G.J. Pottie, Sensing uncertainty reduction using low complexity actuation, in: ACM Third International Symposium on Information Processing in Sensor Networks, IPSN, April 2004. (TR-UCLA-NESL-200404-02).
[46] A. Kisselgoff, Lincoln center festival review: A youth of 80 kicks up his heels, in: New York Times Section E, Part 1, July 23, 1999.
[47] I. Locher, S. Park, A. Savvides, M.B. Srivastava, System Design of iBadge for Smart Kindergarten, winner of the 2nd place at the 39th DAC Student Design Contest, 2002.
[48] M. Lovejoy, Digital Currents: Art in the Electronic Age, 3rd edition, Routledge, New York, 2004.
[49] Macromedia website, URL: http://www.macromedia.com/.
[50] L. Manovich, The Language of New Media, MIT Press, Cambridge, 2002.
[51] A. Marianantoni, H. Park, J. Friedman, V. Holtgrewe, J. Burke, M. Srivastava, F. Wagmister, W. Kaiser, W. McDonald, J. Brush, Demonstration: sensor networks for media production, in: Second ACM Conference on Embedded Networked Sensor Systems, SENSYS, 3–5 November 2004, Baltimore, Maryland, 2004.
[52] P. Maybeck, D. Mercier, A target tracker using spatially distributed infrared measurements, IEEE Transactions on Automatic Control AC-25 (April) (1980) 222–225.
[53] S. Meguerdichian, F. Koushanfar, M. Potkonjak, M.B. Srivastava, Coverage problems in wireless ad hoc sensor networks, in: IEEE Infocom 2001, vol. 3, 2001, pp. 1380–1387.
[54] S. Meguerdichian, F. Koushanfar, G. Qu, M. Potkonjak, Exposure in wireless ad hoc sensor networks, in: International Conference on Mobile Computing and Networking, Rome, Italy, 2001, pp. 139–150.
[55] E. Mendelowitz, The Emergence Engine: A behavior based agent development environment for artists, in: Proc. Twelfth Conf. on Innovative Applications of Artificial Intelligence, IAAI, 2000, pp. 973–978.
[56] E. Mendelowitz, J. Burke, Kolo and Nebesko: A distributed media control framework for the arts, in: First Intl. Conference on Distributed Frameworks for Multimedia Applications, DFMA '05, 6–9 February 2005, Besançon, France, 2005.

[57] W.J. Mitchell et al. (Eds.), Committee on Information Technology and Creativity, National Research Council. Beyond Productivity: Information Technology, Innovation and Creativity, National Academies Press, Washington, DC, 2003.
[58] S. Park, I. Locher, M.B. Srivastava, Design of a wearable sensor badge for smart kindergarten, in: 6th International Symposium on Wearable Computers, ISWC2002, 7–10 October 2002, Seattle, WA, 2002.
[59] E. Piscator, The theatre can belong to this centure, in: J. Gassner (Ed.), Producing the Play, Dryden Press, New York, 1941.
[60] J. Polastre, R. Szewczyk, C. Sharp, D. Culler, The Mote Revolution: Low power wireless sensor network devices, in: Proceedings of Hot Chips 16: A Symposium on High Performance Chips, 22–24 August 2004.
[61] Ragobot Project Website, UCLA, http://www.ragobot.com/.
[62] D. Rokeby, Transforming mirrors: Subjectivity and control in interactive media, in: Simon Penny (Ed.), Critical Issues in Electronic Media, State University of New York Press, Albany, NY, 1995.
[63] A. Savvides, C.C. Han, M.B. Srivastava, Dynamic fine-grained localization in ad hoc wireless sensor networks, in: Proceedings of the International Conference on Mobile Computing and Networking MobiCom, Rome, Italy, July 2001.
[64] A. Savvides, H. Park, M.B. Srivastava, The bits and flops of the N-Hop multilateration primitive or node localization problems, in: Proceedings of the First International Workshop on Wireless Networks and Applications held in conjunction with Mobicom 2002, 28 September, Atlanta, Georgia, 2002.
[65] A. Savvides, M.B. Srivastava, A distributed computation platform for wireless embedded sensing, Invited paper, to appear in The Proceedings of ICCD 2002, Freiburg, Germany, 2002.
[66] A. Smith, H. Balakrishnan, M. Goraczko, N. Priyantha, Tracking moving devices with the cricket location system, in: Proc. 2nd USENIX/ACM MOBISYS Conf., Boston, MA, June 2004.
[67] SOS Operating System Project website, URL: http://nesl.ee.ucla.edu/projects/sos.
[68] F. Sparacino, Scenographies of the past and museums of the future: From the wunderkammer to body-driven interactive narrative spaces, in: Proceedings of ACM Multimedia 2004, 10–16 October 2004, New York, NY, 2004.
[69] F. Sparacino, The Museum Wearable: real-time sensor-driven understanding of visitors' interests for ersonalized visually-augmented museum experiences, in: Proceedings of Museums and the Web, MW2002, 17–20 April 2002, Boston, MA, 2002.
[70] F. Sparacino, C. Wren, G. Davenport, A. Pentland, Augmented performance in dance and theater, in: International Dance and Technology 1999, Tempe, Arizona, February 1999.
[71] V. Stanford, Wearable computing goes live in industry, IEEE Pervasive Computing (October–December) (2002) 14–19.
[72] N.M. Su, H. Park, E. Bostrom, J. Burke, M.B. Srivastava, D. Estrin, Augmenting film and video footage with sensor data, in: Second IEEE International Conference on Pervasive Computing and Communications, 14-17 March 2004.
[73] T. Sweeney, UnrealScript Language Reference. URL: http://unreal.epicgames.com/UnrealScript.htm.
[74] R. Szewczyk, NEST Meeting Presentation. http://ww.cs.berkeley.edu/~binetude/NEST/UWB.ppt, August 27, 2004.
[75] P. Tolmie et al., Unremarkable computing, in: Proc. of the SIGCHI Conference on Human Factors in Computing Systems, Minneanapolis, Minnesota, 2002, pp. 399–406.
[76] R. Venturi, Complexity and Contradiction in Architecture, Museum of Modern Art, New York, 2002.
[77] Vicon website, URL: http://www.vicon.com.
[78] F. Wagmister, J. Burke, Networked multi-sensory experiences: Beyond browsers on the web and in the museum, in: Museums and the Web Conference, Boston, April 2002.
[79] G. Wang, G. Cao, T. La Porta, A bidding protocol for deploying mobile sensors, in: Proceedings of the 11th IEEE International Conference on Network Protocols, IEEE Computer Society, 2003.
[80] G. Wang, G. Cao, T.L. Porta, Movement-assisted sensor deployment, in: INFOCOM, 2004.
[81] S.E. Weil, Making Museums Matter, Smithonian Institution, Washington, DC, 2002.
[82] M. Weiser, Creating the invisible interface: (invited talk) (Marina del Rey, CA, November 2–3, 1994).
[83] R. West, J. Burke, C. Kerfeld, E. Mendelowitz, T. Holton, J.P. Lewis, E. Drucker, W. Yan, Both and Neither: in silico v.1.0, Ecce Homology, Leonardo, MIT Press.
[84] S. Wilson, Information arts: Intersections of art, science and technology, Cambridge, MA, 2002.
[85] Y. Zou, K. Chakrabarty, Uncertainty-aware sensor deployment algorithms for surveillance applications, in: Global Telecommunications Conference, GLOBECOM'03, 2003.