

On Distributed Embedded Systems

Arvindra Sehmi

Bio-medical engineering, Leicester University, United Kingdom

Abstract

Thinking of distributed embedded systems (DES)—let alone the more general area of embedded computing—as a unified topic is difficult. Nevertheless, it is a vastly important topic and potentially represents a revolution in information technology (IT). DES is driven by the increasing capabilities and ever-declining costs of computing and communications devices, resulting in networked systems of embedded computers whose functional components are nearly invisible to end users. Systems have the potential to alter radically the way in which people interact with their environment by linking a range of devices and sensors that will allow information to be collected, shared, and processed in unprecedented ways.

Keywords: *distributed embedded systems*

1. Introduction

DES conjures up images of pervasive collaboration among connected devices, such as tiny, stand-alone, embedded microcontrollers, networking devices, embedded PCs, robotics systems, computer peripherals, wireless data systems, sensors, and signal processors. Combining information from in-place sensors with information from other sources on the network will enable new, rich scenarios to be realized in automotive and avionics control systems, local environmental monitoring for precision agriculture, medical systems, personal health-care monitoring, and manufacturing and process automation, for example.

Widespread use of DES throughout society could dwarf previous revolutions in IT for two reasons. Firstly, Moore's law is primarily driving device miniaturization

and reduced power consumption (instead of increased speed). Secondly, the industry is fast becoming better equipped with development tool chains (software included) and pervasive standards-based communications protocols. As a consequence, embedded computing and communications technology can quite literally be everywhere and extend into all aspects of life—an invisible component of almost everything in everyone's surroundings.

2. Contrasting DES with Traditional Distributed Systems

DES differs from traditional distributed systems in important ways. Embedded systems will eventually interconnect millions, maybe billions, of nodes. This will require changes in the way nodes interact with one another. Broadcast, not peer-to-peer, communications will be the norm. The sheer size of connected clusters of applications will necessitate the use of statistically correct (instead of deterministic) algorithms for resource accounting, fault detection and correction, and system management. These clusters will merge and dissolve rapidly to host functionality that is of interest to groups formed specifically for that purpose. Doing this successfully requires new approaches to naming, routing, security, privacy, resource management, and synchronization. Heterogeneity will factor greatly in the design, implementation, and operation of DES, as will cross-cutting concerns such as dependability, energy-aware computing, critical systems engineering, security, real-time programming, and resource-bounded computing. Some of these factors are familiar to traditional application developers, but many are simply viewed as esoteric boundary issues.

DES tends to be tightly coupled to the physical world and will often be invisible when things are working properly. In contrast to desktop computers, which are part of the office or home furniture, DES instead will be integrated into the furniture and other objects in the environment. You will interact with these objects and devices, but will be unlikely to think of them, as you do when interacting with a computer.

DES components are highly resource constrained. They are typically small, wirelessly connected, bandwidth limited, and operating under physical constraints such as limited energy and the need for adequate heat dissipation. Since they would be integrated in buildings, bridges, vehicles, and so on, they would be expected to last as long as the objects in which they are embedded. The expectation of longevity needs to be taken into account when designing, deploying, and managing these systems. Heterogeneity will be the norm because of the large number of interacting elements that make up DES, so interoperability is a key concern. Managing these constraints and creating a system that functions properly while remaining understandable and manageable by human operators, users, and casual passersby, is a great challenge for DES designers—arguably much more of a challenge than that posed by traditional distributed-systems design. The DES scenarios described later in this paper should clarify these points.

3. Market Opportunity

DES is a fast-growth area of the computing industry and is indicative of the long-term trend of moving away from centralized, high cost, low volume products toward distributed, low-cost, high-volume products.

The next step in this process is the emergence of massively distributed systems—that is, distributed embedded systems that are connected to enterprise systems and the Internet (or the so-called cloud). These systems will penetrate even more deeply into the fabric of society and become the information power grids of the 21st century. As noted, they will be ubiquitous, most will operate outside the normal cognizance of the people they serve, and most will be based on embedded systems that present nontraditional computing interfaces to their users. Their full potential will see them engineered to operate as distributed utilities, much like the energy, water, transportation, and media

broadcast businesses do today [1]. The major industry sectors where DES is likely to be used are automotive, avionics/aerospace, industrial automation (and robotics); telecommunications; consumer electronics and intelligent homes; health and medical systems. The market opportunity of these sectors is significant and the reader is referred to the FAST report [2] for more details. In summary, though, and as Table 1 shows, the overall value of the embedded sector worldwide is about 1600 billion € per year; the three largest markets for embedded systems are telecommunications, automotive, and avionics/aerospace with combined value worldwide of 1,240 billion € per year; and these three largest markets are characterized by growth rates in excess of 10 percent.

Table 1. Estimated total value, through 2006, and growth, through 2010, of major industry sectors using embedded systems (based on the FAST report2 and others)

ES = embedded systems; EC = electronic components

Industry sector	Annual global value	ES value (%)	ES value	EC growth (%)	ES growth (%)
Automotive	800 b€	40%	320 b€	10%	10%
Avionics/Aerospace	750 b€	50%	370 b€	5%	14%
Industrial automation	200 b€	55%	110 b€	5%	7%
Telecommunications	1000 b€	55%	550 b€	9%	15%
Consumer electronics and intelligent homes	300 b€	60%	180 b€	8%	15%
Health and medical systems	130 b€	40%	50 b€	?	18%
Total	3180 b€		1580 b€		

The nature and characteristics of distributed embedded systems are explored below through some example scenarios drawn from the industry sectors mentioned above. This will help draw similarities to traditional distributed-systems development while also recognizing that there are unique systems-design implications, implementation issues, and approaches to solution development.

4. Scenarios

4.1 Automotive Telematics Scenario

In the automotive and avionics industry, embedded systems provide the capability of reaching new levels of safety and sustainability that otherwise would not be feasible, while adding functionality, improving comfort,

and increasing efficiency. Examples of this include improved manufacturing techniques, driver-assistance systems in cars that help prevent accidents, and advanced power-train management concepts that reduce fuel consumption and emissions.

In Western Europe, the “100 percent safe” car is envisioned. It will have sensors, actuators, and smart embedded software, ensuring that neither the driver nor the vehicle is the cause of any accident. This concept extends to all aspects of the driving experience: in-car entertainment, information services, and car-to-car and car-to-infrastructure communication.

For example, the car would know who is allowed to drive it and who is driving it; where it is; where it is going and the best route to its destination; and it would be able to exchange information with vehicles around it and with the highway. It would monitor its own state (fuel levels, tires, oil pressure, passenger compartment temperature and humidity, component malfunction, need for maintenance) and the state of the driver (fatigue, intoxication, anger). The car would first advise and then override the driver in safety-critical situations, use intelligent systems to minimize fuel consumption and emissions, and contain an advanced on-board entertainment system.

Radio-frequency identification (RFID) smart tags within the major car components would communicate with other components during manufacture to optimize the process, communicate with in-car systems during the car’s working life to optimize maintenance cycles, and enable environmentally friendly disposal of the car and its components at the end of its life.

4.1.1 Challenges and Issues

To enable this scenario, components would need to be embedded in long-lived physical structures (such as bridges, traffic lights, individual cars, and perhaps even the paint on the roads). Some components will be permanently connected to a network, but many would be resource constrained (for example, in terms of power) while computing data and thus communicating it wirelessly only when necessary. The many pieces of such a system will of necessity be heterogeneous, not only in form but also in function. There may be subsystems that communicate to consumers in private vehicles, others that relay information from emergency vehicles to synchronize traffic lights, still others that provide traffic data and analysis to highway engineers, and perhaps some that communicate to law enforcement.

How information will be communicated to those interacting with the system is of great importance in such an environment. Safety is a critical concern, and issues of privacy and security arise as well, along with concerns about reliability.

4.2 Precision Agriculture Scenario

Incorporating DES technology into agriculture is a logical development of the advances in crop management over the last few decades. Despite deep understanding and knowledge on the part of farmers about how to adjust fertilizers, water supplies, and pesticides, and so on, to best manage crops and increase yields, a multitude of variations still exist in soil, land elevation, light exposure, and microclimates that make general solutions less than optimal, especially for highly sensitive crops like wine grapes and citrus fruit.

The latest developments in precision agriculture deploy fine-grained sensing and automated actuation to keep water, fertilizer, and pesticides to a minimum for a particular local area, resulting in better yields, lower costs, and less pollution-causing runoff and emissions. Furthermore, the data collected can be analyzed and incorporated as feedback control to adjust irrigation flow rate and duration tuned to local soil conditions and temperature. Sensors that can monitor the crop itself (for example, sugar levels in grapes) to provide location-specific data could prove very effective.

In the future, DES might be used to deploy sensors for the early detection of bacterial development in crops or viral contamination in livestock, or monitor flows of contaminants from neighboring areas and send alerts when necessary. In livestock management, feed and vitamins for individual animals will be adjusted by analyzing data from networks of ingestible sensors that monitor amounts of food eaten, activity and exercise, and health information about individual animals and the state of the herd as a whole.

4.2.1 Challenges and Issues

In this scenario, embedded components must be adaptive, multimodal, and able to learn over time. They will need to work under a wide range of unpredictable environmental conditions, as well as to interact with fixed and mobile infrastructure and new elements of the system as they are added and removed at varying rates of change.

4.3 Aviation and Avionics Scenario

The European Commission has set goals for the aviation industry of reducing fuel consumption by 30 percent by 2021 through the use of embedded systems. This may be a high goal to be achieved solely through the use of technology. But in this industry, the goals appear to very ambitious across the board.

The unmanned aerial vehicle (UAV) for use in surveillance and in hazardous situations such as firefighting promises to be cheaper, safer, and more energy efficient to operate than conventional aircraft. There are apparently many different kinds of UAVs under development: some with long-duration operational cycles and extensive sensor suites; some with military defense and attack capability; others that are small enough to be carried and deployed by individuals; and, in the future, tiny, insect-like, UAVs providing a flying sensor network. The aircraft of the future will have advanced networks for on-board communication, mission control, and distributed coordination between aircraft. These networks will support advanced diagnosis, predictive maintenance, and in-flight communications for passengers. For external communication, future aircraft will communicate with each other in spontaneous, specific-for the-purpose ways similar to peer-to-peer networks.

4.3.1 Challenges and Issues

The aviation and avionics industry has specific needs in terms of security, dependability, fault tolerance and timeliness, stretching the limits of distributed embedded-systems design and implementation. The whole system, if not each of its embedded components, needs to be high precision, predictable, and robust for 100 percent operational availability and reliability. It must enable high bandwidth, secure, seamless connectivity of the aircraft with its in-flight and on-ground environment. It should support advanced diagnosis and predictive maintenance to ensure a 20- to 30-year operational life span.

DES design environments and tools will need to provide significant improvements in product development cycles, ongoing customizations, and upgrades beyond those achievable with current distributed-systems development tools. Design advances in fast prototyping, constructive system composition, and verification and validation strategies will be required to manage this complexity.

4.4 Manufacturing and Process-Automation Scenario

Embedded systems are important to manufacturing in terms of safety, efficiency, and productivity. They will precisely control process parameters, thus reducing the total cost of manufacture. Potential benefits from integrating embedded control and monitoring systems into the production line include: better product quality and less waste through close process control and real-time quality assurance; more flexible, quickly configured production lines as a result of programmable subsystems; system health monitoring, which leads to more-effective, preventive and lower-cost maintenance; safer working environments due to better monitoring and control; and better component assembly techniques, such as through the use of smart RFID tags.

4.4.1 Challenges and Issues

There are many implications of this industry scenario for DES. One is a need for better man-machine interactions in what is fundamentally a real-time, man-plus-machine control loop. Providing better interactions will improve quality and productivity by ensuring that there are no operator errors, as well as by reducing accidents.

Availability, reliability, and continuous quality of service are essential requirements for industrial systems achieved through advanced control, redundancy, intelligent alarming, self-diagnosis, and repair. Other important issues are the need for robustness and testing, coherent system-design methodology, finding a balance between openness and security, integrating old and new hardware with heterogeneous systems, and managing obsolescence.

4.5 Medical and Health-Care Scenario

Society is facing the challenge of delivering good-quality, cost-effective health care to all citizens. Medical care for an aging population, the cost of managing chronic diseases, and the increasing demand for best-quality health care are major factors in explaining why health-care expenditures in Europe are already significant (8.5 percent of GDP) and rising faster than overall economic growth. Medical diagnosis and treatment systems already rely heavily on advances in embedded systems. New solutions that mix embedded intelligence and body-sensing techniques are currently being developed [3], and current

advances in this area address patient-care issues such as biomedical imaging, remote monitoring, automatic drug dispensing, and automated support for diagnosis and surgical intervention.

4.5.1 Challenges and Issues

The medical domain represents a complex and diverse arena for extraordinary developments that deploy a wide range of embedded systems. Implanted devices such as pacemakers and drug dispensers are commonplace, but need to become more sophisticated, miniaturized, and connected to networks of information systems. Wearable devices for monitoring and managing cholesterol, blood sugar, blood pressure, and heart rate must be remotely connected to the laboratory and to the operating room in a secure and reliable manner from beginning to end. Robotic devices are being used today to guide and perform invasive surgery requiring high-integrity engineering practices not even imagined in a typical “mission-critical” enterprise application.

4.6 Mobility Scenario

Combining mobile communications with mobile computing is allowing people to talk to others and access information and entertainment anywhere at any time. This requires ubiquitous, secure, instant, wireless connectivity, convergence of functions, global and short-range sensor networks and light, convenient, high-functionality terminals with sophisticated energy management techniques. Such environments will enable new forms of working with increased productivity by making information instantly available, when needed in the home, cars, trains, airplanes and wider-area networks. Imagine a hand-held or wearable device giving easy access to a range of services able to connect via a range of technologies including GSM, GPS, wireless, Bluetooth and via direct connection to a range of fixed infrastructure terminals.

Potential applications and services include: Entertainment, education, internet, local information, payments, telephony, news alerts, VPNs, interfaces to medical sensors and medical services, travel passes and many more.

4.6.1 Challenges and Issues

Devices would need to reconfigure themselves autonomously depending on patterns of use and the available supporting capabilities in environment or infrastructure and be able to download new services as they became available. To develop such infrastructure, the gap between large, enterprise systems and embedded components would need to be bridged. Significant developments are required in technology for low-power and high performance computing, networked operating systems, development and programming environments, energy management, networking and security.

Issues that need to be resolved in the infrastructure to support these kinds of scenarios include the provision of end-to-end ubiquitous, interoperable, secure, instant, wireless connectivity to services. Simultaneously the infrastructure must allow unhindered convergence of functions and of sensor networks. Addressing the constraints imposed by power management (energy storage, utilization and generation) at the level of the infrastructure and mobile device poses a major challenge.

4.7 Home-Automation and Smart Personal Spaces Scenario

By deploying DES in the home, an autonomous, integrated, home environment that is highly customizable to the requirements of individuals can be foreseen. Typical applications and services available today include intruder detection, security, and environmental control. But in the future, applications and services to support the young, elderly, and infirm will be developed, and these may have the ability to recognize individuals and adapt to their evolving requirements, thereby enhancing their safety, security, and comfort. By tying in with applications and services described in the medical/health-care and mobility scenarios, smart personal spaces could be developed.

4.7.1 Challenges and Issues

Multidisciplinary, multi-objective design techniques that offer appropriate price and performance, power consumption, and control will have to be used if we are to realize the potential of embedded systems for home entertainment, monitoring, energy efficiency, security, and control. Such systems will require significant computational, communication, and data-storage

capabilities. The mix of physical monitoring and data-based decision support by some form of distributed intelligence will rely on the existence of seamlessly connected embedded systems and the integration of sensors and actuators into intelligent environments. These systems will be characterized by ubiquitous sensors and actuators and a high-bandwidth connection to the rest of the world. Technologies will need to be developed that support sensing, tracking, ergonomics, ease-of-use, security, comfort, and multimodal interaction.

Key to achieving this result will be developing wireless and wired communications and techniques for managing sensor information, including data fusion and sensor overloading. The challenges are to make such systems intelligent, trustworthy, self-installing, self-maintaining, self-repairing, and affordable, and to manage the complexity of system behavior in the context of a large number of interoperable, connected, heterogeneous devices. These systems will need to operate for years without service, be able to recover from failure, and be able to supervise themselves.

Managing these embedded systems will require support of all aspects of the life cycle of the application and service infrastructures, including ownership, long-term storage, logging of system data, maintenance, alarms, and actions by the provider (emergency, medical, or security) services, authorization of access and usage, and charging and billing under a range of different conditions of use.

5. Technical Imperatives

Some of the most challenging problems facing the embedded-systems community are those associated with producing software for real-time and embedded systems. Such systems have historically been targeted to relatively small-scale and stand-alone systems, but the trend is toward significantly increased functionality, complexity, and scalability, as real-time embedded systems are increasingly being connected via wireless and wired networks to create large-scale, distributed, real-time, and embedded systems. These combined factors [4] require major technical imperatives to be addressed by both industry and research establishments: (a) self-configuration and adaptive coordination; (b) trustworthiness; (c) computational models; and (d) enabling technologies. Let's now look at each technical imperative in more detail.

6. Self-Configuration and Adaptive Coordination

Self-configuration is the process of interconnecting available elements into an ensemble that will perform the required functions at the desired performance level. Self-configuration in existing systems is realized through the concepts of service discovery, interfaces, and interoperability. But embedded systems, which appear in hybrid environments of mobile and static networks with nodes of varying capability, energy availability, and quality of connectivity, are plagued by diverse and energy-limited wireless connectivity—making low power discovery a challenge. Also scalable discovery protocols, security, and the development of adequate failure models for automatically configured networks will require that solutions be developed.

Adaptive coordination involves changes in the behavior of a system as it responds to changes in the environment or system resources. Coordination will not necessarily be mediated by humans because DES could be so large and the time scale over which adaptation needs to take place too short for humans to intervene effectively. Achieving adaptive coordination in DES will draw on the lessons learned from adaptive coordination in existing distributed systems, but it will also require meeting the radical new challenges posed by the physically embedded nature of collaborative control tasks and large numbers of nodes, and further complicated by the relatively constrained capabilities of individual elements. Thus, to achieve adaptability in DES, solutions are needed in decentralized control and collaborative processing, and techniques must be developed to exploit massive redundancy to achieve system robustness and longevity.

7. Trustworthiness

If we can expect DES to be deployed in large numbers and become an essential part of the fabric of everyday life, five technical imperatives must be taken into account in their design from the outset: reliability, safety, security, privacy, and usability.

On reliability, current monitoring and performance-checking facilities, and verification techniques are not easily applicable to DES because of their large number of elements, highly distributed nature, and environmental dynamics.

In terms of safety or the ability to operate without causing accidents or loss, bounded rational behaviors are essential, especially in the face of real-time systems and massive DES likely to exhibit emergent or unintended behaviors.

It may be virtually impossible to distinguish physical from system boundaries in a large-scale DES, making security a big challenge. And, considering that networking of embedded devices will greatly increase the number of possible failure points, security analysis may prove even more difficult.

Privacy and confidentiality policies will be exacerbated by the pervasiveness and interconnectedness of DES. Users will be monitored, and vast amounts of personal information will be collected. Thus, implementing privacy policies, such as acquiring consent in a meaningful fashion in large scale networks, will be very difficult.

Related to all of the above, embedded systems must be usable by individuals who have little or no formal training. Unfortunately, usability and safety often conflict, so trade-offs will need to be made. Understanding the mental models people use of the systems with which they interact is a good way for designers to start addressing issues of usability and manageability of embedded systems.

8. Computational Models

New models of computation are needed to describe, understand, construct, and reason about DES effectively. Understanding how large aggregates of nodes can be programmed to carry out their tasks in a distributed and adaptive manner is a critical research area. Current distributed-computing models, such as distributed objects and distributed shared memory, do not fully address all of the new requirements of DES. Furthermore, DES's tight coupling to the physical world, the heterogeneity of their systems, the multitude of elements, plus timing and resource constraints, among other things, demonstrate the need for a much richer computing model. Such a model will need to incorporate resource constraints, failures (individual components may fail by shutting down to conserve energy, for example), new data models, trust, concurrency, and location.

For example, in a sensor network, subsystems from different vendors should interoperate easily and be integrated seamlessly into the rest of the IT infrastructure, providing intuitive interfaces for remote and novice users. Multiple, concurrent clients will be exercising different functionalities of the system for different purposes, and

although resources may not be the most stringent constraint for the system, it has to be self-monitoring and aware of its resources. It must have a certain level of autonomy to decide on the best use of available resources to fulfill multiple users' concurrent and uncoordinated requests. The complexity of the computational model may be further exacerbated by the multiple ways to obtain a piece of information. So, the model would have to capture information equivalency relations over the rich variety of sensor data, which in turn would require semantic information and services to process and reason about the sensor data in ways that move beyond protocol agreement and data-format conversion. To quantify the semantic information contained in sensor data and to capture the relations between various semantic entities such as objects and events in the physical world, the model would need to define an ontology for a variety of commonly encountered sensor data and provide a set of semantic transformation services to incrementally extract new semantic information from lower level data. (This topic is of particular interest to the author and the reader is referred to [5], [6], [7], and [8].)

9. Enabling Technologies

The evolution of more and more sophisticated DES and complex embedded-systems solutions is fueled by the revolutionary advances in information technology during the past several decades. Silicon scaling is the main driving force, with exponentially increasing processor performance enabling a world in which sophisticated chips can be manufactured and embedded easily and cheaply. Continued improvements in the price and performance of chip technology are expected for at least the next decade. Still, new developments are needed in specific aspects of communications, geo-location, software, and operating systems.

As wireless (and wired) technologies continue to become less expensive and more sophisticated, the vision of pervasive, connecting, embedded processors becomes increasingly feasible. However, most of the progress to date in wireless technology has focused on medium- to long-range communications (as in cellular phones and pagers) and is not sufficient for the widespread deployment of DES. Work is needed to understand how to create network architectures and designs for low-power, short-range wireless systems. Related to wireless are the issues surrounding geo-location technology. Unlike

conventional computer networks, which are more dependent on the relative positioning of elements in a network topology, DES are often inextricably tied to the physical world (a primary purpose often being to measure and control physical-world attributes or objects), so location in physical space is more important. DES will therefore require ready access to absolute or relative geographic information, especially for sensor networks responsible for physical sensing and actuation.

Attention must also be given to the responsible and effective integration of sensor devices into DES, which will have impacts on the design of network-aware embedded operating systems, application development, and software tooling that has the characteristics required to tailor solutions to the physical constraints, afford in-place deployment, be upgradable, have high availability, and the ability to migrate to new hardware. Since DES will be embedded in long-lived structures, they will have to evolve, depending on changing external conditions and advances in technology as time passes. Thus, operating systems and applications that can cope with this type of evolution will be critical. Furthermore, real-time operation and performance-critical constraints on multiple embedded applications running concurrently on the same device will create a demand for entirely new methods of software development and systems architectures.

Figure 1 is a chart of the technical imperatives for DES development and research discussed in the previous paragraphs. Additional issues present in this field are given in summary form and serve to highlight the significant differences from traditional distributed-systems development.

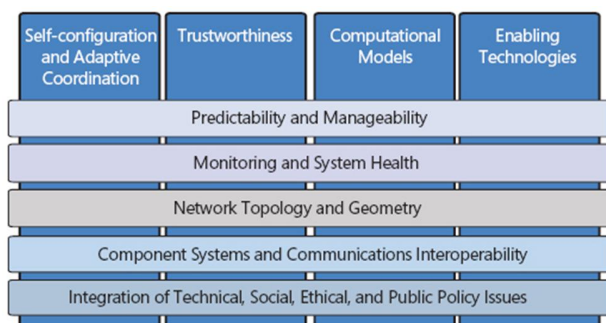


Figure 1. Technical imperatives for distributed embedded-systems development and research

9.1 Further Issues and Needs

1. **Predictability and manageability**—Covers methodologies and mechanisms for designing predictable, safe, reliable, manageable distributed embedded systems.
2. **Monitoring and system health**—Refers to a complete conceptual framework to help achieve robust operation through self-monitoring, continuous self-testing, and reporting of system health in the face of extreme constraints on nodes and elements of the system.
3. **Network topology and geometry**—Represent modeling techniques to support and incorporate network geometry (as opposed to just network topology) into DES.
4. **Component systems and communications interoperability**—Is about the techniques and design methods for constructing long-lived, heterogeneous systems that evolve over time and space while remaining interoperable.
5. **Integration of technical, social, ethical, and public-policy issues**—Is predominantly about the fundamental research required to address nontechnical issues of embedded systems, especially those having to do with the ethical and public-policy issues surrounding privacy, security, reliability, usability, and safety.

10. Development

To understand the nature of embedded-systems development complexity, you should also understand the typical embedded-device value chain. This consists of links between various industry players that comprise silicon vendors (SVs), embedded OS vendors, independent hardware vendors (IHVs), device makers (original equipment developer (OED) and original equipment or design manufacturer (OEM/ODM)), distributors, and end customers (businesses or consumers).

In Figure 2, you see that the embedded-device value chain consists of three to seven links between SVs, OEDs, OEMs, and end customers. Depending on the device maker's supply chain and market, it may also consist of third-party software providers, IHVs, ODMs, and various distribution intermediaries.

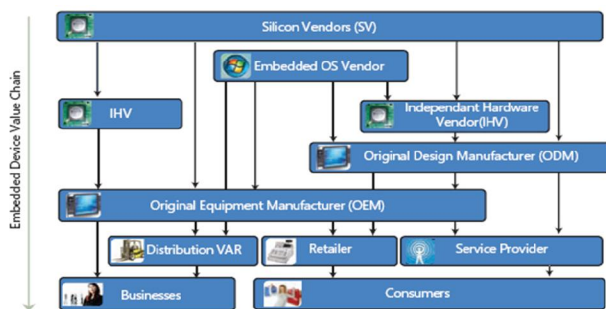


Figure 2. Embedded-device value chain

When developing traditional business applications and systems, “devices” are standardized and readily available off-the-shelf computers, making a significant part of their value chain quite irrelevant to the development process. However, in embedded-systems design and development, a similar level of maturity or commoditization, if you like, of “devices” has not yet been achieved in the industry. As a result, development tool chains, processes, and methodologies are often proprietary and established for a specific goal. The vast majority of the industry in embedded development is using open-source software and custom development tools provided by the hardware vendors (SV, IHV) and open source communities. Time scales for the development of hardware and software at all levels in the device value chain can therefore be long as it requires proprietary or scarce skills and knowledge to put solutions together. In some cases, where DES and real-time scenarios are involved, especially in safety-critical situations, these time scales can easily span a decade. There are interesting offerings from the large operating systems and software tools vendors such as Microsoft that hold the promise of providing a more productive and “traditional” development experience for embedded-systems professionals (including academics and hobbyists) in many key solution scenarios. The hope is that eventually tool chains along the entire device value chain will become mostly standardized and interoperable enabling the embedded-systems industry to scale development as seen in traditional software development. Figure 3 shows some aspects of the embedded-systems development life cycle—which is strongly tied to the device value chain—and the Microsoft platform technologies that may be applied through the life cycle.

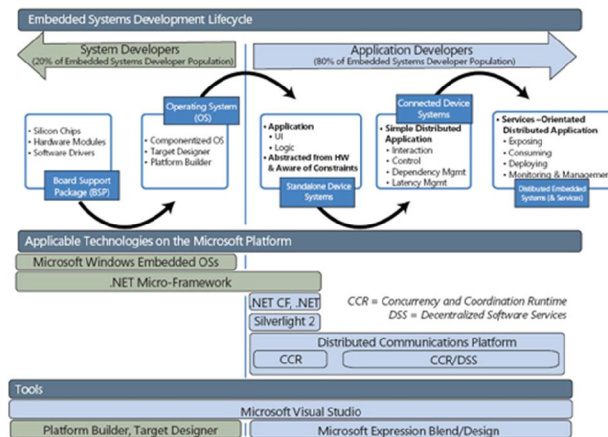


Figure 3. Embedded-systems development life cycle and Microsoft technologies

Because embedded devices contain both hardware and software components understanding the interactions between them are essential. Key skills in this area span electrical engineering and computer science; they also address the interrelationships among processor architectures, performance optimization, operating systems, virtual memory, concurrency, task scheduling, and synchronization. Developers should have a breadth of exposure to appreciate the utility of other fields in embedded systems, such as digital signal processing and feedback control. Finally, as embedded systems are not general operating systems, developers should understand the tight coupling between embedded applications and the hardware platforms that host them.

11. Conclusion

This paper introduces you to the rich field of distributed embedded systems. It is geared towards architects and developers of traditional distributed systems and therefore takes a broad perspective on the task of describing DES. Several references at the end may assist readers in exploring this subject in greater depth. Distributed systems design approaches and their characteristics will influence DES design and implementation, but there are significant differences that have been described in this paper. The way we conceive and program these systems is bound to become easier. It has to, because smart, connected, service-oriented embedded systems will permeate our factories, hospitals, transportation systems, entertainment, and personal spaces, and thus become essential to our

daily lives. In many ways, DES is the computing model for the future, and the mainstream of the IT industry will need to learn how to design, build, and operate these systems well.

References

- [1] Dan Nasset. Massively Distributed Systems: Design Issues and Challenges. USENIX Workshop on Embedded System, 1999.
- [2] FAST Report. Study of Worldwide Trends and R&D Programmes in Embedded Systems. 2005.
- [3] Siegemund F., Haroon M., Ansari J., Mahonen P., Senslets. Applets for the Sensor Internet. Wireless Communications and Networking Conference, 2008.
- [4] National Research Council, Committee on Networked Systems of Embedded Computers. Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers. National Academy Press, Washington, D.C., 2001.
- [5] Jie Liu and Feng Zhao. "Towards Semantic Services for Sensor-Rich Information Systems." 2nd International Conference on Broadband Networks, 2005.
- [6] Jie Liu, Eker, J., Janneck, J.W., Xiaojun Liu, Lee, E.A. "Actor-Oriented Control System Design: A Responsible Framework Perspective." IEEE Transactions on Control Systems Technology, 12(2), March 2004.
- [7] Schmidt D. C., Gokhale A., Schantz R. E., and Loyall, J. P. Middleware R&D Challenges for Distributed Real-Time and Embedded Systems. ACM SIGBED Review, 1 (1), 2004.
- [8] Gul A. Agha, Ian A. Mason, Scott F. Smith, and Carolyn I. Talcott. A Foundation for Actor Computation. J. Functional Programming, 7 (1), 1997.

Arvindra Sehmi holds a Ph.D. in bio-medical engineering from the University of Leicester, an MBA from the Open Business School, and a Bachelor of Science degree from the University of Sussex.