Meeting the Demands of Changing Operating Conditions at Runtime Through Adaptive Programming Techniques for Network Embedded Computing

Richard E. Schantz and Joseph P. Loyall
BBN Technologies, Cambridge, Massachusetts

Complex computing systems are predominantly networked and their reach extends beyond any individual platform. As a result, they must operate under increasingly unpredictable and changeable conditions as viewed from many different perspectives simultaneously. As the conditions change, so too may the requirements for appropriate behavior under those conditions, either to continue to meet operational requirements, to maximize effective resource utilization, or to react to changing requirements of the new conditions. Building these sorts of flexible software systems is a major challenge. The types of systems with these characteristics are simultaneously distributed, real-time and embedded (DRE) requiring coordinated solutions to the challenges brought by each of these attributes. They require predictable mission-critical levels of service coordinated end-to-end under changing conditions, often within shared, constrained resource environments. Adaptive distributed object computing (DOC) middleware is a promising approach to provide these end-to-end services by coordinating lower-level mechanisms and integrating them with application- or user-centric tradeoffs to change behavior based on current operating conditions.

The Quality Objects (QuO) middleware is a set of layered architecture extensions to standard DOC middleware, that is used to control and adapt quality of service attributes in a number of distributed application environments. The QuO architecture decouples middleware and applications along the following two dimensions:

- *Functional paths,* which are flows of information and control between client and remote server applications (or more generally among the networked elements).
- *QoS attribute paths,* which are responsible for determining how well the functional interactions behave end-to-end with respect to key system QoS properties, such as
    1. How, when and which resources are committed to client/server interactions at multiple levels of distributed systems
    2. The proper application and system behavior if available resources are different than the expected resources and
    3. The failure detection and recovery strategies necessary to continue to meet high level end-to-end requirements under varying resource availability

This approach enables these QoS properties and resource strategies to change independently from the application, *e.g.,* over different or varying system configurations or resources for the same application, and/or to be changed in response to changes in the application requirements under the prevailing conditions.

In next-generation network embedded systems, common off-the-shelf middleware—rather than operating systems, networks or applications in isolation—will be responsible for separating and controlling DRE system QoS properties from the functional application properties. Middleware will also coordinate the QoS of the various cooperating DRE system and application resources end-to-end in a manner that is responsive to the changing conditions under which they operate.

This presentation reviews recent progress in developing a runtime adaptive framework for building adaptable components and systems, and illustrates its use in the context of several DoD applications and system concepts being developed using the QuO middleware. We will briefly describe two of these applications, Weapons Systems Open Architecture (WSOA) and Unmanned Aerial Vehicles (UAV) video sensor data management, which have already undergone a series of evaluation steps to determine the suitability of their concepts and implementations under realistic use scenarios. Our focus will be on the adaptivity they exhibit under changing operating conditions, and on the technical basis for the effective marshalling of modified workplans toward keeping the application mission objectives focused using available resources. Our methodology involves developing these applications alongside developing some of the underlying technologies themselves, and we show how they influence each other. In addition we assess the suitability of the solutions developed to date, and discuss some of the difficulties encountered along the way. We conclude with some observations on what has and hasn't been accomplished to date.

These observations include the following:

- Late binding opens many doors to innovative approaches
- Different views of the resource management problem need to be integrated together to form a cohesive system level solution. This involves information sharing across and between these views.
- The ability to change strategies rapidly and with few negative side effects is key (and largely missing today)
- We have successfully demonstrated the feasibility of making do with less so long as it remains targeted at the critical parts
- We have highlighted the context sensitive, changing nature of "what's best" and the need to incorporate that into modern system design strategies
- Scaling to larger contexts and packaging intricate QoS solutions having many interactions with the functional program aspects remain significant issues
- Our solutions remain vulnerable to malicious behavior

Authors Information:
Dr. Richard E. Schantz and Dr. Joseph P. Loyall
BBN Technologies
10 Moulton St.
Cambridge, Ma. 02138
617-873-3550 (voice)
617-873-4328 (fax)
{Schantz, JLoyall}@bbn.com
U.S. citizens