Chapter 11
F-DRARE: A Framework for Deterministic Runtime Adaptation of Cyber Physical Systems

Fahad Bin Tariq
University of Paderborn, Germany

Sandeep Korrapati
University of Paderborn, Germany

ABSTRACT
Modern systems are designed to have various non-functional attributes such as fault tolerance, robustness, et cetera. The ability of systems to adapt while being connected to a global network as in Cyber Physical Systems, presents new opportunities. Extra-functional qualities may be extended to nodes in the network that initially were devoid of them. It is inevitable that such vast global networks will consist of computationally constrained units. Extending such computationally constrained nodes with the ability to adapt is already a difficult challenge, which becomes more difficult when added with the constraint of timely behavior. In this chapter, the authors present an approach to tackle the issue of runtime adaptation on computationally constrained systems with real-time constraints. The focus is on systems with only a single general purpose processor, with the responsibility to run the system and application software. The authors introduce their framework towards this aim, followed by the architecture and a case study.

INTRODUCTION
Software is the most flexible entity on machines that employ programmable devices to carry out their tasks. The inherent flexibility in software instructions (due to their earlier form as punch cards and now in the form of voltage levels) translates into a degree of reusability for the machines or platforms on which they execute. By a mere change in software, the same platform can be used to fulfill different objectives. The desktop machines and handheld devices of today running...
countless software on the same platform serve as an example. Further being connected to a global network such as the Internet allows users of these devices to access various software services to either increase their quality of experience or fulfill a timely need. To fully harness the possibilities provided by an inherently flexible entity such as software, changing the software while the system is operational, based on various parameters such as the current environment, new objectives etc. is necessary. The system is then said to have adapted during runtime to the environment or any changes in objectives. Some of the advantages of the ability to adapt during runtime are, the delaying of decisions from design time to runtime relying on more up-to-date context information resulting in higher decision accuracy, as well as the acquisition of new behavior to tackle new problems.

The pioneering work in runtime adaptation (a.k.a. runtime evolution, dynamic updates, online adjustment) in general may be attributed to the work done in Fabry (1976). In the area of embedded systems, the structure and behavior of a system are typically closed without any room for runtime adaptation. This is because safety and timeliness are the dominant qualities that govern system design in this domain. But due to the benefits that accompany the ability of a system to adapt, it was only a matter of time and recent years have shown an increased interest by the embedded systems community in harnessing the benefits associated with it. Runtime adaptation is the ability of a system to change its behavior and/or structure while remaining in a desirable state of operation. A desirable state of operation is a relative term depending on the domain of application. In a time-critical system a desirable state is where all deadlines are met whereas in a non-time-critical system a desirable state may be certain functional and/or QoS objectives.

Many reasons exist for having runtime adaptation as a characteristic property of the system. Along with an increase in system complexity as well as the complexity of the development process, the costs and efforts to maintain them have also increased. A system with the ability to adapt reduces maintenance costs by inherently supporting change during the runtime phase of the system lifecycle. Factors such as computational load and communication requirements may change overtime resulting in different overall resource requirements. Node failures in a distributed environment may require topological restructuring and load balancing such as the sharing of processing load or the creation of new connections. Systems with the ability to adapt enable them to acquire new properties in such scenarios, for example, fault tolerance.

The increasing complexity of embedded systems is due, but not limited, to the increasing quality and quantity of the resources involved. Multi-core processing units, sophisticated storage and communication devices are examples of the underlying platforms that some embedded systems are enjoying. Interestingly such resources are there in the first place due to increasing complexity brought onto the system by ever-increasing demands of more functionality (realized by software). Dividing the load in a distributed environment is one solution commonly practiced, but this results in a further requirement of managing the distributed resources. Having stated the above, increasing the connectivity of systems at various levels does give rise to potential benefits that have attracted a keen interest by both research and industry. New paradigms such as Cyber Physical Systems (Lee, 2008) and the Cyber Biosphere (Rammig, 2008) as well as concepts such as the Internet of Things aim to address these benefits. The Future Internet aims take standalone devices, autonomous systems, consumer electronics, vehicles etc. into the fold of the internet. From an adaptation perspective, being connected to a source of virtually un-limited resources, information and behavior opens up many possibilities. Behavior and qualities may be acquired by systems that were not present initially.