



# Network of systems

## BUILDS ON PARALLELISM

*As systems emerge and evolve, they are increasingly becoming networks working together in coordination.*

MICROSOFT'S TECHNOLOGIES have strong associations with PCs and servers. The technology behind the Microsoft Robotics Studio provides a clue about how designers will create emerging and future applications for not just PCs, but also robotic systems, embedded autonomous subsystems, online-presentation services, financial systems, and scientific-modeling systems. Tandy Trower, general manager of the Microsoft Robotics Group, and Henrik Frystyk Nielsen, group program manager of the Microsoft Robotics Group, share the underlying concepts of the technology that make development and runtime systems such as the Robotics Studio possible.

The technology borrows from the parallel-computing world because the development team recognizes that the model for designing, building, and operating systems is changing to one that will accommodate multiple data and processing sources that aggregate and coordinate for a coherent system-level behavior. A current example of this concept in action is how Web pages increasingly aggregate material from disparate sources and present them to a reader as a single, coherent set of material. Although the technology's developers did not specifically design it for building robotic systems, Robotics Studio has become a good first test of the underlying technology for building distributed, asynchronous systems. The technology is demonstrating its relevance outside the robotics world, such as with *MySpace* for managing messaging traffic, and other organizations are exploring the technology for financial- and scientific-modeling systems.

The implementation of the technology centers on two runtime components: CCR (concurrency-and-coordination runtime) and DSS (decentralized-software services). CCR supports asynchronous and concurrent operation through a message-oriented-programming model

that can exploit parallel hardware and coordinate messages without the use of manual threading, locks, or semaphores. The primary goal of DSS is to provide interoperability between services regardless of whether these services are running within the same node or across a network.

The technology in the Robotics Studio does not replace how to build the real-time components; hard-real-time components still require development teams to work close to the metal. Rather, the technology targets a higher level of abstraction that focuses on aggregating and coordinating data from a distributed set of concurrently operating subsystems. An analogy could be the relationship in the human neural system between reflexes versus involving the mind. The reflexes are closer to the event and short-circuit the brain from the response; however, the mind can be aware of when a reflex activates and can associate changes in the overall system that allow the aggregate of other subsystems to in the future avoid the circumstances that caused the reflex to activate in the first place.

Rather than focusing on how to accomplish faster processing through parallel processors, the technology

takes the stance that parallelism is not the main challenge of building modern applications. A system can consist of a network of separate entities that provide parallel resources; in such a system, the challenge lies in how to bring the execution threads together so that the system can coordinate and orchestrate all of the subsystems to produce the desired system-level behavior.

The technology also embodies the concept that these types of systems must be able to recognize, handle, and recover from partial faults. The system as a whole must be able to continue to perform even when a subsystem must perform a reset. This concept is not new for safety-critical systems, but, as more systems become the aggregate of parallel subsystems operating together, this concept becomes more relevant to a wider range of applications.

A forward-looking concept in this technology is that autonomous components can be aware of each other. This idea is not far-fetched when you consider the growing number of autonomous safety systems in an automobile. These safety systems are responsible for handling some set of conditions, and the entire automobile needs to ensure that no system invalidates the operation of the others. For energy management within a home, connected appliances could work together to minimize usage spikes. Additionally, as users add systems to the network of systems, they should work together on a larger scale, and the new systems must learn from the older components how the user has customized settings and expects the overall system to behave.

The current incarnation of the Robotics Studio may not be how developers

*See Microsoft* » 36

---

«22 *Microsoft*  
will in the future design applications,  
but it appears to be on the road to ad-  
dressing how the way designers build  
applications is changing to accom-  
modate multiple processing sources  
that may or may not reside in the same  
chip, board, chassis, building, or even  
land mass. Consider that most of the  
human nervous system, one of the most  
complex computing and control systems

that we are aware of, operates at tens  
to hundreds hertz—far slower than the  
gigahertz-speed processors that are hav-  
ing trouble keeping up with the growing  
computational demands of tomorrow’s  
applications. The magic is not in the  
parallel processing; it is in the aggrega-  
tion and coordination of a network of  
processing elements.

—*Robert Cravotta*