A vision for computer science – the system perspective

Joseph Sifakis

Verimag Laboratory, Centre Equation,
2 ave de Vignate, 38610 Gieres, France

Received 18 Feb 2011; accepted 03 Mar 2011

1. The evolution of computer science

Computer Science is a young discipline. Its foundations were laid in 1936 by the seminal work of A.M. Turing and K. Gödel. Its scope and focus have changed continuously over the past seven decades. The first computers ran numerical software for defense applications. In the 70’s, the advent of mainframe computers broadened the scope to include commercial applications. In parallel, large-scale circuit integration allowed exponential increases in computing power (Moore’s law). In the 80’s, the convergence between information technologies and telecommunications opened the way for the Internet, the Web and the Information Society. In the 90’s, another very important, but less visible, revolution started with the dissemination of embedded systems technologies. More than 95% of the chips produced today are for embedded applications. These are electronic components integrating software and hardware jointly and specifically designed to provide given functionalities, which are often critical. They are hidden in devices, appliances and equipment of any kind: mobile phones, cameras, home appliances, cars, aircraft, trains, medical devices etc. In 2008, the average person used about 230 embedded chips every day: 80 chips in home appliances, 40 chips at work, 70 chips in cars, 40 chips in portable devices.

In the near future, another anticipated, important landmark will be the advent of the Internet of Things as the result of the convergence between embedded technologies and the Internet. The idea is to use internet technologies to integrate services provided by hundreds of billions of embedded systems. This will require an upgrade of the internet infrastructure to make it more secure, safer and reactive. Current features for exchanging multimedia documents will be extended to encompass real-time monitoring and control. Systems are becoming ubiquitous: the state of almost everything can be sensed, measured and monitored; people and objects can communicate and interact in entirely new ways; intelligent systems allow enhanced predictability of events and optimal use of resources.

It is hard to imagine what Computer Science will be in two decades. More than any other discipline, it is driven by applications and exponential progress in technology. The broadening of its perimeter is accompanied by a shift in focus from algorithms and programs to systems.

* E-mail: Joseph.Sifakis@imag.fr
2. From programs to systems

I see two main landmarks in the evolution of CS: the advent of Computers and the development of Theory of Computing. The first allowed the move from mechanical computational devices to much faster and reliable automated electronic circuits. The second opened the way for mechanizing computation by developing models of computation for studying algorithms, programs and their properties. The central issue addressed is whether, and when, functions can be computed by using models of computation. A remarkable fact about these models is that they ignore physical time and resources. Computation is a finite sequence of steps corresponding to the execution of primitive operations. Complexity theory is based on abstract notions of time and memory. Programs and algorithms are just relations independent from the physical resources needed for their execution. Their behavior is terminating, deterministic and platform-independent.

In contrast to programs and algorithms, systems are reactive; that is, they continuously interact with an external environment. Their inputs are stimuli that trigger state changes and computation of outputs that may modify the state of their environment. System behavior can be modelled as a relation between histories of inputs and histories of outputs. Systems are, in general, non-terminating and non-deterministic. Furthermore, their behavior is, in general, platform-dependent as their correctness depends on the dynamic characteristics of the execution platform e.g. execution times. Theory of computation is, by its nature, of little help for studying systems. Even if we perfectly understand the properties of a program and the properties of a hardware target platform, we have no theory to predict the behavior of the program running on the platform. The observed shift of focus in Computer Science from programs to systems should be accompanied by research for extending the Theory of Computing. Models of computation should be enriched to take into account physical resources and the interplay between systems and their physical environment.

3. Trends in system design

Embedded Systems break from traditional computing systems such as desktop computers and servers. They must jointly meet technical requirements such as:

- **Reactivity**: responding within a known and bounded delay. This is essential not only for real-time applications but also for efficient quality of service control.

- **Autonomy**: providing continuous service without human intervention. In particular, this means no manual restart but also optimal power management for portable devices.

- **Dependability**: invulnerability to threats including attacks, hardware failures, software execution errors.

- **Scalability**: performance increase is commensurable with the increase of resources.

In addition to these requirements, embedded systems must meet requirements for optimal cost/quality as they are integrated in mass-market products. Seeking an economic optimum in system design is much harder than achieving high quality without cost-effectiveness constraints.

Embedded technologies challenge our capacity to develop systems of guaranteed functionality and quality at acceptable costs. Today we master, at high costs, two types of systems which are difficult to integrate: 1) safety and/or security critical systems of low complexity such as flight controllers or smart cards; 2) complex best-effort systems such as telecommunication systems and web-based applications. Dependability is the main concern for critical systems while best-effort systems seek optimal use of resources at acceptable levels of quality of service.

For future systems, we urgently need technology for

- **developing affordable critical systems**. In many application areas such as transport, health and energy, embedded technologies could be used to provide new services with significant impacts on quality of life and efficient resource management. For instance, drive-by-wire and brake-by-wire in cars could lead to lower production and operational costs by replacing “passive safety” with “active safety”.

- **safe integration of heterogeneous systems-of-systems**. The vision is to develop global services by integrating features provided by geographically distributed systems with different technical characteristics and using various