

Flow-Service-Quality (FSQ) Engineering: Foundations for High-Assurance Network Systems Development

Richard C. Linger
CERT Research Center
Software Engrg. Inst.
Carnegie Mellon Univ.
Pittsburgh, PA
rlinger@sei.cmu.edu

Alan R. Hevner
Information Systems &
Decision Sciences Dept
Univ. of South Florida
Tampa, FL
ahevner@coba.usf.edu

Gwendolyn Walton
Dept. of Mathematics &
Computer Science
Florida Southern College
Lakeland, FL
gwalton@flsouthern.edu

Mark G. Pleszkoch
IBM Global Services &
CERT Research Center
Software Engrg. Inst.
Carnegie Mellon Univ.
markp@us.ibm.com

Abstract

High-assurance development of large-scale, network-centric systems must address challenging engineering realities. Flow-Service-Quality (FSQ) engineering provides foundations and practices that accommodate these realities to maintain intellectual control in system analysis, development, and evolution.

1. Network System Realities

Modern societies are dependent on large-scale, network systems whose complexity can exceed engineering capabilities for intellectual control. The result has been frustrations and delays in development, and errors and compromises in operation. Intellectual control does not mean the absence of uncertainties or failures -- they are inevitable -- but rather the capability to address them with rigorous engineering foundations. Any foundations for network system engineering must deal explicitly with challenging realities: heterogeneous topologies with ever-changing boundaries, components, and user populations; dynamic interconnectivity that limits visibility and control; user task flows that traverse components whose security and reliability are often unknown; uncertain COTS product function and quality; extensive asynchronous operations that challenge human understanding; and enterprise requirements for rapid development and high-assurance systems.

Given these realities, what engineering foundations can be defined to maintain intellectual control? It is not sufficient to focus on component engineering, because components alone are inadequate to define the system-level behavior required by an enterprise. Rather, the foundations must support refinement of enterprise missions and tasks into network system structures that in turn prescribe component specifications, quality attributes, and operational procedures. This process must be seamless and scale-free, and deal directly with the

network system realities summarized above that often frustrate high-assurance objectives. Flow-Service-Quality (FSQ) engineering [1,2,3] is an emerging discipline for high-assurance network system analysis, specification, design, validation, implementation, and operation based on these needs. FSQ engineering is comprised of three integrated technologies: Flow Structures, Computational Quality Attributes, and Flow Management Architectures.

2. Flow Structures

Network systems exist to satisfy enterprise mission requirements that are instantiated in sets of user task flows. These flows can be refined into compositions of network services provided by hardware, software, and human components, all subject to quality attribute requirements. In FSQ engineering, Flow Structures are definitions of steps and decisions in mission-based user tasks and their refinements into system service uses. Flow Structures are a stable, first-class artifact with semantics defined to maintain intellectual control amid the uncertainties of network system development and evolution. Flows can be expressed in simple control structures, and can be refined, abstracted, and verified with precision. Flows invoke system services, which can be refined into flows, etc., in a recursive processes that employs identical methods at all levels of design.

Network systems are realistically viewed as topologies of asynchronously communicating components providing expected but occasionally unpredictable behavior that can be composed in various patterns to satisfy user task flows. Such unpredictability, whether stemming from errors, failures, or intrusions, is a pervasive reality of network systems, and an enterprise risk management problem with potentially serious consequences. The mathematical semantics of Flow Structures are defined to support development and verification of flows for such uncertainties as a standard engineering practice. To allow for the unpredictable behavior of services, flow semantics

require specification of only the processing that a flow itself performs, and not the processing of the services it invokes. It is unnecessary to specify details of what services do (which may be unknowable), only what a flow does with their responses, expected or unexpected. This response-based semantics means flows can exhibit deterministic behavior for human understanding, despite the underlying asynchronism of shared service uses. Flow Structure semantics are based on a function-theoretic model that treats flows and their control structures as rules for mathematical functions or relations. Table 1 summarizes requirements for Flow Structure semantics based on the engineering realities of network systems.

Table 1. Flow Structure Semantics

Network System Realities	Requirements for Flow Structure Semantics
Enterprise- and task-based view of network-centric systems and QoS	Semantics treat task flows as primary artifacts for system specification, design, and use
Seamless refinement from enterprise mission to system implementation	Semantics support mission refinement into task flows, then into system service uses
Incomplete knowledge of full behavior of network services	Flow semantics based solely on service responses, not complete service behavior
Uncertainties of function and quality attributes of network services	Flow semantics that require design for uncertainties of failure and compromise
Complexities of pervasive asynchronous behavior in network operations	Flow semantics that can be deterministic for human understanding and analysis
Complexities of network topology specification and design	Flow semantics that define sufficient network topology as the union of all flows
Complexities of system service specification and design	Flow semantics that define service requirements as the union of all uses in flows

Flow Structures exhibit desirable properties for high-assurance system development. A flow verification theorem defines conditions for verifying flows with respect to their intended effects. Primary enterprise flows can be employed in the executive suite for defining and validating overarching business processes. Primary flows reveal requirements for secondary flows, etc., in a process of transitive closure that helps ensure completeness of system specifications. We envision initial design of network topology driven by the union of resulting flows, and the specification of each service driven by the union of all its uses in flows.

4. Computational Quality Attributes

FSQ engineering treats quality attributes (reliability, security, availability, etc.) as high-assurance properties to be defined, computed, and acted upon as dynamic characteristics of systems, with values constantly changing in operation [2]. That is, quality attributes are treated as functions to be computed, and not solely as static, a priori descriptions of properties to be achieved. While such functions rely on what can be computed and differ thereby from traditional methods, they permit new approaches to attribute analysis, design, and evaluation. Attribute requirements can be associated with system service uses embedded within flows. The requirement that attributes be measurable in defined metrics for computation also permits human analysis not otherwise possible. Such Computational Quality Attributes are another first-class artifact in FSQ engineering. CQA embodies a functional attribute model that maps service usage feedback into attribute values, a state transition model for evaluating attributes, and Bayesian methods for dynamic attribute evaluation. Future work will develop attribute-specific models within this framework.

4. Flow Management Architectures

Flow Structures and Computational Quality Attributes support system architectures that carry out dynamic flow and attribute management in execution [2]. Flow Management Architectures can provide design and implementation frameworks for this purpose. Future work will define FMA templates of system topologies and functional capabilities for managing flow instantiations and reconciling their quality attribute requirements with service behavior and capabilities in real-time operation.

5. References

- [1] A. Hevner, R. Linger, A. Sobel, and G. Walton, "The Flow-Service-Quality Framework: Unified Engineering for Large-Scale, Adaptive Systems," *Proceedings of 35th Hawaii International Conference on System Sciences (HICSS-35)*, Hawaii, January 7-10, 2002, IEEE Computer Society Press, Los Alamitos, CA, 2002.
- [2] R. Linger, M. Pleszkoch, G. Walton, and A. Hevner, *Flow-Service-Quality Engineering: Foundations for Network System Analysis and Development*, CMU/SEI-2002-TN-01, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2002.
- [3] A. Hevner, R. Linger, M. Pleszkoch, and G. Walton, "Flow-Service-Quality (FSQ) Engineering for Specification of Complex Systems," *Practical Foundations of Business and System Specifications*, Editors, H. Kilov and K. Baclawski, Kluwer Academic Publishers, Inc., NL, 2003.