A Software Product Line Architecture for Distributed Real-time and Embedded Systems: A Separation of Concerns Approach

By Shih-Hsi “Alex” Liu
University of Alabama at Birmingham

Number of requirements

DRE COTS Composition

Functional

Nonfunctional

Problem Statements

Requirements Tangling:
Func. and nonfunc. req. require verification interchangeably

Abundant Alternatives:
Numerous design alternatives generated

QoS Sensitive:
QoS Satisfaction affects the correctness and performance of systems
Overview of the QoS-driven Product Line

Select DRE components by their spec.

Analyze the commonality, variability, and satisfaction of QoS systemic path families by the TLG++ approach.

A Rule Engine and Knowledge Base
- Stores the functional and nonfunctional spec. of existing components, component dependencies, and composition rules
- Including facts, queries and rules for inferring component composition

Domain Engineering

Application Engineering

Construct a set of software products that share common features by QoS-UniFrame
Current Status and Future Work

An Incremental-and-iterative lifecycle model

Domain Eng.

QoS-driven TLG++

Domain Analysis

ADL in TLG++

Domain Design

Glue/wrapper code

Domain Impl.

Appl. Eng.

QoS-UniFrame

Appl. Analysis

TCPNs

Appl. Design

Glue/wrapper code

Appl. Impl.

Common and variable features

A Software Product Line