Two-Level Assurance of QoS Requirements for Distributed Real-time and Embedded Systems

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

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

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

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

A QoS-Based Semi-Automatic Design Space Exploration and Assurance Toolkit for Distributed Real-Time and Embedded Heterogeneous Components

Project Objective

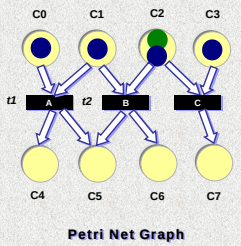
This project presents a semi-automatic toolkit to achieve the following objectives in the context of Distributed Real-time and Embedded (DRE) systems *at system assembly time*:

- Explore possible alternatives based on different design and deployment decisions
- Eliminate infeasible and less probable alternatives based on the evaluation of quality of service (QoS) requirements by using Evolutionary Algorithms (EAs) and statistics
- Assure feasible alternatives and obtain the optimal one based on the QoS utility functions defined in the evolutionary algorithms

This project assists in reducing the overload of designing and developing DRE systems, and provides the advantages of flexibility, and modularization for alternatives analyses.

Background

Petri Nets: A formalism beneficial in modeling concurrent and asynchronous systems



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

Flow: control the flowing direction of QoS parameters

QoS parameter: consists of identity, type and range

Event: triggers transition

Time: transition is triggered at specific time

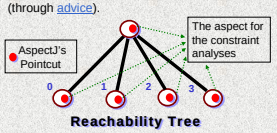
Place: represents a component in a DRE system

PPCEA

- PPCEA: Programmable Parameter Control for Evolutionary algorithms
- A Domain-Specific Scripting Language for Evolutionary Algorithms
- Compute the statistical results for the fitness functions
- ***Discarding policy*** and the ***discard rate*** decide the alternatives to be eliminated by statistics

AspectJ

- Reachability tree: generated based on the dataflow of QoS parameters, controlled by the transitions, events and flows in the Petri Net graph
- AspectJ: an aspect-oriented extension to Java. In this work, it permits the isolation of crosscutting constraint analyses in the nodes of the reachability tree. It helps in identification of participating nodes (through a pointcut), and defines the analyses results on each node (through advice).



Reachability Tree

Generic Modeling Environment (GME)

A metaconfigurable modeling tool that permits the customization of visual domain languages that are capable of code generation.

- Metamodel: define the modeling paradigm, including the syntax, semantics, constraints & presentation of the domain - *Define the modeling paradigm of the Petri Net graph*

- Model: define modeling case by the definitions of its metamodel - *Define a Petri Net graph*

- Interpreter: automatically generate source code based on the model created by the modeler – *Provide execution semantics for a Petri Net*

Key Challenges

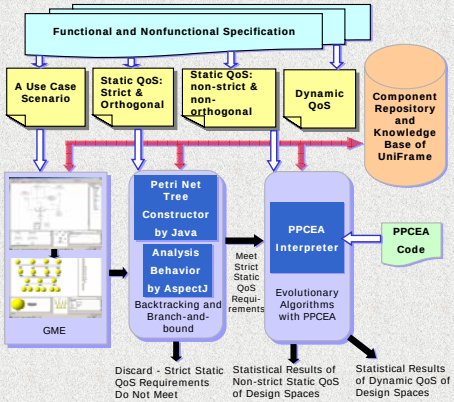
Challenge 1: Design and Deployment for DRE systems

- Numerous components may be selected from the repository.
- The execution permutation, priorities, and event and time triggers for components influence the QoS of DRE systems.

Challenge 2: Characteristics of DRE systems

- DRE systems are naturally expensive and less modifiable.
- Unpredictable behavior occurs in DRE systems, and degrades the confidence of validation.

Overview of QoS-UniFrame

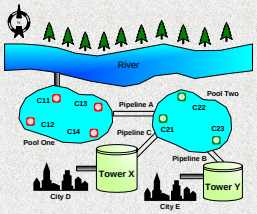


Phases of QoS-UniFrame

- Analyze functional and non-functional requirements.
- Classify QoS parameters.
- Construct a Petri Net model for a DRE system based on the Petri Net metamodel established on the GME.
- The GME interpreter automatically generates the source code of Petri Net reachability construction. This source code is embedded in the backtracking or branch-and-bound algorithm.
- Constraint analyses code written in AspectJ is weaved into the source code of Petri Net reachability tree construction.
- ***Dynamic and parallel*** elimination for infeasible alternatives by backtracking or branch-and-bound algorithm.
- PPCEA obtains the optimal solutions of QoS utility functions. Less probable alternatives may be eliminated based on the statistical results. The results are stored back to the knowledge base as the references of runtime evaluations.

A Case Study

Description



QoS Parameter Classification

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

Non-functional Requirements

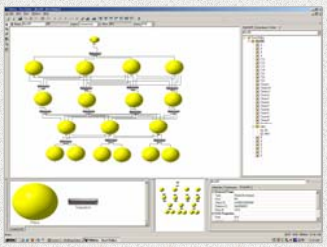
(a) The total flow processing capacity is at least 50 million gallons per day.

(b) The battery life of each Treatment Unit (TU) has at least 15 hours left.

(c) Total CPU usage is at most 70%.

(d) Total water treatment volume of selected TUs is at least 35 million gallons per day.

Petri Net Modeling in GME



Constraint Analysis by AspectJ

The constraint analysis code for "Maximum Flow Processing Capacity" is written in AspectJ

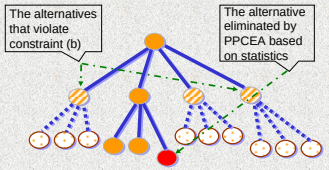
```
public aspect Analysis {
  pointcut Monitor (QoSPar par) :
  call(public void *.createNode(..)) &&
    args(par);
    after(QoSPar par1) : Monitor(par1) {
      double temp=0;
      if (par1.getName().equals("MPC")) {
        temp=par1.getValue();
      //evaluate MPC's QoS requirements here
}}}
```

PPCEA

PPCEA eliminates the less probable alternatives

```
genetic
   Discard := 1.1;   //discard rate
   while ( t <= 10 ) do
     init; call_EA;  //initiate and run EA
     Temp := Temp + Worst;
     t := t + 1
   end; Temp := Temp / t;
   if (Temp > QoS * Discard) delete_gene; fi;
end genetic
```

Experimental Results of QoS-UniFrame



Key Contributions

- A QoS-Driven Approach to deal with abundant QoS parameters.
- Petri Nets as a formalism including predicates, time and event constraints.
- GME as a configurable and customizable tool for Petri Nets.
- AspectJ provides modular and changeable merits for constraint analyses.
- Dynamic and parallel approach for eliminating infeasible alternatives.
- Statistical method to delete less probable alternatives.