Distributed Intelligent MEMS: Progresses and perspectives

Julien Bourgeois and Seth Copen Goldstein

Abstract MEMS research has until recently focused mainly on the engineering process, resulting in interesting products and a growing market. To fully realize the promise of MEMS, the next step is to add embedded intelligence. With embedded intelligence, the scalability of manufacturing will enable distributed MEMS systems consisting of thousands or millions of units which can work together to achieve a common goal. However, before such systems can become a reallity, we must come to grips with the challenge of scalability which will require paradigm-shifts both in hardware and software. Furthermore, the need for coordinated actuation, programming, communication and mobility management raises new challenges in both control and programming. The objective of this article is to report the progresses made by taking the example of two research projects and by giving the remaining challenges and the perspectives of distributed intelligent MEMS.

1 Introduction

Microelectromechanical systems (MEMS) have reached a state of design maturity which has led to some interesting prototypes and profitable products. While most MEMS devices have been used as independent elements of a larger system, this article deals with distributed MEMS systems composed of many MEMS devices which work together to achieve a global goal. The distinguishing feature of MEMS devices is that they are small and that they

Julien Bourgeois
University of Franche-Comté, 1 cours Leprince-Ringuet, 25200 Montbeliard, France
e-mail: Julien.Bourgeois@univ-fcomte.fr

Seth Copen Goldstein
Carnegie Mellon University, Pittsburgh, PA 15213, USA
e-mail: seth@cs.cmu.edu

can be efficiently mass-produced. This naturally engender thinking of how they can be used together as a distributed system. Due to their small size, their low-cost and the fact that they can be mass-produced, millions of units can be used in very small space. For example, a volume of less than 1 m^3 of 1mm-diameter silicon balls contains as many nodes as in the Internet. This characteristic requires paradigm-shifts both in hardware and software parts.

Past research focused on challenges of the engineering process, future challenges will consist in adding embedded intelligence to MEMS systems, so that they will be able to collaborate efficiently. This will require embedding MEMS sensors/actuators, electronics, communication capabilities, control of actuators and programs in the same unit. We suggest the use of the phrase "distributed intelligent MEMS (diMEMS)" when referring to such systems. DiMEMS systems will certainly contain heterogeneous units. However, to simplify the programming challenge, we consider in this article only systems composed of homogeneous units.

Designing and managing diMEMS inherently requires multiple disciplines (e.g. hardware and software research). The challenges are therefore present in every field of research as well as in the integration of all the parts. In the 90's, DARPA Information Science and Technology funded a study on the state of the art and the perspectives of distributed MEMS. The conclusions of this report [7], published in 1997, were that the challenges involved in realizing diMEMS were mainly in controlling large numbers of MEMS sensors and actuators, the emergence of distributed intelligence, the use of MEMS devices as computational elements and the multiple-energy-domain simulation, analysis, and design. This article examines these challenges and the new ones that have been identified since 1997 using the results of two projects that have been conducted in the field.

2 Challenges

Many of the challenges raised by diMEMS have been studied in isolation in different research fields. However, in diMEMS they must be examined together and they become even more extereme. The scale of diMEMS needs new software paradigms as well as requiring new hardware capabilities. In the main, scaling up is the main concern of software challenges, while scaling down is the main concern of hardware challenges.

2.1 Software challenges, scaling up

Scalability

Scalability is the main concern of diMEMS as the number of units will likely number in the millions. Scalability therefore impacts the way units will communicate. Systems using synchronous communications can't scale as well as those using asynchronous communications because the constraints of synchronization lower the system efficiency [6]. Synchronous communication is therefore difficult to achieve and asynchronous communications have already shown better results.

To ensure scalability, the programming model and the language must hide the complexity from the programmer and the compiler should enable programming the system as a single ensemble. Within the Claytronics project, new languages like LDP [11] and Meld [4] been have developed to cope with this challenge.

As scalability has to be tested up to millions of units, simulation tools also have to scale up. Dprsim [25] which has been developed within the Claytronics project has successfully simulated millions of units.

Uncertainty tolerance

Faulty behavior is inherent to any diMEMS system. This is due to several factors. The batch process used in MEMS fabrication creates different levels of reliability. While some of the devices will have no defects, most of them have a high percentage of failed units. On the software side, this characteristic has to be handled and fault-tolerance has to be implemented.

In the case of mobile distributed MEMS a logical topology has to be maintained in order to communicate between the units. Maintaining a logical topology over a physical one is the concern of many research topics like P2P [1], swarm intelligence [18, 23], ad hoc networks [26] or wireless sensor networks [29]. Mobile distributed intelligent MEMS is even more complex than these examples. Mobility, scalability, fault-tolerance and limited processing capability are the main challenges to solve in order to create and maintain a logical topology.

Communications

The tradeoff between computation/communication/sensing is a challenge that has already been studied in wireless sensor networks but it needs some adaptation to take into account the scalability factor which is inherent to diMEMS.

Each diMEMS project has its own communication model directly linked to the application. The question here would be to study the cost and the interest of having some abstraction layer.

Control

When each unit of a system is mobile, the changes in the physical topology modify the logical topology by changing network connectivity. This is one of the main concerns of MANET. The inverse is also true, the logical topology can drive the mobility. This is usually done by covering an area which needs to be sensed [27] but it can also be used for modifying the logical topology, for example to keep connectivity in a sensor network [14].

DiMEMS are composed of actuators which needs control and a degree of synchronization. Three synchronization schemes can be used: no synchronization between MEMS units, means that the control loop doesn't have to synchronize with other units, local synchronization means that a MEMS unit has to be synchronized with its neighbors while global synchronization means that all the MEMS units have to act synchronously.

Having actuators to control requires real-time deadlines. Some applications need a very high frequency from the controler. If the control is fully decentralized like it is the case in distributed intelligent MEMS and that different modules requires a local or a global synchronization, the time to communicate has to be very short.

Reliability through properties verification

Reliability is difficult to achieve in any information technology project. The approach taken to achieve it often uses modularity which allows one to define interfaces and to segment the causes of failures. In diMEMS, this modularity is limited. Methods to model the whole system with VHDL-AMS and UML/SysML would allow one to verify some properties of the system and to increase its reliability.

CPS and IoT, relations with macro world

DiMEMS are systems that can interact with other intelligent systems. This interaction is the focus of cyber-physical systems (CPS) [22] and Internet of Things (IoT) [5]. The new challenge with distributed inteligent MEMS systems is to manage the different density of communication between the macro-objects (low density) and the micro-objects (high density).

2.2 Hardware challenges, scaling down

Seamless integration of MEMS and logic

Integrating MEMS with CMOS is still a challenge in the fabrication process [30, 15]. Most of the MEMS-CMOS integration follow a hybrid integration through wire-bonding but this approach is not well-suited for diMEMS which requires too many connections. DiMEMS requires a monolithic integration for two reasons. First of all, only a monolithic approach can guarantee scalable and affordable fabrication process whereas hybrid approaches often requires manual intervention. Secondly, the weight of a hybrid system is more important than a monolithic one. MEMS actuation requires higher voltages than logic which can create problems. Some solutions have been proposed [20] to tackle this problem but a real voltage difference management between actuation and logic is still a challenge.

Designing robust MEMS

Due to their size, MEMS are very sensitive to external factors (e.g., dust and air quality) which can change the behavior of certain types of MEMS actuators. Modeling and simulation have proven to be efficient to solve design issues [10] but new solutions have to be found to increase MEMS robustness.

Building micro-communication devices

DiMEMS needs communication capabilities, but integrating communication and MEMS is still a significant challenge. Some of them are linked to the previous challenges described here like the voltage difference between the actuation and the logic and the integration of MEMS and logic but scalability is also an issue. One-way communication to the MEMS has been implemented, for example, in DMD where each micro-mirror has to be oriented is the right direction [19]. While this approach works for a fixed topology and even for a regular network of actuators, it doesn't scale for more complex devices or mobile units. Building a real micro-communication device is still a challenge that haven't been solved yet.

3 Examples

Many projects have been built around MEMS but no killer application has yet been realized. Two active projects which are studying diMEMS are the Smart projects (Smart Surface and Smart Blocks) [9] and Claytronics [17].

3.1 The Claytronics project

3.1.1 Introduction

The past several decades have demonstrated the remarkable power of *programmability*: general-purpose computers have enabled new application domains that were not anticipated when these computers were designed. As impressive as cyberspace is these days, it is useful to remember that we live in a world that is physical and not merely virtual. To enable new classes of exciting applications that may stretch our imaginations (e.g., shape-shifting medical instruments, hands-on interactive 3D design, etc.), the claytronics project aims to bring the power of general-purpose programmability to everyday physical artifacts in a fundamental way through a new form of *programmable matter*. In the long term, our goal is to construct the programmable matter such that its shape, motion, appearance, and response to human touch can be arbitrarily controlled by software.

Our vision for realizing programmable matter is to harness the collective power of a vast number (perhaps millions) of tiny (e.g., millimeter sized) spherical robots that can stick together and move around each other to form an overall material with somewhat fluid properties that we call "claytronics". The name "claytronics" was inspired by the word "claymation," since the material might resemble a form of modeling clay that can shape itself. We refer to the individual robots that make up this ensemble as "catoms", which is short for "claytronics atoms".

3.1.2 Hardware results

At first glance the ability to create a coherent ensemble of millions of units appears fantastical. But, if we step back and examine it, the question is not "if" we can manufacture it, but "when." It is clearly possible to do so in principle, e.g., biology builds ensembles of units which coordinate together to form dynamic 3D shapes which can interact in the real world. And, we already have MEMS processes which creates 3D devices. Reid has constructed spherical shapes by first printing a projection of the sphere and then, by harnessing the inherent stresses in thin film silicon dioxide, causes the projection to self-assemble into a sphere (See Figure 1) [24, 28]. This same process can be applied to a pre-fabricated CMOS wafer to create 3D units with integrated processors and actuators.

3.1.3 Software results

In pursuit of our goal we are exploring different programming approaches and have developed two new programming languages: LDP [12, 2] and Meld [3].

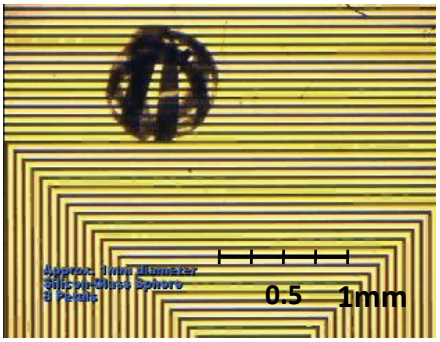

Fig. 1 *An example of a spherical shell (with diameter of ˜0.9mm) made using standard photolithography and stress induced curling. The shell (in the upper left) is sitting on a circuit board which can move the shell using electrostatic forces. Included with the permission of Rob Reid.*

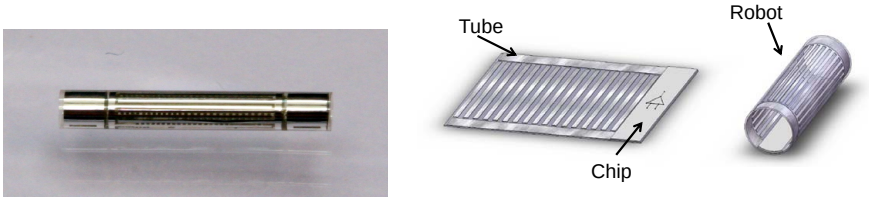

Fig. 2 *An example of a complete cylindrical catom. On the left if a cad drawing of the silicon shell and compelted robot. On the right, the realized result: you can see the logic die inside the exterior silicon shell. Included with the permission of M. Karagozler.*

Both of these languages are declarative in nature and result in programs which are about 20x shorter than equivalent imperative programs. They each take an ensemble perspective, allowing a programmer to create simple, concise, programs that are automatically compiled down to programs that run on each unit.

One of the advantages of concise programs is the ability of the programmer to focus on the program logic, facilitating correctness and affording greater opportunity for algorithm enhancements (like optimizations). In the case of larger programs we often see better performance in the Meld implementation than an equivalent C++ one. The main reason for this is that Meld and LDP make it easier to write a parallel program than a sequential one, utilizing much of the latent parallelism inherent in the algorithm, while the C++ implementation is limited to the parallelism that the programmer can manage and explicitly encodes.

3.2 The Smart Surface project

3.2.1 Introduction

The objective of the Smart Surface project is to design a distributed and integrated micro-manipulator based on an array of micro-modules in order to realize an *automated positioning and conveying surface*. Each micro-module will be composed of a micro-actuator, a micro-sensor and a processing unit. The cooperation of these micro-modules thanks to an integrated network will allow to recognize the parts and to control micro-actuators in order to move and position accurately the parts on the Smart Surface. The parts are small, they cover a few numbers of micro-modules (e.g. 4×4).

Figure 3 shows one prototype of Smart Surface. The rectangular holes seen on the front-side are the air nozzles. Air-flow comes through a micro-valve in the back-side of the device and then passes through the nozzle. The advantage of this solution is that the micro-actuators, the most fragile part of the surface, are protected.

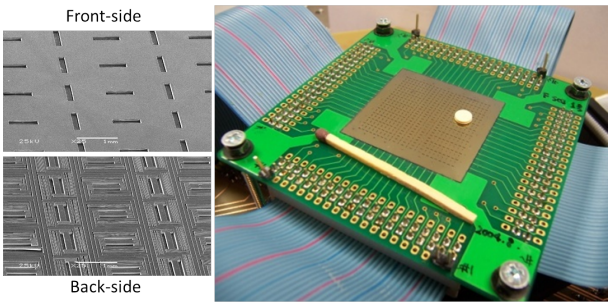

Fig. 3 Smart Surface prototype with grouped control of actuators

3.2.2 Hardware results

Three prototypes have been built within the Smart Surface project, these prototypes have different kinds of actuators, but their main difference can be found in the way the actuators can be controled.

Remotely placed actuators surface.

Each cell has oriented holes in 4 directions which means that each can create an airflow in the 4 directions. There is only one actuator per column so that

all the cells of a column have the same behavior. The cells are therefore passives as they don't have an actuator.

Grouped control of actuators.

This type of prototype is composed of actuators that can create airflow in 2 directions (see figure 3)and as the previous one, the actuators are controlled by column.

Individual control of actuators.

The more advanced prototype has the same actuators as the previous one, i.e. 2 directions but each actuator can be controlled directly which allows a finer control. This last prototype is the more complex to build as each actuator as to be connected to the controller.

3.2.3 Software results

Even in a technological project, software plays an major role. Four aspects have been studied within this project, sensor feedback, communications, control and modeling.

Sensor feedback.

Each cell that compose the surface comprises a processing unit, communication capabilities, an actuator and sensing capabilites. Sensing gives an indication on the presence of an object on top of it. More precisely, each sensor sends a binary information to its processing unit regarding the presence or the absence of the object. The object is therefore highly discretized. The first work has been to work on the differentiation possibility of highly discretized objects and on the choice of the best criteria to do so. The Exhaustive Comparison Framework (ECO)[8] has been designed in order to test exhaustively the efficiency of different differentiation criteria, in term of differentiation efficiency, memory and processing power needed. The second work has been to set the optimal number of sensors that have to be embedded inside the surface. The Sensor Network Calibrator (SNC) [13] allows to test different number and organization of sensors.

Communications.

The physical topology of the Smart Surface is a rectangle, and where each cell is connected to its four neighbours, the network topology is then a mesh 2D. As the network topology is known and fixed, the challenges are in the algorithmic part. A mathematical model of discrete state acquisition and several distributed iterative algorithms have been proposed and tested [6]. Synchronous and asynchronous state acquisition methods and the asynchronous ones have shown better results. Simple initial points and convergence results for distributed algorithms have been proposed. In both the synchronous and asynchronous cases, stopping criteria have been designed. The Smart Surface Simulator has been designed to evaluate and to validate experimentally the proposed distributed algorithms

Control.

Controlling the Smart Surface is very challenging as pneumatic actuation has many problems that need to be solved. The reinforcement learning method which has been implemented is decentralized and addresses the global-local trade-off [21]. An integration of sensing, communications and control has been proposed [9] and the experiments have shown good properties of the system.

Modeling

The VHDL-AMS model [10] that have been developed inside the Smart Surface project can simulate the behavior of the surface while the SysML model [16] gives a more higher-level description of the architecture. The SysML model is derived from the VHDL-AMS one and the objective is to link the SysML description of the hardware to the UML description of the software. This would allow properties verification for the whole system.

4 Conclusion

The concept of diMEMS has been proposed 15 years ago but only few of the challenges have been fully addressed. One reason is that diMEMS is a multidisciplinary field of research involving the topics of MEMS sensors/actuators design and fabrication, electronics, networking, programming and control. Having all these skills in one project is diffficult. Another reason is that the topic of diMEMS relies on MEMS technology and it couldn't exist until MEMS research reaches its complete maturity. This time of maturity has

come and we think that the time for pushing further the topic of diMEMS has come as well.

Acknowledgements We would like to thank all the Smart Surface and Claytronics groups and in particular Eugen Dedu, Kahina Boutoustous and Didier El Baz, Michael Ashely-rollman, and Mustafa Emre Karagozler. This work was supported in part by ANR (ANR-06-ROBO-0009), DARPA (FA87501010215), NSF (CNS-0428738) and Intel Corporation.

References

1. Andersen, D.G., Balakrishnan, H., Kaashoek, M.F., Morris, R.: Resilient Overlay Networks. In: Proc. 18th ACM Symposium on Operating Systems Principles (SOSP), pp. 131–145. Banff, Canada (2001)
2. Ashley-Rollman, M.P., De Rosa, M., Srinivasa, S.S., Pillai, P., Goldstein, S.C., Campbell, J.D.: Declarative programming for modular robots. In: Workshop on Self-Reconfigurable Robots/Systems and Applications at IROS '07 (2007). URL http://www.cs.cmu.edu/~claytronics/papers/ashley-rollman-derosa-iros07wksp.pdf
3. Ashley-Rollman, M.P., Goldstein, S.C., Lee, P., Mowry, T.C., Pillai, P.: Meld: A declarative approach to programming ensembles. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS '07) (2007). URL http://www.cs.cmu.edu/~claytronics/papers/ashley-rollman-iros07.pdf
4. Ashley-Rollman, M.P., Lee, P., Goldstein, S.C., Pillai, P., Campbell, J.D.: A language for large ensembles of independently executing nodes. In: Proceedings of the International Conference on Logic Programming (ICLP '09) (2009)
5. Ashton, K.: That 'internet of things' thing. RFID Journal (2009)
6. Baz, D.E., Boyer, V., Bourgeois, J., Dedu, E., Boutoustous, K.: Distributed part differentiation in a smart surface. Mechatronics (2011)
7. Berlin, A., Gabriel, K.: Distributed mems: New challenges for computation. IEEE Computational Science and Engineering Journal **4**(1), 12–16 (1997)
8. Boutoustous, K., Dedu, E., Bourgeois, J.: An exhaustive comparison framework for distributed shape differentiation in a MEMS sensor actuator array. In: International Symposium on Parallel and Distributed Computing (ISPDC), pp. 429–433. IEEE computer society press, Krakow, Poland (2008)
9. Boutoustous, K., Laurent, G.J., Dedu, E., Matignon, L., Bourgeois, J., Fort-Piat, N.L.: Distributed control architecture for smart surfaces. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2018–2024. IEEE, Taipei, Taiwan (2010)
10. Chapuis, Y.A., Zhou, L., Casner, D., Ai, H., Hervé and, Y.: FPGA-in-the-loop for control emulation of distributed MEMS simulation using VHDL-AMS. In: Proc. of the 1st Workshop on hardware and software implementation and control of distributed MEMS (dMEMS2010), pp. 92 –99. IEEE CPS (2010)
11. De Rosa, M., Goldstein, S.C., Lee, P., Campbell, J.D., Pillai, P.: Programming modular robots with locally distributed predicates. In: Proceedings of the IEEE International Conference on Robotics and Automation ICRA '08 (2008)
12. De Rosa, M., Goldstein, S.C., Lee, P., Campbell, J.D., Pillai, P.: Programming modular robots with locally distributed predicates. In: Proceedings of the IEEE International Conference on Robotics and Automation ICRA '08 (2008). URL http://www.cs.cmu.edu/~claytronics/papers/derosa-icra08.pdf

13. Dedu, E., Bourgeois, J., Boutoustous, K.: Simulation to help calibration of a mems sensor network. International Journal of Pervasive Computing and Communications **6**(4) (2010)
14. Derbakova, A., Correll, N., Rus, D.: Decentralized self-repair to maintain connectivity and coverage in networked multi-robot systems. In: Proc. of IEEE International Conference on Robotics and Automation (ICRA) (2011)
15. Ghosh, S., Bayoumi, M.: On integrated cmos-mems system-on-chip. In: IEEE-NEWCAS Conference, 2005. The 3rd International, pp. 31 – 34 (2005)
16. Giorgetti, A., Hammad, A., Tatibouët, B.: Using SysML for smart surface modeling. In: dMEMS'10, 1st workshop on design, control and software implementation for distributed MEMS, pp. 100–107. IEEE Computer Society Press, Besançon, France (2010)
17. Goldstein, S.C., Mowry, T.C., Campbell, J.D., Ashley-Rollman, M.P., De Rosa, M., Funiak, S., Hoburg, J.F., Karagozler, M.E., Kirby, B., Lee, P., Pillai, P., Reid, J.R., Stancil, D.D., Weller, M.P.: Beyond audio and video: Using claytronics to enable pario. AI Magazine **30**(2) (2009)
18. Gro, R., Bonani, M., Mondada, F., Dorigo, M.: Autonomous self-assembly in swarm-bots. IEEE Transactions on Robotics **22**(6), 1115–1130 (2006)
19. Hornbeck, L.J.: Digital Light Processing for high-brightness high-resolution applications. In: M. H. Wu (ed.) Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 3013, pp. 27–40 (1997)
20. Karagozler, M.E., Thaker, A., Goldstein, S.C., Ricketts, D.S.: Electrostatic actuation and control of micro robots using a post-processed high-voltage soi cmos chip. In: IEEE International Symposium on Circuits and Systems (ISCAS) (2011)
21. L. Matignon G. J. Laurent, N.L.F.P., Chapuis, Y.A.: Designing decentralized controllers for distributed-air-jet mems-based micromanipulators by reinforcement learning. Journal of Intelligent and Robotic Systems **145**(2), 59–80 (2010)
22. Lee, E.A.: Cyber physical systems: Design challenges. Object-Oriented Real-Time Distributed Computing, IEEE International Symposium on **0**, 363–369 (2008)
23. Lewis, A., Bekey, G.: The behavioral self-organization of nanorobots using local rules. In: In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (1992)
24. Reid, J.R., Vasilyev, V.S., Webster, R.T.: Building micro-robots: A path to sub-mm^3 autonomous systems. In: Proceedings of Nanotech 2008 (2008)
25. Rister, B.D., Campbell, J., Pillai, P., Mowry, T.C.: Integrated debugging of large modular robot ensembles. In: ICRA, pp. 2227–2234 (2007)
26. Royer, E., Toh, C.K.: A review of current routing protocols for ad hoc mobile wireless networks. Personal Communications, IEEE **6**(2), 46 –55 (1999)
27. Schwager, M., Slotine, J.J., Rus, D.: Decentralized, adaptive control for coverage with networked robots. In: Robotics and Automation, 2007 IEEE International Conference on, pp. 3289 –3294 (2007)
28. Vasilyev, V.S., Reid, J.R., Webster, R.T.: Microfabrication of si/sio2-spherical shells as a path to sub-mm^3 autonomous robotic systems. In: MRS Fall Meeting (2008)
29. Wang, Y.: Topology control for wireless sensor networks. In: Y. Li, M.T. Thai, W. Wu (eds.) Wireless Sensor Networks and Applications, Signals and Communication Technology, pp. 113–147. Springer US (2008)
30. Witvrouw, A.: Cmos-mems integration: why, how and what? In: Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design, ICCAD '06, pp. 826–827. ACM (2006)