Challenges and Opportunities in Electronic Textiles Modeling and Optimization*

Diana Marculescu, Radu Marculescu, and Pradeep K. Khosla
Department of Electrical and Computer Engineering
Carnegie Mellon University
{dianam,radum,pkk}@ece.cmu.edu

ABSTRACT - This paper addresses an emerging new field of research that combines the strengths and capabilities of electronics and textiles in one: electronic textiles, or e-textiles. E-textiles, also called Smart Fabrics, have not only "wearable" capabilities like any other garment, but also local monitoring and computation, as well as wireless communication capabilities. Sensors and simple computational elements are embedded in e-textiles, as well as built into yarns, with the goal of gathering sensitive information, monitoring vital statistics and sending them remotely (possibly over a wireless channel) for further processing. Possible applications include medical (infant or patient) monitoring, personal information processing systems, or remote monitoring of deployed personnel in military or space applications. We illustrate the challenges imposed by the dual textile/electronics technology on their modeling and optimization methodology.

Categories and Subject Descriptors: I.6 [Simulation and Modeling]: Modeling methodologies; B.8.2 [Performance and reliability]: performance analysis and design aids.

General terms: design, performance

1. INTRODUCTION

Electronic textiles or e-textiles are a new emerging interdisciplinary field of research which brings together specialists in information technology, microsystems, materials, and textiles. The focus of this new area is on developing the enabling technologies and fabrication techniques for the economical manufacture of *large-area*, *flexible*, *conformable* information systems that are expected to have unique applications for both consumer electronics and military industry.

E-textiles offer new challenges for designers and CAD tool developers due to their unique requirements, cutting across from the system to the device and technology:

- The need for a new *model of computation* intended to support widely distributed applications, with highly *unreliable* behavior, but with stringent constraints on *longevity* of the system.
- *Reconfigurability and adaptability* with low computational overhead. E-textiles rely on simple computing elements embedded into fabric or directly into active yarns. As operating conditions change (environmental, battery lifetime, etc.), the system has to adapt and reconfigure on-the-fly to achieve a better functionality.
- *Device and technology* challenges imposed by embedding simple computational elements into fabric, by building yarns with computational capabilities, or by the need of unconventional power sources and their manufacturing in filament or yarn form.

There have been a handful of attempts to design and build prototype computational textiles. In one of the earliest accounts [1], the authors demonstrate attaching off-the-shelf electrical components such as microcontrollers, surface mount LEDs, piezoelectric transducers etc., to traditional clothing material, transforming the cloth into a breadboard of sorts. In fabrics which contain conductive strands, these may be used to provide power to the devices, as well as to facilitate communication between devices. In [2], the authors extend the work presented in [1], detailing methods by which items such as user interfaces (keypads) and even chip packages, may be constructed directly by a textile process. The routing of electrical power and communications, through a wearable fabric, was addressed in [3] where the authors provide a detailed account of physical and electrical components for routing electricity through suspenders made of a fabric with embedded conductive strands. A complete apparel with embedded computing elements is described in [4]. The authors describe a jacket designed to be worn in the arctic environment, which augments the capabilities of the wearer with a global positioning system, sensors (accelerometers, conductivity electrode, heart rate monitors, digital thermometers) and heating. However, all these approaches are single design points and do not provide a methodology for the evaluation and validation of e-textile systems.

It is our belief that in support of e-textiles a new design paradigm is needed, as well as a more general model of computation which supports *local computation* and *inexpensive communication* among computational elements. In the classic design cycle (Fig.1(a)), the application is mapped onto a given platform architecture, under specified constraints (performance, area, power consumption). When these constraints are met, the prototype is tested, manufactured and used for running the application. In the case of e-textiles (Fig.1(b)), wide area textile networks are built onto e-textiles, with no prescribed functionality. To achieve high yields (that is, low defect rate), as well as high fault-tolerance later in the lifetime cycle, *regularity* is important. The application is modified so as to expose as much local computation as possible, via partitioning or refinement. At power-up, the application is mapped so as to optimize different metrics of interest (such as, quality of results, power consumption, operational longevity, fault-tolerance) and later re-mapped whenever operating conditions change.

Although e-textiles ostensibly present distributive computing challenges similar to those currently being pursued in adaptive control network and pervasive computing, the specific characteristics and demands of an e-textiles info-computing environment add new dimensions to those challenges. In fact, e-textiles impose specific challenges as opposed to other applications in the general area of networked systems:

*This research has been supported in part by DARPA under contract no. DABT63-96-C-0083 and SRC under grant 2001-HJ-898.

DAC 2002, June 10-14, 2002, New Orleans, LA, USA.

What e-textiles are *not*

- Classical data networks. While the underlying structure of an e-textile application implies the existence of many processing elements, connected in a Textile-Area Network (TAN), they have limited processing and storage capabilities, as well as very limited power consumption budgets. Hence, classic techniques and inherent methodologies for coping with mapping an application, communication among nodes and dealing with network failures (including congestion), are not appropriate. In addition, having very limited processing capabilities, e-textiles are not the equivalent of "desktops/laptops on a fabric," restricting significantly the range of applications that can be mapped on them.
- Networked embedded systems (including wireless sensor networks). Wireless sensor networks share with e-textile based applications the limited power budgets, and somewhat the limited processing and storage capabilities. However, communication among processing elements is wired, and thus, much less expensive than the wireless communication. In addition, as opposed to ad-hoc networks, the topological location of different processing elements is fixed throughout the lifetime of the application (although the mapping of the application on processing elements may change). Lastly, e-textiles must have low manufacturing costs, and thus, the defect rate of the processing nodes (and physical links) will be much higher and very different than in the case of wireless networks. More limited processing and storage capabilities, in conjunction with higher failure rates imply that existing body of research for sensor networks is not applicable directly to TANs.

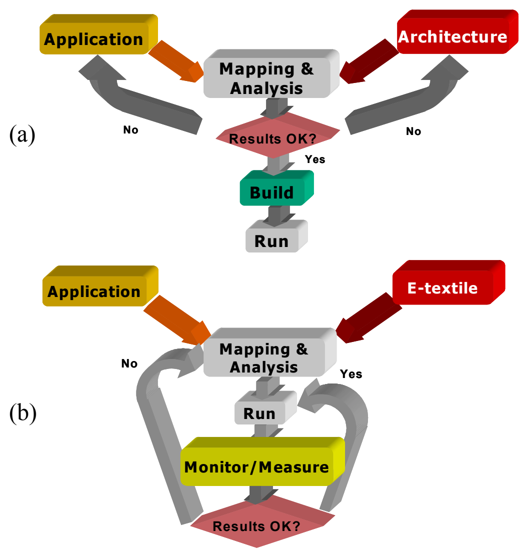

Fig.1 Classic design (a) vs. e-textile lifetime cycle (b)

What e-textiles are

E-textiles are "living designs," consisting of highly unreliable low-cost simple components and interconnect. These characteristics are in fact shared by other non-silicon computing systems (such as those based on nanoelectronics). In many ways, e-textiles systems will be the conformable, fabric equivalent of rigid body, embedded MEMS. However, unlike MEMS, e-textiles require coping with *tears* in the fabric, whether it is due to regular wear of the fabric, or due to unexpected damage. The salient features of e-textiles are:

- limited processing, storage and energy per computational node
- potential failures for *both* nodes and communication links
- highly *spatially and temporally correlated* node and/or link failures due to topological placement or due to external events
- need for scalability and flexibility of resource management, as well as local vs. global management trade-off
- active or smart links which embed simple logic (passive or active components)

In this paper, we present our vision on developing a methodology for *modeling*, *analysis* and *middleware support* for e-textiles that includes:

1. The *Model of Colloidal Computing* (MC^2). Operational longevity and effective usage of e-textiles has to be supported by an entirely new computational and architectural model. We call such a model colloidal computing (MC^2)[5]: simple computation particles are dispersed in a communication medium which is inexpensive, (perhaps) unreliable, yet sufficiently fast. This type of clustering into computation intensive kernels with loose inter-particle communication, but tight intra-particle communication is typical not only for the underlying hardware, but also for the actual software application that runs on it. In the MC^2 model the application is viewed as a set of "nano-particles" loosely communicating so as to exchange information. Likewise, the underlying architecture has local computation, as well as remote communication capabilities, when needed.
2. An *analytical and simulation-based* framework that allows to explore the design space early on for energy/performance/fault-tolerance trade-offs leading to designs that are feasible and easily adaptable to the application profile. Such a framework is able to provide optimal power/performance operating points, increase operational lifetime under prescribed fault-tolerance limits, and ensure application-driven adaptability during the entire e-textile lifetime cycle.
3. A *Port-Based Adaptable Component Architecture* (PBACA) that allows every computational node to reconfigure its computation and communication with other nodes, in real-time. The motivation stems from the ultra-redundant and distributed component networks that comprise an e-textiles computing environment with the ability to sense, act, store, emit. Such capabilities require a lightweight software infrastructure that enables resource management, communication, global and local tasking, and fault-tolerance between components and across the entire system.

The rest of the paper is structured as follows: Section 2 presents the colloidal model MC^2. Section 3 addresses the problem of application modeling, partitioning and mapping. Section 4 presents the architecture and communication modeling, while fault-tolerance management is addressed in Section 5. Finally, some experimental results are presented in Section 6.

2. COLLOIDAL COMPUTING

The model of colloidal computing supports local computation and inexpensive communication among computational elements: simple computation particles are dispersed in a communication medium which is inexpensive, (perhaps) unreliable, yet sufficiently fast (Fig.2). This concept has been borrowed from physical chemistry [6][1] since some of its properties and characteristics resemble the features that an e-textile based computational model would need. Specifically:

- In case of unstable colloidal suspensions, colloidal particles tend to *coalesce* or *aggregate* together due to the Van der Waals and electrostatic forces among them. *Coalescing* reduces surface area, whereas *aggregation* keeps all particles together, without merging.

1.*Colloid* [käl´oid] = a substance consisting of very tiny particles (1 nm and 1000 nm) suspended in a continuous medium, such as a liquid, a solid, or a gaseous substance.

Similarly, the resources of a classic system are *coalesced* together in a compact form, as opposed to the case of e-textile-based computation where useful work can be spread among many, small, (perhaps) unreliable computational elements that are dynamically *aggregated* depending on the needs (Fig.3). *Dynamic* or *adaptive aggregation* is explicitly done whenever operating conditions change (e.g., failure rate is too high or battery level is too low).

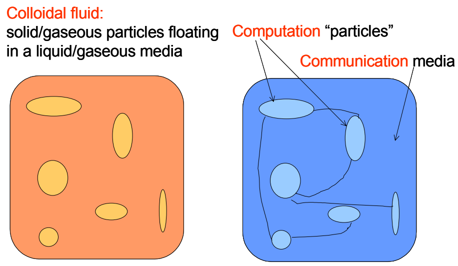

Fig.2 Colloidal computing model

- The MC^2 model supports a wide range of computational granularities, from *computation embedded in the yarn*, up to simple sensors, clusters of sensors, or even specialized computing elements for *local* and *global* resource management, in a tiered, *hierarchical organization*. Perhaps the most striking difference between classical, performance-oriented computing systems and smart fabric-based systems is the possibility of exporting some of the computational power onto the links or active yarns. Since transistor densities of 10,000/cm^2 are possible in the next 3-4 years [7], some functionality can be deployed onto this extra logic in the form of A/D converters or simple computing elements (i.e. adders, multipliers, pseudorandom generators).

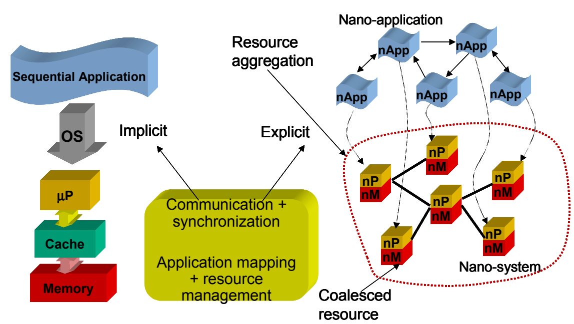

Fig.3 Coalesced vs. aggregated resources

2.1 Explicit vs. Implicit Communication and Management

We propose to apply the MC^2 model to both the application software *and* architecture platform. Specifically, most of applications under consideration consist of a number of computational kernels with high spatial locality, but low degree of communication among them. Such kernels (typically associated with media or signal processing applications) can thus be mapped on separate computational "particles" that communicate infrequently for exchanging results.

The underlying architecture of the e-textile is also complying with the proposed MC^2 model. Both the architecture configuration and the mapping of the application on the computational elements is done dynamically, at power-up and whenever the system becomes "unstable". As an analogy, in case of unstable colloids, a minimal energy configuration is achieved via *coalescing* (e.g., oil in water) or *aggregation* (e.g., polymer colloids).

In e-textiles, a "stable" configuration is one that achieves the required functionality, within prescribed performance, power and probability of failure limits. We propose to employ *aggregation* (Fig.3) or *dynamic connection* of computational "particles" based on their state (i.e. functional or not; idle or active) and their probability of failure so as to achieve a required quality of service (QoS) (characterized by performance, power and fault-tolerance). The mapping and reconfiguration process of the application onto the underlying architecture is achieved via *explicit* mechanisms, as opposed to classic computing systems where mapping and resource management is done via *implicit* mechanisms.

Re-organization and re-mapping need *thin middleware clients*, sufficiently simple to achieve the required goals without prohibitive overhead. In addition, fault and battery modeling and management become intrinsic components for achieving certain levels of quality of results or *operational longevity.*

We describe in the sequel some of the issues that are critical to the e-textile application lifetime, namely application modeling and partitioning, architecture and communication modeling, as well as fault modeling and management.

3. APPLICATION MODELING AND PARTITIONING

The fundamental problem which should be addressed in order to efficiently map complex applications onto e-textiles is the issue of *concurrency.* Given that the e-textiles contain many computational particles distributed on large surfaces, it is of crucial importance to expose the entire concurrency available at application (behavioral) level since this will dictate the ultimate limit of parallelism that may be obtained when using real hardware and software on e-textiles. To this end, we propose *trace-driven application modeling* for e-textiles. This is motivated by our observation that, constraining a given application with various input traces leads to very different *clusterings* of the probability distribution that characterize the application itself. Tuning the target e-textile architecture to this large spectrum of different probability distributions is the most important development in obtaining efficient mappings with respect to certain performance metrics.

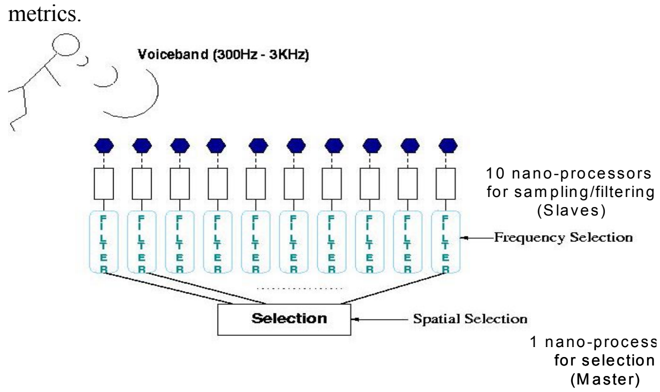

Fig.4 Acoustic beamforming application

A practical example from acoustic beamforming illustrates our point (Fig.4). The acoustic beamforming application uses a processor (master) in conjunction with an array of sensors (slaves) to provide a versatile form of spatial filtering [9]. The entire process graph corresponding to the acoustic beamforming application can be easily modeled using a *Producer-Consumer* paradigm as in [8]. The *Producer* and the *Consumer* are the *computational particles*, while the buffer between them is the *communication medium* in the MC^2 model. For different configurations, we get in a fully analytical manner all the metrics of interest (e.g., throughput, utilization, average response time, mean time to failure or *performability*[1]). Although less accurate than simulation, the analytical approach avoids lengthy off-line profiling simulations for evaluating performance. While for 10 processes we have to deal with a space of 236K states, for a scenario involving 20 processes we may have more than 14G states. This will dramatically increase the time complexity needed in any simulation-based environment for performance evaluation.

4. ARCHITECTURE, COMMUNICATION AND FAULT MODELING

The *architecture* modeling step consists of developing abstract specifications of the e-textile platform and then producing analytic models that reflect the behavior of that particular specification [10]. The model of the *communication architecture* relies on information gathered during and an initial *topology discovery* step which reveals the set of computational and communication resources that are available. Then, based on the performability threshold set for the application at hand, application's computation particles are mapped to available hardware so as to achieve a certain functionality. As operating conditions change (e.g., permanent failures due to fabric tear-off, or intermittent failures due to battery depletion), the application has to be remapped and/or communication links re-routed.

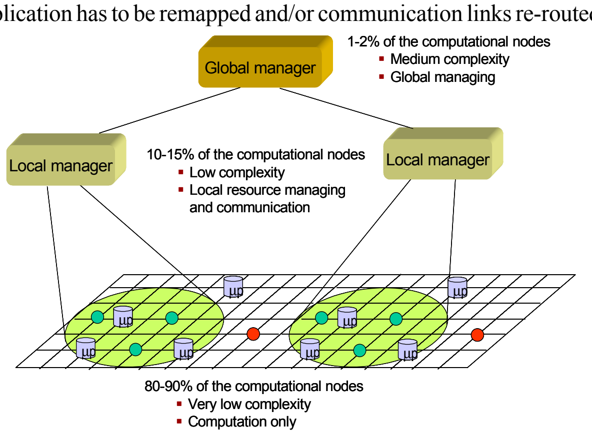

Fig.5 Tiered e-textile architecture

In addition, to better tackle scalability, we propose a tiered architecture (Fig.5) with *local management* of resources (including power consumption, performability, fault tolerance or QoS) and *global coordination* of local managers. Such an architecture offers the advantage of better scalability, as well as fault-tolerance. Specifically, the vast majority of the nodes are very simple computational elements, possibly embedded onto active links. These *nano-nodes* are responsible mostly for computation, and take very simple local decisions (e.g., randomly route data to the neighbors). The next level is comprised of several *micro-nodes* doing local resource management on *clusters* of nano-nodes that cooperate closely to achieve a certain goal (e.g., beamforming). Micro-nodes have more computational capabilities and may run a lightweight operating system kernel that deals with process migration among nano-nodes and overall cluster-level resource management (including energy, quality, fault-tolerance). At the highest level, very few *macro-nodes* are available to realize global resource management and overall coordination of local managers for the achievement of a certain goal (such as, a certain quality under given fault-tolerance constraints, or a certain operational longevity under prescribed battery lifetime limits).

1.Performability is defined as the probability of maintaining a certain level of performance on a finite time horizon.

We also assume that the power distribution system follows the same *clustered* organization, with nodes sharing a power source. Power and battery management is also done locally, and only coordinated globally, whenever an entire region has to be remapped for quality reasons.

Of particular interest is the modeling of the active yarns (*links*) that are used to ensure local and global communication. As opposed to classical distributed systems, for e-textiles the links between nodes may incorporate tens to a few hundreds of transistors and then can perform simple functions necessary to route data to the destination nodes. This is extremely useful if we want to move some of the computational complexity from the nodes onto the yarns and ensure a smoother migration of tasks from one region to another on the e-textile fabric.

4.1 Fault Modeling and Analysis

While the problem of reliability or fault-tolerance of systems with unreliable components has been addressed before, the vast majority of approaches for large networked systems consider only the failure of links or consider that failure probabilities are independent. This is not necessarily true for the case of e-textiles where physical locations of nodes and links are close one to another. In fact, one important characteristic that sets e-textiles apart from classic networking applications is the possibility of *joint* failures for nodes and links. Moreover, the probabilities of failure for nodes and links are *highly correlated*. This comes in sharp contrast with real data networks that assume either node or link failures, but not both.

In addition, the way faults affect the overall behavior is very different, depending on the level in the hierarchy. Graceful quality degradation in a fault-tolerant environment is achieved at *micro-level* by using spare or redundant nodes available in a region or cluster of simple *nano-nodes* that cooperate to achieve a certain goal. Extra redundant nodes are used to run code "migrated" from defective nano-nodes, or nodes running low on battery. In contrast, at macro-level, entire regions can be migrated on demand by the global manager, whenever there is no more redundancy to be exploited at lower levels for achieving a certain level of quality.

5. FAULT-TOLERANCE MANAGEMENT

The model of the communication architecture relies on information gathered during an initial topology discovery step which reveals the set of computational and communication resources that are available. Then, based on the performability threshold set for the application at hand, application's computation particles are mapped to real hardware so as to maximize the overall fault-tolerance. As operating conditions change (e.g., permanent failures due to fabric tear-off, or intermittent failures due to battery depletion), the entire application has to be remapped and/or communication links re-routed (Fig.6). What is unique for e-textiles, is the concept of *cost function of a region* which represents a weighted function of quality of results (QoR), power and fault-tolerance constraints that we use to guide the application mapping and remapping process onto a TAN. Such reconfiguration mechanisms assume that redundancy exists for both nodes and links, such that whenever quality decreases under a certain threshold the virtual communication channel is rerouted or remapped on a different physical channel (Fig.6(a)), or the entire application is remapped onto another set of processing nodes (Fig.6(b)).

Many possibilities exist for implementing redundancy. In a fixed infrastructure, the logical implementation of redundancy is to duplicate resources, with one resource taking over on the failure of the other. This is usually achieved via code migration from a non-functional node to a spare, fully functional one. Code migration is generally a difficult problem, as it could, in the worst case, require the movement of the entire state of an executing application. However, for e-textiles, migration of running applications can be made feasible by restricting the amount of application state that must be preserved. A particular advantage in e-textiles is that, since the environment in which they operate is inherently error-prone, applications can already be constructed to minimize state and facilitate rapid restartability.

In general, process migration [11] makes the fundamental assumption that each node will provide an identical execution environment. At the minimum, the processor architectures of the nodes should be identical, or a hardware abstraction layer (such as a virtual machine) must provide a uniform platform for execution.

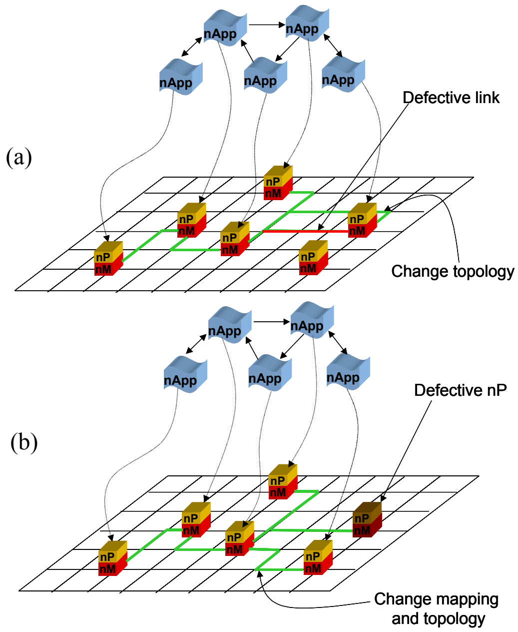

Fig.6 Dynamic reconfiguration in the presence of failures

While migrating among nano-nodes will be very different from migration at micro/macro-node or cluster/region level, a virtual machine may provide a uniform abstraction across varied hardware platforms.

5.1 Middleware Architecture

The ultra redundant and distributed component networks that comprise an e-textiles info-computing environment requires a middleware infrastructure that enables resource management, communication, global and local tasking, and fault tolerance between components and across the entire system. This fabric middleware software should foster locally optimized computation that result in near globally optimal behavior while simultaneously routing around component and communication failures resulting from fabric wear and tear. To this end, we believe that the Port-Based Adaptable Component Architecture (PBACA) approach is well suited to meet these demands [12].

In short, the PBACA middleware suite facilitates the development and deployment of self-adaptive, distributed, e-textile applications. Unlike sequential programming models that require an application to be a single stream of instructions, PBACA can utilize the MC^2 model allowing simultaneous streams of instructions. To exploit the power of PBACA, a solution to a problem must be decomposed into a hierarchy of interconnected tasks. We consider a task to be some flow of execution that takes zero or more inputs, produces zero or more outputs, and may modify some internal state. These input and output points are called *ports*, and the fundamental unit of execution is a Port-Based Module (PBM). For e-textiles platforms, a single PBM is the most fine grain unit of code able to execute on nano-nodes which are comprised of dedicated computational nodes or lighter weight elements embedded in active links. Whereas a PBM represents the most basic unit of execution, a Port Based Process (PBP) represents the most basic unit of self-reconfiguration and self-adaptability. In other words, a PBP is the smallest unit of an application that can measure its performance, quality of service, or operational longevity and take steps to improve those metrics. PBPs are both recursive and hierarchical in that they are composed of sets of interconnected PBMs or other PBPs. In typical e-textiles applications, a PBP will execute on a micro-node's cluster of nano-nodes to achieve logical or spatial sub-goals while satisfying local constraints. Similarly, the cooperating PBPs that form a macro-PBP work together to achieve the overall goal and satisfy the global requirements (such as fault tolerance constraints, or an operational longevity under prescribed battery lifetime limits.)

The overall PBACA middleware approach will foster plug and play integration within an e-textiles system, between e-textiles systems, and between e-textiles systems and existing/future computing systems. Furthermore, once integration is established, higher-level processing may be off-loaded to more powerful computational elements to fit the other information spaces requirements.

6. E-TEXTILE VALIDATION

As a newly emerging field, e-textiles impose new challenges not only on modeling and analysis, but also on simulation tools used for validation. Specifically:

- The simulation environment must be built around a multi-processor architectural instruction set simulator able to provide detailed power/performance numbers for the processing elements.
- Of particular interest is the process of modeling active yarns (*links*) that are used on e-textiles to ensure local and global communication. As opposed to classical distributed systems, in case of e-textiles the links between nodes can perform simple functions to help the data routing process. This is extremely useful in ensuring a smoother migration of tasks from one region to another on the e-textile fabric.
- In addition, the simulation testbed must include detailed battery models for predictive *power* and *fault-tolerance management*, sufficiently accurate network simulation, as well as *failure modeling* and support for *code migration* at nano- and macro-level (region).
- In addition to modeling the nonlinear characteristics of battery storage depletion, the *local resource manager* must store and manage information on how many batteries are instantiated, and on the communication architecture itself.
- Since processors, active or passive links may each fail independently, or may have correlated failures, multiple *failure models* have to be supported. The simulation environment should enable modeling of failure probabilities, mean duration of failures and failure probability distributions, for each of the nodes and communication links. Furthermore, failures between nodes and links may be correlated and a *correlation coefficient* can be specified between any given network segment (active or passive) and any processing element (node). In TANs, correlated failures may appear due to situations such as power and communication being simultaneously severed (as it is likely that all system wiring will be bundled together).

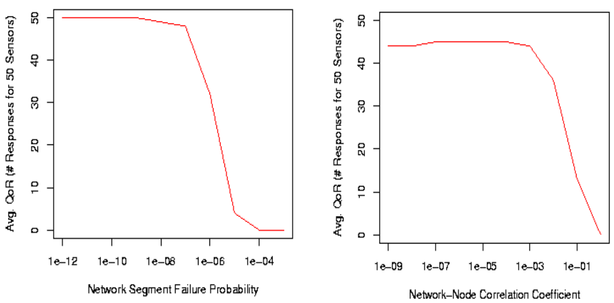

Fig.7 Impact of link failure rates and failure correlation among links and nodes on quality of results (QoR)

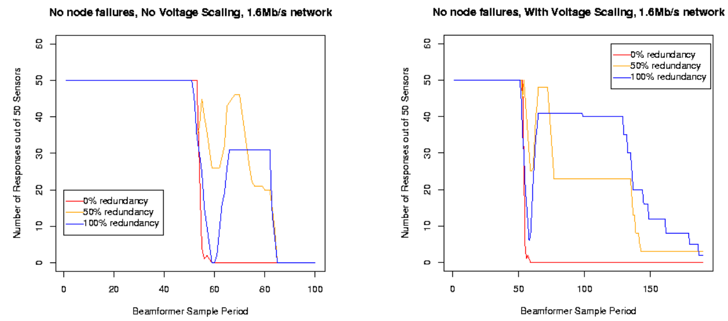

Fig.8 Effects of dynamic voltage scaling during code migration

- Alternatively, failures may appear due to ambient conditions such as high humidity or electromagnetic interference leading to failures in both nodes and links. Lastly, nodes may fail as a side-effect of failures in the network (for example, increased activity can lead to increased peak power consumption, leading to increased failure probability).

To validate results achieved via analytical modeling, we have developed a simulation environment which provides an infrastructure for investigating the many aspects of e-textile based systems. Specifically, we have investigated the impact of link failure rates and correlations among failures on QoR, as well as the feasibility of code migration, with and without voltage scaling.

We show in Fig.7 some results on the effect of *link failure rates*, as well as *correlation among link and node failures* on the operational longevity. The driver application in both cases is the acoustic beamforming application which was configured for 50 sensors that retrieve information and do some local processing before sending data to a central node doing selection of the beam signal. In Fig.7 we show the quality of results (defined as the average number of responses received from slave nodes per time unit) as a function of the network failure rate and the network-node correlation coefficient. As it can be seen, increased link failure rates affect dramatically the quality of the results, while shortening significantly the operational longevity of the system. Correlation among node and link failures have also a negative effect on the overall system quality.

Similar behavior can be observed if a real battery model is used for the local power source (per region or cluster). In such cases, since battery capacity information can be used to *predict* when the node will fail due to insufficient power resources, process migration on a spare/redundant node can be scheduled accordingly. The amount of redundancy, the speed of the communication medium, as well as the communication architecture all have a significant impact on the migration process. The more nodes try to migrate on spare, fully functional nodes, the higher the collision rate on the communication medium is, and thus, the quality in fact decreases. As shown in Fig.8, having more redundancy does not necessarily mean better quality for the same operational lifetime, due to finite speed and capacity of the communication channel. However, since process migration is a non time-critical task, to alleviate this problem, dynamic voltage scaling can be used during the migration process. By slowing the clock speed in combination with voltage supply scaling, collisions due to concurrent process migrations may be avoided and better operational longevity can be achieved with increased redundancy, as shown in Fig.8.

7. CONCLUSION

In this paper, we have addressed a few fundamental issues of an emerging new field of research called *electronic textiles*. Since this new venue of research is likely to introduce *revolutionary* changes in the classic design style of today electronic systems, there will be significant long term impact on both industry (potential e-textile manufacturers, CAD vendors, IP providers) as well as academic research community. Although production of e-textile-based products is limited today, it is very likely to grow sometime in the near future. Thus the DA community has to be ready to deliver tools and techniques for designing, testing and reconfiguring such products. We believe that considering fundamentally new issues in the context of e-textiles can potentially harvest fruitful research and significant results with major impact for the DA community at large.

8. ACKNOWLEDGMENTS

The authors would like to thank the students in CMU's *COATNET* team (especially Phillip Stanley-Marbell and Ted Pham), as well as Bob Graybill of DARPA for insightful comments on our work.

REFERENCES

[1] E. R. Post and M. Orth. Smart Fabric, or Wearable Clothing. In *Proc. Intl. Symp. on Wearable Computers*, pp. 167-168, Oct. 1997.

[2] E. R. Post, M. Orth, P. R. Russo, and N. Gershenfeld. E-broidery: Design and fabrication of textile-based computing. *IBM Systems Journal*, 39(3&4):840-860, 2000.

[3] M. Gorlick. Electric Suspenders: A Fabric Power Bus and Data Network for Wearable Digital Devices. *Proc. Intl. Symp. on Wearable Computers*, Oct. 1999.

[4] J. Rantanen, T. Karinsalo, M. Mkinen, P. Talvenmaa, M. Tasanen, and J. Vanhala. Smart Clothing for the Arctic Environment. *Proc. Intl. Symp. on Wearable Computers*, pp. 15-23, Oct. 2000.

[5] R. Marculescu and D. Marculescu. Does $Q=MC^2$? (On the relationship between quality in electronic design and model of colloidal computing). *Proc. IEEE/ACM Intl. Symp. on Quality in Electronic Design*, March 2002.

[6] Paul C. Hiemenz, Raj Rajagopalan. Principles of Colloid and Surface Chemistry. Chapters 1 and 13, Marcel Dekker, 3rd Edition, 1992.

[7] Electronic Textiles Concept Session, STA/DARPA, Jan. 2001.

[8] R. Marculescu, A. Nandi. Probabilistic Application Modeling for System-Level Performance Analysis. In *Proc. DATE*, Munich, Germany, March 2001.

[9] B. D. Van Veen, K. M. Buckley. Beamforming: A Versatile Approach to Spatial Filtering. In *IEEE ASSP Magazine*, April 1988.

[10] A. Nandi, R. Marculescu. System-Level Power/Performance Analysis for Embedded Systems Design. In *Proc. ACM/IEEE Design Automation Conference,* Las Vegas, NV, June 2001.

[11] P. Stanley-Marbell and D. Marculescu. Code Migration in Low Power Networked Embedded Systems.CMU-CSSI Technical Report, 2002.

[12] K. R. Dixon, T. Q. Pham, and P. K. Khosla. Port-based adaptable agent architecture. In *Proc. Intl. Workshop on Self-Adaptive Software*, 2000.