Multi-core/many-core systems offer the potential both for cheap, scalable high-performance computing and also for significant reductions in power consumption. In order to achieve this potential, it is essential to take advantage of new heterogeneous architectures comprising collections of multiple processing elements (CPUs or GPUs). A key technical problem limiting the use of integrated multicore parallel systems is maintaining portable performance across different CPU/GPU combinations. The core of the problem is in providing high-level programming support with a flexible implementation model.

This talk introduces  a new approach (developed in the ParaPhrase project (www.paraphrase-ict.eu)) that aims to address this problem by developing and deploying new high-level design patterns for parallel applications that allow alternative parallel implementations, and that can be initially mapped and subsequently re-mapped to the available hardware. The approach builds on a two-level (or ultimately multi-level) model of parallelism, where the implementations of parallel programs are expressed in terms of interacting components, and where components from different applications are collectively mapped to the available system resources. By expressing parallelism in terms of high-level parallel patterns that have alternative parallel implementations, we aim to redeploy/refactor parallel components to match the available hardware resources.