http://www.cis.uab.edu/liush

A Software Product Line Architecture for Distributed Real-time and Embedded Systems:

A Separation of Concerns Approach

Rajeev R. Raje, Mihran Tuceryan, Andrew M. Olson
Indiana University-Purdue University Indianapolis
{rraje, tuceryan, aolson}@cs.iupui.edu

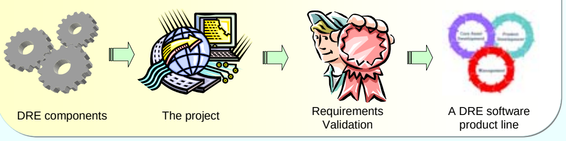
Shih-Hsi "Alex" Liu, Barrett R. Bryant, Jeff Gray
University of Alabama at Birmingham
{liush, bryant, gray}@cis.uab.edu

Mikhail Auguston
Naval Postgraduate School
maugusto@nps.navy.mil

Project Objective

This project presents a novel software product line architecture for component-based Distributed Real-time and Embedded (DRE) systems. The project concentrates on the phases of domain engineering and application engineering to achieve the following objectives:

- Every member of a software product line satisfies its functional and QoS requirements synergistically
- Every member possesses an architectural design
- All members of a software product line share a number of common features. Members possessing various margins of QoS satisfaction are differentiated by variable features



Background

Quality of Service (QoS)

- **Functional Path**: flows of application-specific and functionality-determined information between components*
- **QoS Systemic Path**: determines ***how well*** a functional path behaves in terms of a specific QoS property*

QoS Classification

- **Static**: parameters are design-related
- **Dynamic**: parameters are influenced by the deployment environment
- **Strict**: parameters must satisfy requirements
- **Non-strict**: parameters allow margins of error when meeting requirements
- **Orthogonal**: two parameters have no mutual effects regarding a specific resource
- **Non-orthogonal**: two parameters have mutual influence regarding a specific resource

* N. Wang et al., "QoS-enabled Middleware," in *Middleware for Communications*, Wiley and Sons, 2003.

Two-Level Grammar++ (TLG++)

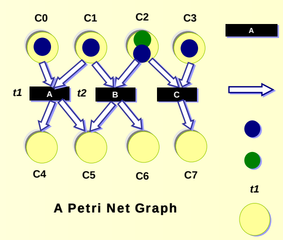
An object-oriented formal specification language which consists of two Context Free Grammars (CFGs)

- The first CFG defines a set of parameters
- The second CFG defines a set of function definitions
- TLG++ has been applied to define programming languages
  - The first CFG defines syntax by production rules
  - The second CFG defines semantics of the production rules
  - An example

```
query :: Boolean.
Syntax :: Sensor Comm1 Comp Comm2 Present.
semantics of QoSSum :
  query := semantics of queryComponent with Sensor, Comm1, Comp,
           Comm2, and  Present;
  if query then semantics of sum with Sensor, Comm1, Comp, Comm2,
           and Present;
```

Timed Colored Petri Nets (TCPNs)

A formalism beneficial in modeling concurrent and asynchronous systems



A Petri Net Graph

- **Transition**: determine what, when and how QoS parameters are to be processed with associated predicates and functions for time, priorities, and event triggers
- **Arc**: control the flowing direction of QoS parameters
- **QoS parameter**: consists of identity, type and range
- **Event**: triggers transition
- **Time**: transition is triggered at a specific time
- **Place**: represents a component in a DRE system

Key Challenges

Challenge 1: QoS sensitive

DRE systems are sensitive to the availability of system resources, which directly or indirectly affect the QoS properties of the system. The magnitudes of such properties influence the feasibility and performance of a DRE system. More precise and less subjective QoS property measurements are required

Challenge 2: Component Composition

- **Evaluation after composition**: As hundreds of QoS properties require satisfaction, it is difficult for the QoS tuning approach to balance and obtain the optimal solution after system composition. In addition, effort is wasted on many infeasible design alternatives after composition
- **Evaluation during composition**: Composition perspective changes between components and QoS properties (i.e., functional and nonfunctional requirements) are tedious and error prone

Challenge 3: Abundant alternatives

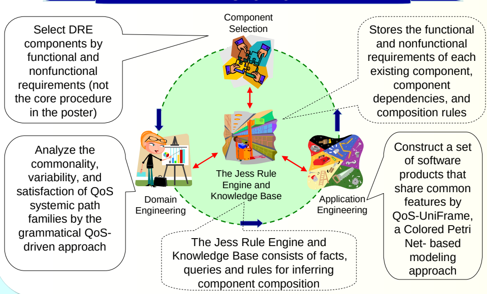
Abundant design alternatives generated from the combination and permutation of selected components are infeasible in terms of functional and nonfunctional requirements

Challenge 4: Costly DRE systems

Many DRE systems are costly and hard to modify. A software product line, which consists of a set of software products sharing common features, will solve the problem

The Separation of Concerns Approach

An Overview



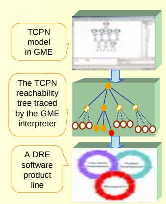
Domain Engineering

TLG++ syntactically and semantically expresses QoS systemic paths

- The first CFG utilizes Extended Backus-Naur Form (EBNF) to define the components and direction of a QoS systemic path
  - EBNF represents ***mandatory, alternative, optional,*** and ***OR*** features
  - Symbol tables are utilized for analyzing the ***commonality*** and ***variability*** of QoS systemic path families
- The second CFG defines component dependencies, composition rules, and QoS satisfaction formula
  - Component dependencies: the relationships between components in terms of function-determined and application-specific tasks
  - Composition rules: verify interface consistency between components and pre- and post-conditions of composition by inferences
  - QoS satisfaction formula: quantitatively estimate the satisfaction of the QoS property of a QoS systemic path
- TLG++, as an Architecture Description Language (ADL), describes the reference architecture

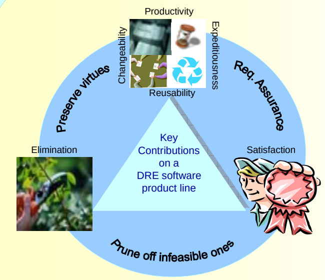
Application Engineering

QoS-UniFrame: utilizes the Generic Modeling Environment (GME), a metaconfigurable modeling tool for expressing TCPNs



- **Objective**: simulates the flows of the QoS systemic paths using time, event and/or priorities of TCPNs
  - Depict state and behavior views of a software system
- **TCPNs**: represents a set of software systems by collections of QoS systemic paths
- **Reachability tree**: explores design alternatives based on different design decisions and permutations
- **QoS requirements**: eliminate ***infeasible*** and ***less probable*** alternatives on the reachability tree by the evaluation of QoS requirements (i.e., the utility functions the corresponding constraints)
- **Consequence**: a set of software products that share common features and possess different satisfaction of QoS properties

Key Contributions



The DRE software product line constructed by the project possesses three major contributions:

- The advantages of applying component based software engineering and software product lines are preserved
- The infeasible design alternatives are pruned off, which reduces the extra workload stated
- Each member satisfies its functional and nonfunctional requirements at requirements and design workflows

A Case Study: The Battlefield Training System

Mobile Augmented Reality Systems

A DRE system concentrating on enriching the user environment by merging ***real*** and ***virtual*** objects

- Six subsystems:
  - **Computation**: performs specific functionalities for the application
  - **Presentation**: exhibits virtual multimedia objects
  - **Tracking and registration**: tracks user's position and orientation and registers virtual objects
  - **Environment model**: store the geometrical and detailed hierarchical 3D information
  - **Interaction**: a user friendly interface for input and output
  - **Wireless communication**: provides mobile communications
- Examples:
  - A battlefield training system (shown at right)

Battlefield Training System

The Battlefield Training System (BTS) assists in training soldiers in different scenarios, strategies, and battlefields

- The BTS consists of:
  - **Real objects**: buildings and obstacles in the battlefield
  - **Virtual objects**: enemies and a hostage displayed on a Head Mounted Display (HMD)
  - **Sensors/Trackers**: fetch the position and orientation of the soldiers
  - **Scenario**: rescue the hostage from the enemies
- The advantages of BTS
  - **Adaptable scenarios**: trains soldier to react and respond properly in different scenarios
  - **Less cost**: simulates highly cost battlefield devices (e.g., tanks and aircrafts)
  - **Less wounded**: reduces the possibilities that soldiers being wounded in the real battlefield