A co-designed hardware/software architecture for augmented materials

Simon Dobson[1], Kieran Delaney[2], Kafil Mahmood Razeeb[3], and Sergey Tsvetkov[4]

[1] Department of Computer Science, University College, Dublin IE
simon.dobson@ucd.ie
[2] Cork Institute of Technology, Cork IE
kieran.delaney@cit.ie
[3] Tyndall National Institute, Cork IE
kmahmood@tyndall.ie
[4] Department of Computer Science, Trinity College, Dublin IE
sergey.tsvetkov@cs.tcd.ie

Abstract. Recent advances in materials, sensing, power harvesting, context-awareness and miniaturisation have opened-up the possibility of constructing materials that directly include considerable computing power. We present an architecture for the hardware/software co-design of such "augmented" materials that allows designers to address the links between the physical and informational properties of artefacts. The architecture is highly modular in terms of sensor, communications, power and processing capabilities, and utilises an advanced semantically well-founded software model.

1 Introduction

The inner workings of today's technologies (and those of any era) are always notionally complex to everyday users, and the same can be said of objects made from such technologies. The process from which these objects are made to appear intuitive and simple hides a multitude of technological and conceptual development cycles, reaching a stage where their capabilities are well-proven, trustworthy, easily understood, and effectively taken for granted.

Recent advances in materials technology, sensor and processor miniaturisation, power harvesting and context-aware software have opened-up the possibility of constructing "augmented" materials that include computational intelligence and communications directly into the fabric of artefacts. These materials can be

---

† This paper appears as Simon Dobson, Kieran Delaney, Kafil Mahmood Razeeb, and Sergey Tsvetkov. A co-designed hardware/software architecture for augmented materials. In *Proceedings of the 2nd International Workshop on Mobility Aware Technologies and Applications*, LNCS, 2005. To appear.

used for a range of applications where designers need to combine physical and informational effects very closely. By co-designing the physical, hardware and software aspects of such materials it is possible to leverage the strengths (and address the weaknesses) of the individual components to achieve a seamless pervasive technology.

Our focus on materials is driven by four observations:

1. the useful behaviour of an object is a function of the capabilities of its component materials, both physical and informational;
2. many behaviours depend both on the geometric shape adopted by the object; and its relationships with other objects;
3. complex behaviours and interactions are more easily understood if composed from discrete and recognisable objects; and
4. the close correspondence between physical affordances and information effects is what allows simple, scrutable pervasive computing.

We believe that – by broadening what we mean by the “behaviour” of materials to include sensing, processing and communications, as well as physical, aspects – it is possible to develop an architecture for co-designed pervasive computing systems expressed in terms of material interactions. Materials are infused with systems capability that allows a digital representation to be developed at a selected formation stage (*e.g.* curing) and maintained thereafter. To pursue the analogy, an augmented material is an alloy of a traditional material and a processing capability that yields behaviours different from those of its individual components. An effective implementation yields a situation where any subsequent materials processing capability behaves as a programming step, affecting the digital as well as physical characteristics of the material.

In this paper we describe an architecture for augmented materials. The architecture is highly modular in terms of sensor, communications, power and processing capabilities, and utilises an advanced semantically well-founded software model. By integrating materials properties, sensing and semantics we aim to deliver a robust and flexible platform for self-managing, autonomic smart systems and artefacts.

Section 2 motivates our work and places it into the context of pervasive computing research. Sections 3 and 4 respectively describe the hardware and software components of the architecture, with particular emphasis on their complementarity. Section 5 contrasts our approach with other work in the field, especially in wireless sensor networks, while section 6 concludes with our immediate future directions.

2 Design space and issues

Materials science has expanded greatly in the past few years with contributions from nanotechnology as well as from traditional physics. However, even the smartest of “smart” materials lacks the flexibility and sophistication of software. Augmenting materials with information technology addresses this gap, allowing materials to process information as well as functioning physically, reflecting on its behaviour and providing feedback to the wider IT environment.

An augmented material might be used as part of a rigid structure such as an airframe. With embedded sensing the material can detect changes in its physical composition. It can provide ground crew with an on-going report on its health or, in case of damage or failure in-flight, inform the flight control systems to take remedial action such as reducing the manœvering envelope to avoid excessive strain. These observations may be performed using piezo-electric sensors, and are used externally on some airframes: internalising the sensing is more robust and decreases the load on the central computers. Similar techniques can be applied to other built structures such as buildings, bridges and ships.

At a more personal level, an increasing number of people undergo physiotherapy to accelerate the healing of damaged bones. A major problem occurs in monitoring a treatment regime to ensure that the patient performs the necessary exercises but does not over-tax the injury. A augmented cast could monitor the impact on the supported limb to ensure that it remains within acceptable limits, providing feedback to the patient and therapist. A material whose stiffness is variable could actively modify its physical characteristics according to a complex scheme encoded in software.

Truly pervasive computing applications are limited by the availability of sufficiently small devices on which to run applications. Typically one encounters smart-building systems in which artefacts are tagged with RFID and tracked by computers in the walls. This is an asymmetric solution, as the artefact cannot perform actions autonomously. By constructing artefacts with augmented materials the symmetry is restored, allowing us to build (for example) books that know their location and relationships to other artefacts. This is a more peer-to-peer solution than is possible at present, and relies critically on the ability to place processing alongside sensing in the fabric of everyday things.

There is a body of work on sensor miniaturisation, sensor networking, pervasive and amorphous computing (section 5). What makes the current context novel is the application of these technologies specifically to new materials from which to construct objects, rather than as addenda to existing objects.

At the physical level, an augmented material consists of the substrate material and a (possibly extremely large) collection of embedded processor elements with local sensing and communications. The choice of substrate governs the gross physical properties of the augmented material, and so is conditioned by the final application. Possible choices include flexible and non-flexible plastics, fibre-glass, fabrics, concrete and metals – indeed *any* material whose application would benefit from augmentation with IT.

The added value of an augmented material comes from its additional interaction capabilities, which in turn derive from the software it can support. Such materials offer significantly different characteristics to more traditional embedded systems, both in terms of the number of elements involved (potentially thousands) and the unusually close relationship between the elements and the material in which they are embedded.

We may view an artefact constructed from an augmented material at four distinct levels.At the **physical** level, the material exhibits certain structural properties such as stiffness, ductility, conductivity and so forth, which condition the physical applications for which it can be used. At the **element** level, each processing element in the material functions as an independent component capable of local sensing, local processing and communications with nearby elements. At the **material** level the individual elements co-ordinate their actions to present a global behaviour, typically integrating the local sensing information into a global view of the material. At the **artefact** level the material can "understand" its function and relationships with other augmented materials in its vicinity. An augmented-material book lying on an augmented-material table provides a good illustration of the four levels of concern.

This model also illustrates the challenges facing augmented materials. Although each individual element may provide local sensing and processing, element-level observations are of little use in isolation: they need to be integrated at the material level. Even given this integration, the significance of the various observations can only be determined by using quite sophisticated reasoning informed by a knowledge of the real-world environment (artefact level). At the element level we need to be able to drive the available sensors, perform calculations and communicate with other elements. These individual observations must be combined at the material level to provide an integrated view of the material, and used to drive inferencing at the artefact level. None of these problems is amenable to direct programming solutions of the kind normally found in embedded systems.

3 Sensing and processing

These considerations lead us to a *co-designed* architecture in which we consider the hardware (material, sensing, processing, communication) and software (knowledge, reasoning, task) aspects together. The various properties of augmented materials can then be traded-off against each other to provide a self-configuring and robust information platform.

Making a material self-aware involved two stages. At the first stage (curing), the elements establish a "baseline" view of their configuration. The second stage (lifetime) updates this representation over the material's lifetime. As an example, consider a flexible sheet of plastic containing elements with local strain sensors. At the first stage it establishes its initial shape (flat) and the relative locations of its elements; during its lifetime it updates its view of its own shape by integrating the local strain observations of the elements. Such a material "knows" its own shape on an on-going basis, and this can be used to inform its more advanced behaviours.

3.1 Communications and location

There are three aspects of the communication for augmented materials: between sensors and processors, between elements and between materials. Sensor/processor communications can be performed locally by integrating them onto the same device. Inter-material communications can occur either wirelessly when elements at the edge of one material come into range of elements at the edge of another, or through wires if materials are plugged together or connected to the wider population of internet servers.

Inter-element communication is perhaps the most interesting. Communications may be point-to-point or broadcast; point-to-point will typically use wires run through the matrix; broadcast may use radio, conduction through the substrate, or even acoustics. The point is to match the communications requirements to the properties of the substrate and the target application.

Since elements are returning information about their locality, many applications will require that they know their position in the material matrix in some co-ordinate system in order to correlate observations with locations. It is generally possible to identify element locations at fabrication time, for example placing elements in a uniform grid, and then informing each element of its location at the curing stage. While this is the simplest solution, it makes fabrication more difficult and may introduce errors if elements drift during processing.

A more attractive solution from a fabrication perspective is to locate elements randomly within the substrate and have them work out their location post-fabrication. Locating an element in this way is an interesting problem in its own right, but that we believe self-location of nodes with wireless communication to be feasible given assumptions of a sufficiently dense network with a sufficient number of elements, suitably distributed, whose locations are known.

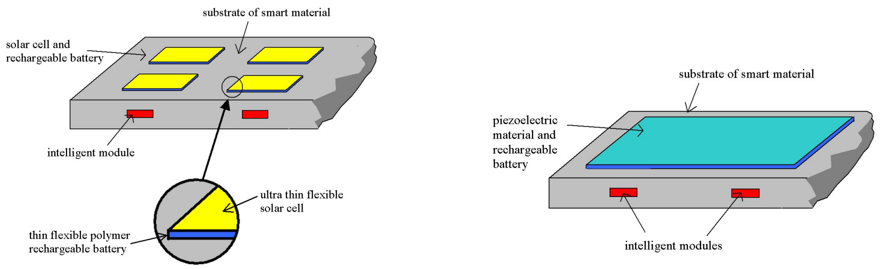
3.2 Current state

We have designed and constructed elements for use within a range of substrates across the design space described above. Our design is highly modular, allowing individual components to be changed within the overall design framework. Modularisation includes communications, sensing, processing, memory and power source. It allows us to (for example) change the communications technology independently of the sensors, or to deploy additional sensors at the expense of memory or processing power.

Fig. 1. 25mm prototype module with coin battery

Our current hardware platform[2, 10] consists of a number of 25mm-on-a-side FPGA-based elements (using Xilinx Spartan IIE cores) placed in formation without a substrate material (figure 1). We are currently evaluating the sensing, power management and inferencing capabilities of this platform, using the results to inform the design of the next steps on the roadmap: 10mm and 5mm elements, embedded in a flexible plastic.

The communications transceiver consists of a fully integrated frequency synthesiser, a power amplifier, crystal oscillator, modulator and antenna. Output power and frequency channels are easily programmable. Current consumption is very low, and a built-in power-down modes makes power-saving easily realisable. The designer can easily program the device with custom protocols. The element's sensor array connects though a dual-channel RS232 transceiver. The module contains two A-to-D converters for interfacing up to seven external sensors that change their resistance according to the parameter being measured. The modules use a stackable connector system to make the electrical and mechanical interconnections between themselves.



(a) Solar cell array (b) Piezoelectric battery charging

Fig. 2. Possible power harvesting approaches

Power supply is one of the major obstacle towards the miniaturisation of the autonomous sensor nodes. Coin cells may be used to provide power to the system, connecting directly using the stackable connector system, but power harvesting is obviously a much more attractive option for a long-lived, autonomous material. Figure 2 shows two examples of a smart material utilizing an array of ultra-thin flexible solar cells as the energy harvesting power source and an flexible polymer rechargeable battery for energy storage (figure 2(a)), suitable for outdoor use, and an alternative piezoelectric crystal that can generate small currents from vibrations to charge the rechargeable batteries (figure 2(b)). These are laminated onto the augmented material.

4 Programmability

Individual elements present a diverse set of possibilities. Although each element shares a common microcontroller core (and so can potentially run the same software), the population of sensors, actuators and other devices can vary widely. Open-ended behaviour means that augmented materials are not amenable to direct programming solutions of the kind normally found in embedded systems, so we have adopted a less familiar (but more powerful) approach based around rich, scalable, self-organising context models and inference.

Our overall goal is to integrate programming, as far as possible, into the process of manufacturing augmented materials, and to capture clearly the relationship between factors affecting the material and their behavioural effects[6].

4.1 Programming elements

We took the decision to mirror the structure of the hardware module in the software platform. The initial population of software components mirrors the design choices made in hardware. As well as simplifying configuration, this approach allows us to make modules "reflective", in the sense that they "know" their sensing and other capabilities from the outset.

In the wider field of pervasive computing, many systems have adopted a more knowledge-based approach, modeling context as a collection of assertions (a good example is [13]). Typically this involves combining a basic knowledge representation with an ontology management system for expressing the constraints on knowledge. Given the constraints of power and space we are working under with augmented materials we have chosen to use RDF[7] for representing context, decoupling the data model from its usual XML format to compress the knowledge base sufficiently to operate on a microcontroller. RDF structures knowledge in terms of $(subject, predicate, object)$ triples. Each triple captures a single binary relationship (represented by the predicate) between two entities (the subject and the object). The collection of triples can be read as a concept graph, with each triple defining a labelled edge. The available predicates for a vocabulary for talking about a particular sort of knowledge, with the relationships between the predicates being structured by an ontology. The subjects are the "denotable values" in the knowledge domain, typically either identifiers for things or literal values. Predicates are also URLs, allowing unique labeling of predicates within an ontology without a centralised registry.

Each sensor on an element has an associated vocabulary providing predicates for each type of sense data. A strain sensor, for example, might define predicates for the strain in newtons and the direction of strain in some co-ordinate system. For each sensor, the device driver maps the sense data into the local context model using this representation (figure 3).

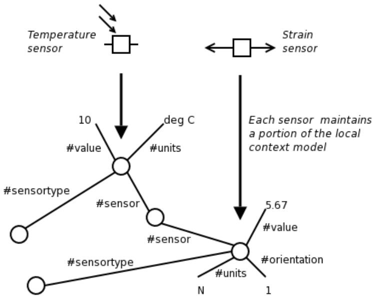


Fig. 3. An individual model held on an element

The advantage of this approach is three-fold. Firstly, it provides a common framework within which to represent and query *all* sensed data for any client that knows the predicates. Secondly, it allows the same information to be sensed by different sensors in a manner that is largely transparent to the programmer, simply by sharing vocabularies. Thirdly, it raises the abstraction level for programmers away from hardware and into the knowledge plane, making it easier to express logical constraints and queries.

4.2 Programming materials

A single augmented material might include several hundred sensor and processing elements. Current techniques to embedded systems stress programming the *devices* as the basis for applications, which will not scale up to such environments. Instead we need to program *materials as a whole.*

Since we are focusing on augmented materials, we are dealing with a more limited application domain than might be found in general amorphous computing[1]. Specifically, we assume that applications are primarily concerned with reacting to the *physical environment* of the material and its *physical and informational properties.* This establishes a close correspondence between the material and the elements within it: the location of an element within the substrate will typically be a significant factor in interpreting its information.

Given a located element, its sensor observations can be related directly to the environment and/or operation of the augmented material. In one sense integrating sense data is simple: any element wanting to know the state queries the model on each appropriate element and performs the necessary numerical integration. In a system with low-power unreliable elements and local communications, however, things are not so simple.

A naïve solution would nominate a single master element to maintain a global view of the material. Such a node would require significant storage and computing power and would be a single point of failure for the material. At the other end of the spectrum, nodes would exchange their local states with all other nodes *via* a gossiping protocol[8] .Gossip protocols are used extensively in *ad hoc* allowing more queries to be answered using only local knowledge.

While a fully decentralised implementation has its attractions, distributing even aggregated data from the entire material across all elements greatly increases their computational, storage, communication and (most importantly) power requirements. Locality means that elements do not in general need to know results from remote elements in the material (although they may need to be able to find out special cases). We believe that a hybrid approach offers a good trade-off between decentralisation and efficiency. Elements are divided into two categories – sensing elements and aggregating elements – which are then evenly distributed through the substrate. Aggregating elements can have more storage capacity than sensing elements.

The two classes gossip, but in different ways. Sensor elements gossip with nearby aggregating elements by sending changes in their local states, which may then be aggregated to provide a summary of the state of the local area. The intention is to provide explicit storage and computational capacity for these summarising activities. By summarising locally the system scales well as the material grows (assuming a “fair mix” of elements) and requires only local communications.

Aggregate elements gossip with other aggregate elements, but exchange management information about which aggregate is summarising what locale. The protocol essentially distributes the set of subjects and predicates that an aggregate is holding, providing an RDF-level summary of the information stored on the element. When a query is made to an aggregating element, it can determine which elements can have information that may satisfy the query and perform the appropriate sub-queries. In effect the material behaves as a single RDF model that can be queried as a whole, with queries being decomposed and distributed as required. This structure means that data is held locally within the material but accessed globally. The overall information content of a material is available *via* any aggregate element using an identical interface, regardless of the actual location of the information in the substrate.

4.3 Internal *versus* external semantics

While the low-level sensor data can be managed largely automatically, being tied closely to the physical realisation of the material, inferred information is handled using rules provided by the programmer within a truth-maintenance framework in which changes in lower-level information propagate to higher-level views inferred from that information.

However, it is important to realise the way in which sense data – the "internal semantics" of the material – relates to the higher-level, "external" semantics of pervasive computing. A particular type of augmented material (for example a rigid plastic) can be used to form any number of classes of objects, each of which will exhibit different user affordances and behavioural relationships. This means that a material must know what it is externally as well as knowing its internal structure.

The external semantics is provided by describing the interaction rules by which the material should interact with other materials to which it is connected. This common use of rule bases leads to a single level of programming, built on top of an application-neutral sensing, communications and reasoning architecture. The external interactions of a material can use standard approaches such as web services and RDF (in its XML transfer format). This makes it easier to integrate artefacts using augmented materials into the wider context of pervasive computing.

4.4 Current state

We are exploring a number of possible programming approaches ranging from the pure logic programming outlined above to a more flexible approach based around domain-specific programming languages[9], building on the substrate of a compositional language technology based on Scheme[4]. The advantage of this approach is that we can leverage the benefits of type systems and language constructs while targeting micro-scale devices.

5 Related work

One of the most relevant research activities on autonomous systems worldwide is the "smart dust" project at the University of Berkeley. In this project, the goal is to develop a system of wireless sensor modules where each unit is the size of a mote of dust of around 2mm × 2.5mm – although each mote still needs external power and antenna, and so in practice is somewhat larger. This sets the design challenge for augmented materials, focusing on the design of miniaturised antennæ and power harvesting as well as power management and programmability.

The essential problem of communication within *ad hoc* networks has been the subject of vigorous research within the wireless sensor community. While a number of authors have studied localisation using infrastructurally-positioned beacons, recent work (for example [3, 12]) has addressed the problem of self-localisation of a group of nodes with sparse knowledge of their initial positions. We believe that this work may be extended to address augmented materials, with the crucial simplification that edge effects may provide additional localisation cues.

Software for pervasive systems ranges from the conventionally-structured desktop- and room-oriented approaches typified by the Context Toolkit[11] to the more chaotic approach of amorphous computing[1]. Although the latter remains largely conceptual, we believe that it provides good pointers to environments for augmented materials.

Some notable work has taken place under the European Union's "Disappearing Computer" initiative. The Extrovert Gadgets project integrated sensor networks into everyday objects. The GLOSS project investigated co-ordinating the global behaviour of pervasive systems. These (and other) projects have helped establish the broader landscape within which augmented materials must function.

6 Conclusion

We have presented a general approach to the problem of augmenting materials with embedded sensing and processing elements in a way that can be used to construct artefacts that combine physical and informational capabilities. This involves answering two distinct but related sets of questions:

- what are the hardware considerations in terms of location, communication, sensing and power involved in building a co-operative network of sensor elements?, and
- what is the appropriate programming model for applications on such a constrained platform?

Our tentative answers to these questions form the basis for an architecture for augmented materials in which a heterogeneous collection of low-power elements is co-ordinated by means of a hierarchical context model programmed in a highly declarative, whole-material style. We believe that this combination provides a good trade-off between the constraints encountered in building such augmented materials.

While our work is at an early stage, we have demonstrated the adequacy of the individual components and the way in which co-design can be used to leverage the best possible functionality from even limited individual units in a way that will scale to large materials. A fuller demonstration is currently underway to integrate hardware and software into a sample substrate to explore the ways in which augmented materials can complement and advance the capabilities of pervasive computing.

References

1. Harold Abelson, Don Allen, Daniel Coore, Chris Hanson, George Homsy, Thomas Knight, Radhika Nagpal, Erik Rauch, Gerald Jay Sussman, and Ron Weiss. Amorphous computing. *Communications of the ACM*, 43(5):74–82, May 2000.
2. Stephen Bellis, Kieran Delaney, John Barton, Kafil Razeeb, Brendan O'Flynn, and Bivragh Majeed. Development of field-programmable modular wireless sensor network nodes for ambient systems. *Computer Communications*, 2005. To appear.
3. Nirupama Bulusu, John Heidemann, Deborah Estrin, and Tommy Tran. Self-configuring localization systems: design and experimental evaluation. *ACM Transactions on Embedded Computing Systems*, 3(1):24–60, February 2004.
4. Simon Dobson. Creating programming languages for (and from) the internet. In *Workshop on Evolution and Reuse of Language Specifications for domain-specific languages, ECOOP'04*, 2004.
5. Simon Dobson, Kieran Delaney, Kafil Mahmood Razeeb, and Sergey Tsvetkov. A co-designed hardware/software architecture for augmented materials. In *Proceedings of the 2nd International Workshop on Mobility Aware Technologies and Applications*, LNCS, 2005. To appear.
6. Simon Dobson and Paddy Nixon. More principled design of pervasive computing systems. In Rémi Bastide and Jörg Roth, editors, *Human computer interaction and interactive systems*, volume 3425 of *Lecture Notes in Computer Science*. Springer Verlag, 2004.
7. Ora Lassila and Ralph Swick. Resource Description Framework model and syntax specification. Technical report, World Wide Web Consortium, 1999.
8. Alberto Montresor, Márk Jelasity, and Ozalp Babaoglu. Robust aggregation protocols for large-scale overlay networks. In *Proceedings of the International Conference on Dependable Systems and Networks*, 2004.
9. Andy Nisbet and Simon Dobson. A systems architecture for sensor networks based on hardware/software co-design. In Mikhail Smirnov, editor, *Proceedings of the 1st IFIP Workshop on Autonomic Communications*, volume 3457 of *LNCS*. Springer Verlag, 2005.
10. Kafil Razeeb, Stephen Bellis, Brendan O'Flynn, John Barton, Kieran Delaney, and Cian O'Mathuna. A hybrid network of autonomous sensor nodes. In *Proceedings of the Second European Symposium on Ambient Intelligence*, ACM International Conference Series, pages 69–70, 2004.
11. Daniel Salber, Anind Dey, and Gregory Abowd. The Context Toolkit: aiding the development of context-enabled applications. In *Proceedings of the ACM Conference on Computer-Human Interaction, CHI'99*, pages 434–441, 1999.

12. Xi Wang, Fabio Silva, and John Heidemann. Infrastructureless location-aware configuration for sensor networks. In *Workshop on Mobile Computing Systems and Applications*, pages 174–183. IEEE Press, 2004.
13. Xiaohang Wang, Jin Song Dong, Chung Yau Chin, Sanka Ravipriya Hettiarachchi, and Daqing Zhang. Semantic Space: an infrastructure for smart spaces. *IEEE Pervasive Computing*, 3(3):32–39, July–September 2004.