Modeling and Simulating Electronic Textile Applications

Thomas Martin, Mark Jones, Joshua Edmison, Tanwir Shiekh, Zahi Nakad Virginia Tech
Dept. of ECE Blacksburg, VA 24061 tlmartin, mtj, jedmison, tanwir@vt.edu

***Abstract*— This paper describes our experiences with a simulation environment for electronic textiles. This simulation environment, based upon Ptolemy, enables us to model a diverse range of areas related to the design of electronic textiles, including the physical environment they will be used in, the behavior of the sensors incorporated into the fabric, the on-fabric network, the power consumption of the system, and the execution of the application and system software. We present results for two different applications, a large-scale acoustic beamformer and a pair of shape-sensing pants. Categories and Subject Descriptors The ACM Computing Classification Scheme: http://www.acm.org/class/1998/**

***General Terms*-Measurement, Performance, Design, Reliability, Experimentation.**

***Index Terms*— Electronic textiles, sensor networks, wearable computing, simulation.**

I. INTRODUCTION

ELECTRONIC textiles (e-textiles) are fabrics with interconnections and electronics woven into them. The electronics consist of both processing and sensing elements, distributed throughout the fabric. Potential applications for e-textiles include wearable computing (context-awareness, medical monitoring, and military uniforms) and sensor networks. E-textiles are embedded systems with a unique form factor, and are an extreme form of distributed computing: physically spread over a relatively smaller space, but having a greater dependence on physical locality of computation, lower bandwidth for communication, less available energy, and requiring knowledge of the dynamic shape of the fabric.

The design space for e-textiles is large, with many choices for yarns, weaves, components, system software, interconnection networks, and sensor placement. While the area is still in its infancy, it is clear that prototyping alone cannot adequately explore the design space for most applications. Consequently, a simulation environment is necessary, one that allows a designer to adequately model a range of areas, including the physical environment that the e-textile will be operating in, the behavior of sensing and processing elements, and software execution. At the same time, simulation results cannot be trusted unless the simulation environment is based in reality., so we have built prototypes and compared their actual behavior to the behavior predicted by simulation. In this paper, we describe the current status of our simulation environment for e-textiles and present results generated by both the environment and associated prototypes for two applications, a large-scale acoustic beamforming fabric for locating vehicles and a pair of pants for determining user motions. The remainder of the paper is organized as follows: Section 2 presents background material on e-textiles. Section 3 describes the elements of the simulation environment. Results from the environment for both applications are presented in Section 4. Finally, Section 5 summarizes the paper, presents conclusions, and discusses future work.

II. BACKGROUND

The two main areas of background information are potential applications of e-textiles and the design issues involved in creating them.

A. E-textile Applications

The Wwearable Mmotherboard project and related work at Georgia Tech has led to the creation of a system for monitoring a user’s health, including heartbeat and respiration as well as the location of a bullet wound [1][10]. Applications include monitoring infants for Sudden Infant Death Syndrome as well as monitoring the status of soldiers on the battlefield. In these projects, wires were woven into the fabric for communication of data along with optical fibers to detect the location of bullet holes.

Discrete sensors were attached and computing analysis was performeddone outside of the garment. Further work at Georgia Tech has investigated the use of FPGAs as self- configuring, fault-tolerant switches. In industry, a range of products areis either on the market or under development. ElekSen has developed a dual-purpose fabric keyboard that serves a dual-purpose in that it can be folded into a carrying case for a PDA device. ElekSenThis company has also developed fabrics for health-care applications and vehicle interior interaction user interaction with vehicle interiors as well as health-care applications [1]. Durability tests show these fabrics areis as durablerobust as normal textiles and that the sensing capability does not degrade even over millions of user cycles. Infineon has produced a wearable MP3 player [8]. The primary contribution of Infineon's work is a method for packaging and attaching the digital and analog components in a washable, durable form factor with pins at a suitable pitch for fabric. We have constructed a number of e-textile prototypes, including large-scale (up to thirty feet long) acoustic-beamforming textiles that can determine the location of vehicles [13] and garments for context-awareness, shown in Figure 1. These woven prototypes are embedded with a communicating network of sensors and computing devices and can operaterun continuously for significant periods of time on a standard nine-volt battery. We have also investigated the inclusion of novel materials, including thin piezoelectric films, into e-textiles [1]. Our long-term vision for e-textiles includes other innovative sensing capabilities such ashaving novel fibers woven into the fabric, such as fibers that can sense chemicals, act as batteries, or change color dynamically [13].

B. E-textile Design Issues

In a previous paper, we described the design issues associated withinvolved in e-textiles [10]. A successful simulation environment for e-textiles must encompass the areas of those design issues, a task complicated by their diversity across multiple areas. , so wWe summarize them here. A major challenge for designing e-textiles is that these issues span a diverse range of areas, including:

1) Physical environment
2) Sensor behavior,
3) Human body and motion,
4) Motion/draping of clothing,
5) Manufacturability (weave & piecework),
6) Networking,
7) Power consumption, and
8) Software execution.

One of our primary goals for e-textiles is to remain within the confines of existing manufacturing techniques in the textile and garment industries. This will enable e-textiles to take full advantage of the efficient and cost effective mass production techniques employed by the textile industry. Since garments are typically constructed of pieces cut from a large rectangle of fabric and then sewn together, the fabric must be designed such that the pieces can be easily interconnected electronically, allowing sensing and processing elements on different pieces of the garment to communicate.

Applications of e-textiles will have to sense the physical environment, both for wearable computing applications and non- wearable computing applications. Non-wearable applications will usually betypically manifest as sensor networks, which by definition are monitoring the physical environment. A particularly direct appealing and appealing wear-wearable computing application for e-textiles is context-awareness summarized simply as k, knowing what the user is doing, where the user is, and what is going on around the user, so that the system can adjust its behavior to the user's current situation [1][6]. Consequently, Aany simulation tool for e-textiles must be able to model the physics of the environment, and the behavior of the sensors used to monitor thate environment. For wearable computing applications, an e-textiles simulation tool must also be able to capture aspects of the human body, e.g., body size and limb motion, in order to model the dynamic location of the sensors. Because most garments do not fit tightly on the human frame, the draping and motion of the cloth in response to the underlying movement of the wearer may also have to be modeled to fully account for the movement of the sensors. One of our primary goals for e-textiles is to remain within the confines of existing manufacturing techniques in the textile and garment industries. This will enable e-textiles to take full advantage of the efficient and cost effective mass production techniques employed by the textile industry. Since garments are typically constructed of pieces cut from a large rectangle of fabric and then sewn together, the fabric must be designed such

that the pieces can be easily interconnected electronically, allowing sensing and processing elements on different pieces of the garment to communicate. One of our goals for e-textiles is to stay as close as possible to existing manufacturing techniques in the textile and garment industries. This will enable to e-textiles to take advantage of the cost-effectiveness of those industries. Garments are made up of pieces cut from a large rectangle of fabric woven on a loom and then sewn together. The fabric must be designed such that the pieces can be interconnected, allowing sensing and processing elements on different pieces of the garment to communicate. An e-textile simulation environment must also account for the computing aspects of the application: nNetworking, power consumption, and software execution. Sensing and processing elements must be interconnected through an on-fabric network [11]. The network topology is constrained by the properties of the weaving process. Particularly, similar to the threads themselves,In particular, like the threads in the fabric, electronic interconnections must be run in two perpendicular directions. The network must also be fault tolerant, asbecause the fabric will be subject to manufacturing defects as well as wear and tear during use. Like other mobile and wearable computing devices, e-textiles will typically be battery-powered, and thus power consumption must be modeled in order to make the e-textile energy efficient. E-textile systems differ from other low power systems in that the power sources will be modular and distributed in order to maintain flexibility. Whereas other low power systems must optimize energy use from a single power source, fabric substrates will likely have to optimize energy use from many power sources. Consequently, the optimization problem is more difficult for fabrics. In order to achieve true system level fault tolerance in an e-textile, power fault tolerance in addition to network fault tolerance, must be implemented.To make the fabric tolerant to tears and other faults, p Paths from power sources to sensor and computation nodes must be dynamic, allowing power to be routed around damaged sections of the fabric. Due to Because of the physical locality of computation, e.g. a node will search for other nodes within a given physical region for beamforming, power consumption may be non-uniform across the fabric. As a result, the number and distribution of power sources on the fabric will have a great impact on lifetime and application performance, which we will elaborate upon later in the paper. As will be shown later in the paper, a single power source is not very fault tolerant; a single short circuit may cause the entire fabric to lose power. But having multiple power sources may mean that one part of the fabric may exhaust its source before other parts do, leading to a degradation of performance. Because of the expense and uncertainty of prototypes, it is desirable and necessary to test and evaluate the functionality and performance of the software without having hardware on hand. To achieve this, weFinally, it is also necessary to simulate the behavior of the software in order to test functionality and estimate performance without having a hardware prototype available. We have built modules for our simulation environment that allows the same software to be executed in simulation and on the target hardware platform, with only minor changes. It should be stressed that while we have the option of targeting and porting to specific hardware that we emulate not a specific device, but a general interrupt based computational model.

III. SIMULATION ENVIRONMENT

To address these simulation needs, we have selected Ptolemy II as a means of integrating a wide range of simulation capabilities. Ptolemy II provides the type of diverse environment required to handle very different simulation domains and an open architecture that can accommodate interfacing to other environments [12]. For example, Ptolemy II has provisions for simulations in the continuous domain, which is useful for simulating the physics of the environment, while simultaneously providing a discrete event domain that is appropriate for computation. At present, our simulation environment has encompassed each of the areas described in section 2 except for the motion/draping of cloth and manufacturability and manufacturability. Our approach has been evolutionary, adding one or two areas to the environment at a time. For a given application, however, it may not be necessary to model all of the areas. In some cases, the area may not apply, e.g., it is not necessary to model human motion for non-wearable applications. Our approach has been evolutionary, adding one or two areas to the environment at a time. For example, in the acoustic beamforming application, we first modeled the physics of sound traveling from a moving vehicle to the cloth and the execution of the beamforming code on the processing elements. We then added models of the power consumption and the networking scheme to the simulation. We currently have modeled aspects of each of the areas listed in section 2 except for motion/draping of cloth and manufacturability. For a given application, however, it may not be necessary to model all of the areas. In some cases, the area may not apply, e.g., it is not necessary to model human motion for non-wearable applications. Figure 2 presents the general framework of our simulation environment. As was mentioned previously, Ptolemy has several different models of computation, each with its own method of handling time and concurrency. This permits systems to be represented using both continuous time and discrete event domains, among others. The ”physical environment” and ”sensor” blocks in Figure 2 were modeled in the continuous domain, while the other blocks were modeled in the discrete event domain. ¡Verify that.¿ Furthermore, Ptolemy is extensible, allowing it to be connected externally to existing simulators and others tools so that that functionality does not have to be created within Ptolemy itself. This is designated in Figure 2 by the portions of the simulator that are outside of the dashed box. For example, for modeling human motion we used three-dimensional motion data captured using special video equipment, such as would be used for creating computer- animated movies [2]. The motion data was pre-processed using MATLAB to calculate the acceleration of points on the body; this acceleration data and was then used as input to the sensor models within Ptolemy, as will be described in the next section. To address these simulation needs, we have selected Ptolemy II as a means of integrating a wide range of simulation capabilities. Ptolemy II provides the type of diverse environment required to handle very different simulation domains and an open architecture that can accommodate interfacing to other environments [12]. For example, Ptolemy II has provisions for simulation in the continuous domain, which is useful for simulating the physics of the environment, while simultaneously providing a discrete event domain that is appropriate for computation. At present, our simulation environment has encompassed each of the areas described in section 2 except for the motion/draping of cloth and manufacturability. Figure 2 presents the general framework of our simulation environment. Modules within the dashed line are implemented within the Ptolemy II environment, while those on the outside represent extant systems. For example, our models of the physical environment are implemented in the Ptolemy continuous simulation domain while our data from human motion is read from processed data files created by a video motion capture system [3]. Some aspects of the system are application dependent, some are fabric configuration dependent, and others are constant across application. The models for the sensors and the associated model of the physical environment are application dependent. For example, the microphone models and the propagation of vehicle acoustics are specific to the beamforming application and are described in the next section. The simulation environment has a representation of the number and location of sensors, processors, communication wires, and power lines. The specific configuration is clearly dependent on the characteristics of the e-textile being modeled, but this configuration is not dependent on the application. To allow for exploration of a range of architectures and sensor configurations, we have created an automatic generation tool that allows for the Ptolemy model of the textile to be generated from a configuration file. The physical model of the fabric considers the placement of the components on the fabric and the interconnections between pieces of fabric. It can be made to automatically generate locations of power and ground lines through each piece based upon an input wiring density per unit length. The model then automatically connects each component on the fabric to the nearest power/ground line pair. The power/battery model assumes that the battery has a fixed amount of energy and kept track of how much energy had been drawn from it and how much remained. To model the energy consumption, the processor emulation modules must provide a table of power management states and the amount of power consumed in each state to the power/battery model, which we fashioned after the ACPI power management interface [2]. The power/battery model can then track how long each component is in a given state, and uses the timing information and the power for that state to calculate how much energy had been consumed. A more elaborate battery model that accounts for changes in battery capacity with load power is possible [13], but this first- order model is sufficient for our purposes. Finally, there are aspects of the system that are constant across application and textile configuration. Most important of these is the processor emulation module, with one module representing each processor in the e-textile. Each module accepts input from sensors as well as raw network data from other processor modules. The processor module implements an interrupt-driven model of computation, with several types of interrupts managed, including sensor data arrival, network data arrival, and clock data. The processor module allows for the user to insert their own C code to handle these interrupts as well as initialize the system. This C code has been ported intact to actual processors, allowing for emulation of e-textile prototypes in a simulation mode. The processor model is capable of outputting results to a network port (both to a debug port and to other processor modules) and providing results on power consumption and processing time. It is important to note that this is not a cycle- by-cycle processor simulator; the data on power consumption and processing time is based on user estimates for a given processor and interrupt handler to be modeled. Other aspects of the system that are constant across application include the power/battery monitoring system and the fault-generation module. Because e-textiles are at an early stage of development and a range of applications are being investigated, it is not possible to confine all exploration of the design space to a pure simulation mode. Further, there is a need for extensive debugging in this novel hardware/software environment, but there is a complete dearth of such tools. To partially address these needs, we have augmented the environment with the capability to accept live sensor data as well as a hybrid mode of operation. In the live sensor data mode, we can replace select sensors models with sensor data provided over a TCP/IP link. In cases where the data processing cannot keep up with the sensor input rate, prerecorded data is fed from a file. The hybrid mode of operation, inspired by [cite Mani S.], allows for the simultaneous integration of the simulation environment with physical e-textile prototypes. For example, we have constructed a physical acoustic beamforming e-textile with four processing elements [7], but would like to investigate the performance of this prototype in an architecture with tens of processors. The hybrid mode bridges the simulated and physical e-textile networks over a TCP/IP link, allowing for extensive investigation and emulation of network and application behavior [cite Zahi thesis]. Finally, this same bridging technique can be used to decompose simulations of large e-textiles across multiple processors to reduce simulation time.

IV. RESULTS

In this section we present results from two different applications, a large-scale acoustic beamforming fabric for locating vehicles and a pair of pants for determining user motions.

A. Large-scale acoustic beamformer

The purpose of the first application, the large-scale acoustic beamformer, is to locate passing vehicles by triangulating on their sound. The beamformer, pictured in Figure 3, consists of four clusters. Each cluster has seven microphones and a DSP processing module. Using the difference in the time of arrival of a sound to each microphone, a cluster can calculate an angle to a sound source. The clusters then communicate their angles to each other, and triangulate to find the location of the sound source. A picture of one cluster of the prototype was shown in Figure 1. We have previously described the use of the simulation environment to analyze the impact of communication schemes and sampling rates on the power consumption and accuracy of the beamformer, showing that a trade-off exists between accuracy and power consumption based upon the sampling rate and the set of microphones chosen within each cluster [7]. Another interesting trade-off arises when fault tolerance is considered. Because of manufacturing defects and normal wear and tear on the fabric, it is likely that there will be broken and shorted wires in the fabric over the course of its lifetime. If the system is powered by a single battery, then a short circuit in one portion of the fabric may cause the entire fabric to lose power. One solution to this problem is to have multiple batteries, with each battery powering an isolated portion of the fabric. However, this is less than optimal if no faults occur, because if a portion of the fabric is active more often, its battery may be exhausted while other, less active portions of the fabric still have energy remaining. The system may still be able to operate using only these portions, but its accuracy may be reduced. We modeled this problem using our simulation environment by creating a physical fault model that introduces short- and open- circuits into the fabric. Because e-textiles are not yet widely available, there is no information on the types and frequency of the faults that occur. As a first step, we assumed that faults could either be short- or open-circuits, and given the intended location for the deployment of the beamformer (a roadside in a hostile military environment), we generated a random set of faults of various types, shapes, and densities. For example, one type of shape was a rectangular strip at some angle across the fabric such as might occur if a vehicle were to drive over the fabric. Another type was fashioned after a shotgun blast, a collection of random point faults over a constrained area. Brief description of the application, construction of the e-textile Modeling the battery, power consumption, faults. To model the power consumption, each major state of the beamforming algorithm was identified. The average power consumption was found to be constant while in each state, although the power varied considerably from state to state. Trade-offs between fault-tolerance and performance with no faults. Possible configurations of the batteries. We investigated three different configurations of the batteries for the beamformer. Configuration 1 had one battery for the entire beamformer, configuration 2 had one battery for each pair of clusters (two batteries total), and configuration 3 had one battery for each cluster (four batteries total). The overall energy capacity was equal for each of the three configurations. Figure 4 shows the acoustic beamformer in configuration 3. Each cluster has its own battery, as indicated by the small square near the left of each cluster Two faults have been introduced, a vehicle track fault on the leftmost cluster and a shotgun fault on the cluster next to it. The faults are all short circuits, and the placement model has determined that the components on those two clusters have been disabled by the short circuits, as indicated by the X's. Components on the right two clusters are still functional, as indicated by the diamonds.

B. *Shape-sensing pants*

Our second example of the using the simulation environment in action involves the design of a pair of shape-sensing pants. The intended function of the spants is e pants are intended to be used to determine the activity of the user, e.g., running, walking, standing. A number of other groups have investigated using sets of sensors to determine use activity [6][16][18]. All of these groups built hardware prototypes of their sensor systems and then tested their functionality on a few individuals, usually other people working in their laboratory., Our however the simulation environment haprovides two advantages to this method. First, by using motion data collected from a wider set of individuals, we have a greater confidence that our system will function for a broad segment of the population. Second, we are able to study issues of sensor choice and sensor placement without having to first build a hardware prototype. Based upon our explorations in simulation, the unique opportunity to analyze sensor output without a prototype. Tthe first iteration garment incorporates two types of sensors, accelerometers and thin piezoelectric films. The development of the models and their verification are described in the following paragraphs. The accelerometer is modeled by calculating the discrete derivative of the position data gleaned from the motion capture source. The resulting accelerations in mm/s^2 are converted to g's, a multiple or fraction of Earth's gravitational constant $9.81m/s^2$. The specifications for the ADXL311 from Analog Devices, specifically the number of millivolts per g, were used to calculate the final accelerometer output voltage. This, however, did not completely model the behavior of the sensor. Since the sensor and the body are closely coupled, the accelerometer may undergo a change in orientation, particularly if the sensor is used on a part of the body that exhibits a high degree of freedom such as the wrist or ankle. To account for the change in orientation, a rotation matrix is used similar to that show in Figure X and governed by the equations Xcorrected=Xuncorrected*cos(?) + Yuncorrected*sin(?) and Ycorrected=Yuncorrected*cos(?) + Xuncorrected*sin(?). The accelerometer model includes optional inputs for position data, such as the knee and heel for the ankle, which are used to calculate the angle necessary for computation of the rotation matrix. The resulting values Xcorrected and Ycorrected should and do correspond to actual sensor output as shown in Figure Y. In practice, piezoelectric films are utilized to sense both force and joint angles, however, the simulation model for the piezoelectric film, due to lack of force data, can only encompass joint sensing. The ideal (mathematical) model of a piezoelectric film is an ideal voltage source in series with a capacitor (typically tens of nano farads) across an input impedance forming an high-pass RC circuit with a cutoff frequency equal to *(pi*RC). Even for large values of R, the resulting cutoff frequency is typically hundreds of hertz. Above this cutoff frequency, the film produces voltage proportional to the physical stimulus; below this cutoff frequency, the film produces a voltage proportional to the rate of change of the input. Since the frequency of most human motions (typically 1 Hz or less) is well below this frequency, particularly joint angles, we can assume that the piezoelectric film will measure the angular rate of change if placed on the joints. The simulation model consists of inputs for three points that form a joint angle, such as the hip, knee, and heel, which form the knee joint. The angle formed by these three points as seen in Figure A is calculated and the discrete derivative of the resulting angles is the final piezoelectric sensor output. This was verified using the analytical results from a pendulum in [10][ISWC paper]. Further verification, using the piezoelectric film to measure the derivative of the knee joint angle is shown in Figure Z.

V. CONCLUSIONS

VI. ACKNOWLEDGMENTS

This material is based upon work partially supported by the National Science Foundation under grant no. CCR-0219809 and by DARPA under contract F30602-00-2-0548. The authors gratefully acknowledge Dana Reynolds and Elouise Coupey for their efforts in the weaving of the fabrics and construction of the pants.

REFERENCES

[1] Promatech web site, *http://www.promatech.it*.