An End-to-End Domain Specific Modeling and Analysis Platform

Arman Shahbazian, George Edwards, Nenad Medvidovic
Computer Science Department
University of Southern California, Los Angeles, USA

Motivation

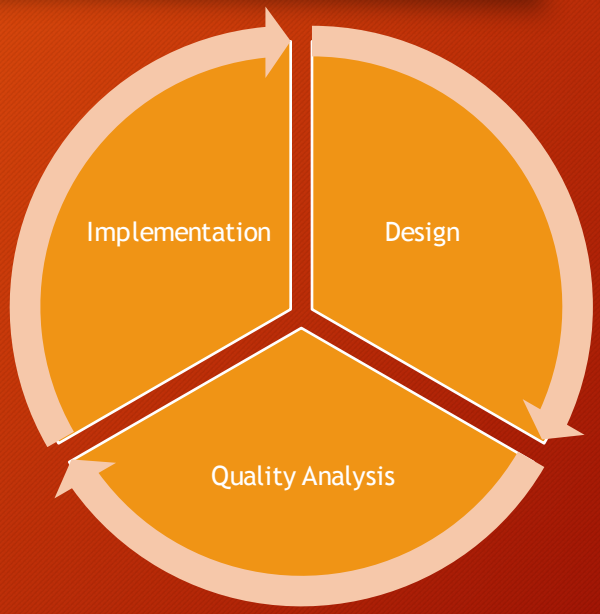
USC Viterbi
School of Engineering

- Architecture modeling is critically important
  - Software systems are growing in size
  - Early design decisions drastically affect eventual system quality
  - Models form a basis for rationalizing design decisions
- Advances in different areas tend to be disconnected
  - Modeling
  - Analysis
  - Simulation
  - Implementation
  - Deployment
  - Evolution
- “One size fits all” approaches have drawbacks

Purpose of Architecture Modeling

USC Viterbi
School of Engineering

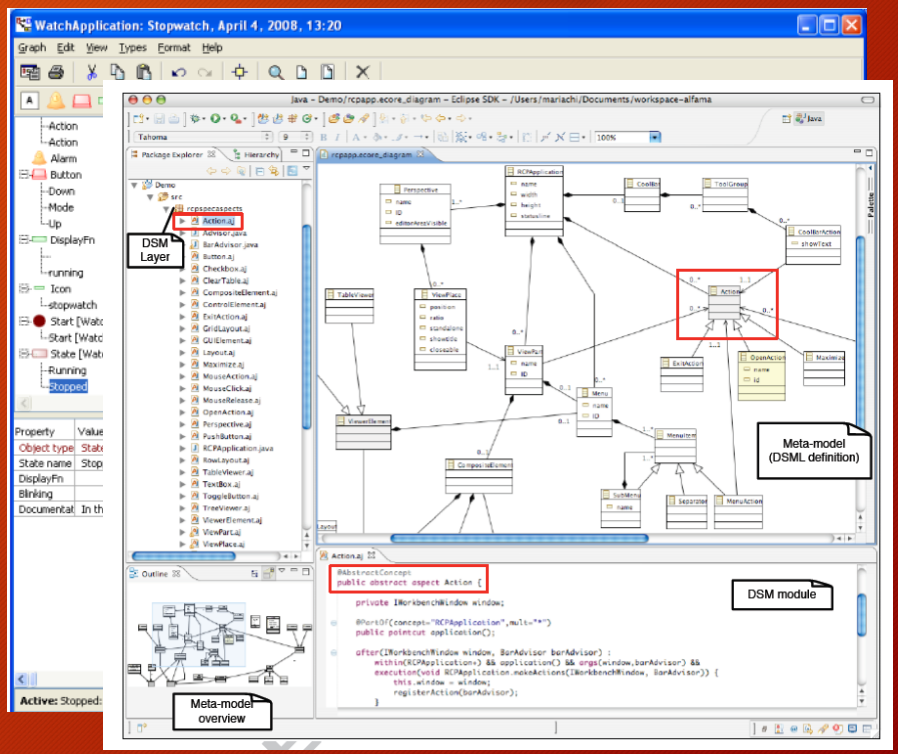
- Documentation and communication
- Optimization and verification
- Automation of engineering tasks



Domain-Specific Models

USC Viterbi
School of Engineering

- Rely on domain specific languages (**DSLs**)
  - Customized for a problem family
  - Defined via **metamodels**
- Concise and intuitive
  - No missing or extra features
  - Capture patterns
  - Enforce constraints
  - Use native symbols and terms
- Can be modified, evolved, composed



DomainPro

USC Viterbi
School of Engineering

- Leverages DSLs
- Simplifies and automates development
  - Tool support for defined DSLs
  - Fully customizable modeling UI
  - Out-of-the-box simulation-capable models
- Supports engineers in key activities
  - Model design
  - Model analysis (currently via simulation)
  - Implementation
- Extensible via pluggable architecture

Key Elements

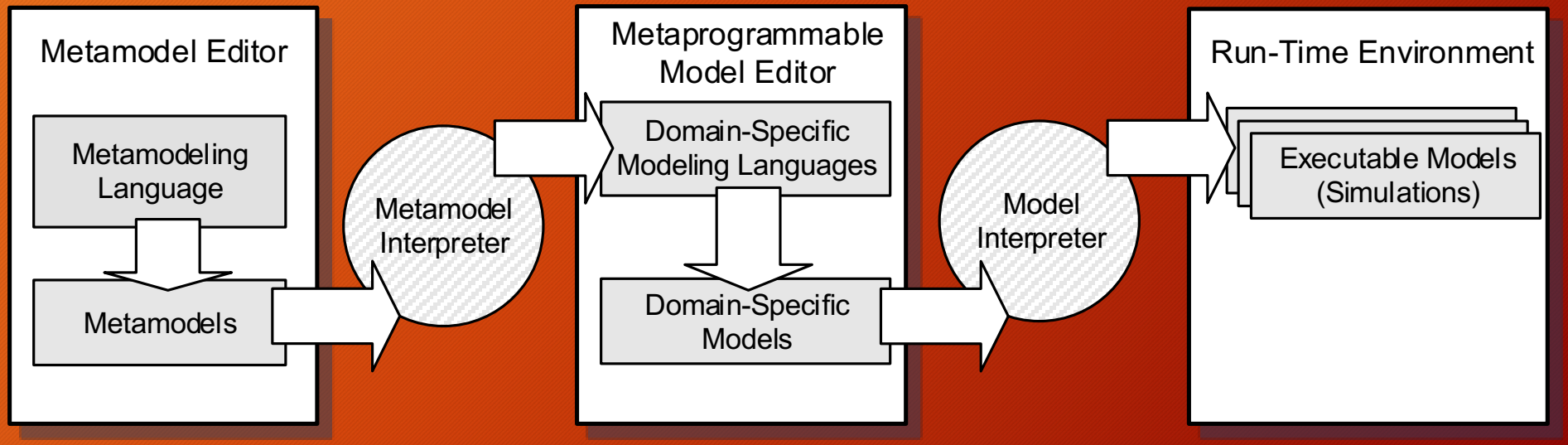
USC Viterbi
School of Engineering

- Metamodel
  - Concisely captures DSL semantics
  - Metamodel interpreter generates model transformation rules
- Model interpreter framework (MIF)
  - Applies model transformation algorithms according to rules
- Simulation
  - MIF for fully configured discrete-event simulations

DomainPro Workflow

USC Viterbi
School of Engineering

1. Metamodel editor with built-in metamodeling language
2. Metamodel interpreter configures a model editor
3. Model interpreter generates executable simulations



Metamodeling

- **Metatypes** founded on canonical architectural constructs
- Basis for styles, patterns, reference architectures
- May reflect desired analysis techniques
- Embedded **metatype semantics** defined in terms of domain-independent capabilities and constraints
- **Metatype properties** capture the capabilities and constraints of a particular domain

Model Interpreter Frameworks

USC Viterbi
School of Engineering

- Reusable templates for constructing model interpreters
- Artifacts useful in a variety of contexts
  - *e.g.*, finite state machines, discrete event simulations
- Algorithms for performing semantic transformations
  - *e.g.*, pattern matching, model traversal
- Auto-generated type specifications for "understanding" domain-specific models
  - *e.g.*, XML files, C# plug-ins

Simulation

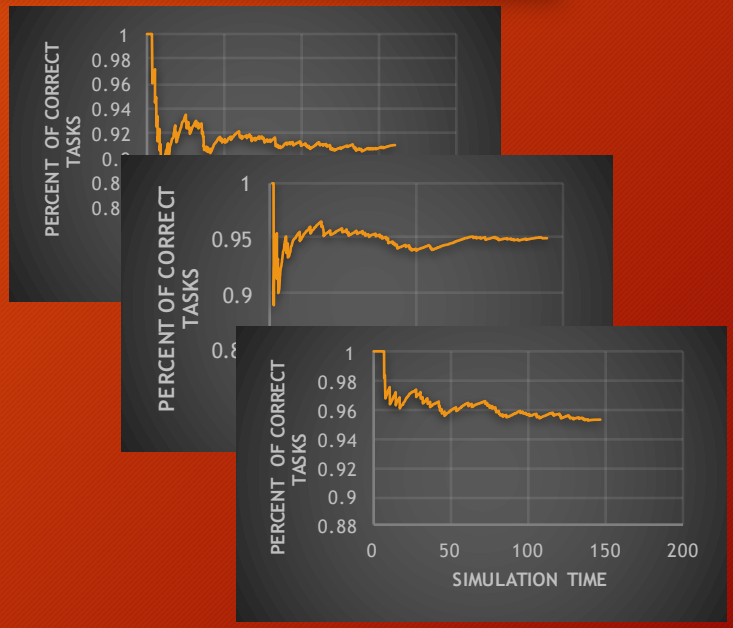
USC Viterbi
School of Engineering

- An **architecture-centric** discrete event simulation engine
- Facilities for inserting **event listeners**
  - Monitor, compute, and record simulation data
  - Functional or non-functional behavior
- Allows optimization of simulation engine functions
  - Scheduling
  - Routing
  - Dispatching

Simulation Use Cases

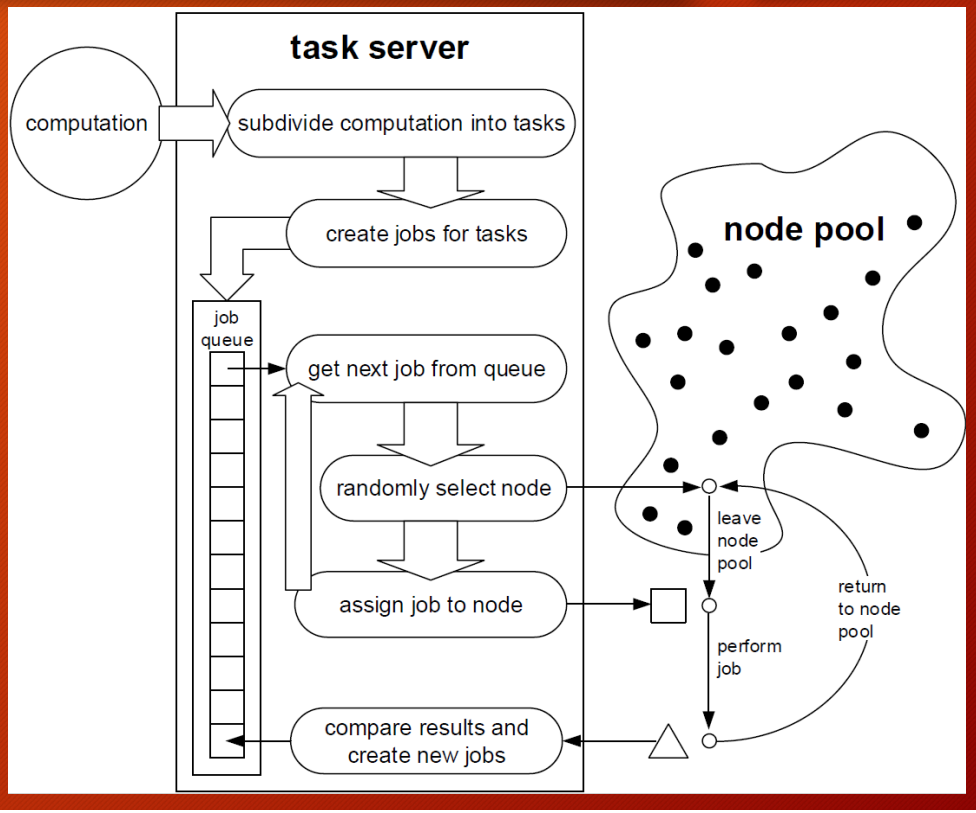
USC Viterbi
School of Engineering

- Providing design rationale
- Weighing architectural trade-offs
- Discovering emergent behaviors
- Validating component implementations



Example System

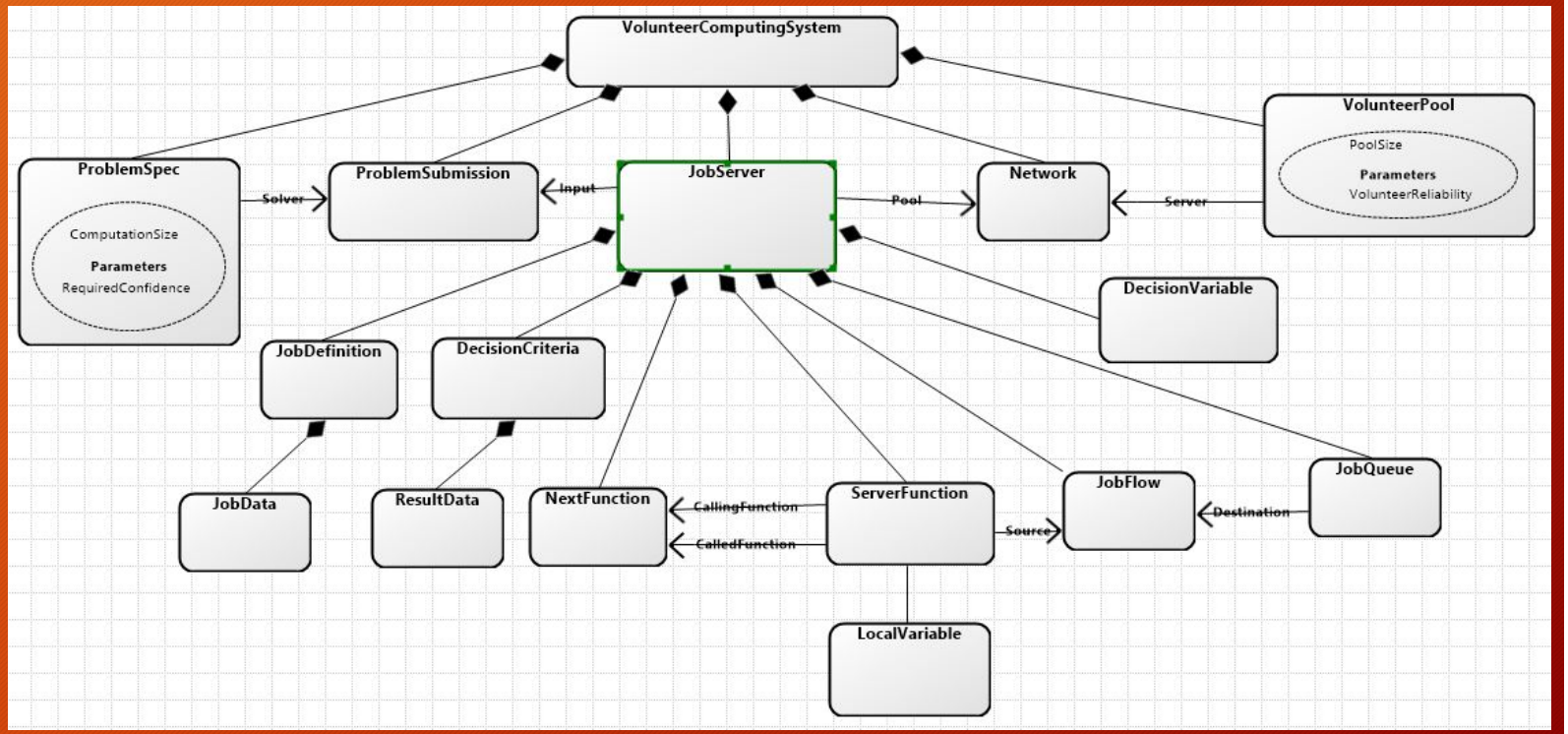
- Distributed computation architecture (DCA)
  - Skype
  - Hadoop
  - BOINC

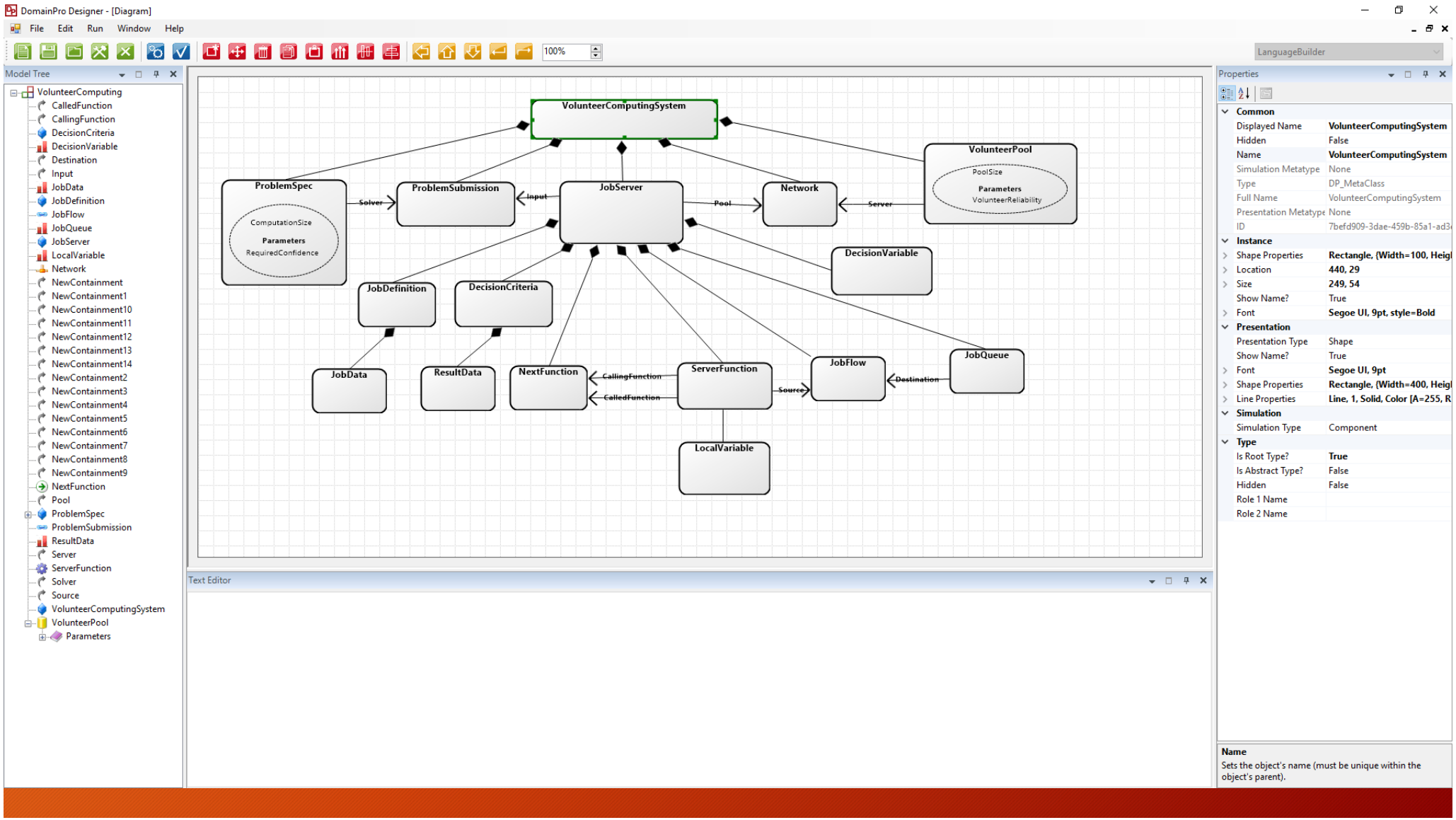


DCA Metamodel

USC Viterbi
School of Engineering

- Metatype properties:
  - *Computation Size*
  - *Required Confidence*
  - *Node Reliability*
  - *Pool Size*

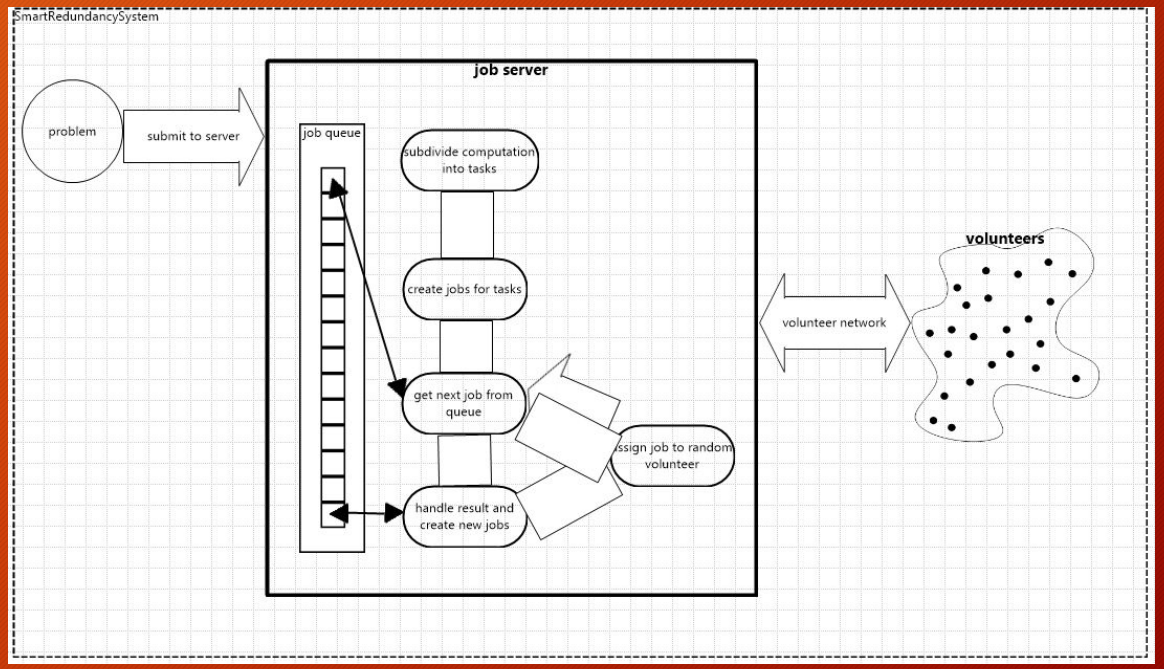



DomainPro Designer - [Diagram]
File
Edit
Run
Window
Help
100%
LanguageBuilder
Model Tree
VolunteerComputing
CalledFunction
CallingFunction
DecisionCriteria
DecisionVariable
Destination
Input
JobData
JobDefinition
JobFlow
JobQueue
JobServer
LocalVariable
Network
NewContainment
NewContainment1
NewContainment10
NewContainment11
NewContainment12
NewContainment13
NewContainment14
NewContainment2
NewContainment3
NewContainment4
NewContainment5
NewContainment6
NewContainment7
NewContainment8
NewContainment9
NextFunction
Pool
ProblemSpec
ProblemSubmission
ResultData
Server
ServerFunction
Solver
Source
VolunteerComputingSystem
VolunteerPool
Parameters
VolunteerComputingSystem
ProblemSpec
ComputationSize
Parameters
RequiredConfidence
Solver
ProblemSubmission
Input
JobServer
Pool
Network
Server
VolunteerPool
PoolSize
Parameters
VolunteerReliability
DecisionVariable
JobDefinition
DecisionCriteria
JobData
ResultData
NextFunction
CallingFunction
CalledFunction
ServerFunction
Source
JobFlow
Destination
JobQueue
LocalVariable
Text Editor
Properties
Common
Displayed Name
VolunteerComputingSystem
Hidden
False
Name
VolunteerComputingSystem
Simulation Metatype
None
Type
DP_MetaClass
Full Name
VolunteerComputingSystem
Presentation Metatype
None
ID
Instance
Shape Properties
Rectangle, (Width=100, Heig
Location
440, 29
Size
249, 54
Show Name?
True
Font
Segoe UI, 9pt, style=Bold
Presentation
Presentation Type
Shape
Show Name?
True
Font
Segoe UI, 9pt
Shape Properties
Rectangle, (Width=400, Heig
Line Properties
Line, 1, Solid, Color [A=255, R
Simulation
Simulation Type
Component
Type
Is Root Type?
True
Is Abstract Type?
False
Hidden
False
Role 1 Name
Role 2 Name
Name
Sets the object's name (must be unique within the object's parent).

DCA Model

USC Viterbi
School of Engineering

- Quality attributes:
  - *Reliability*
    - Percentage of tasks computed correctly
  - *Efficiency*
    - Total number of generated jobs



DomainPro Designer - [Diagram]

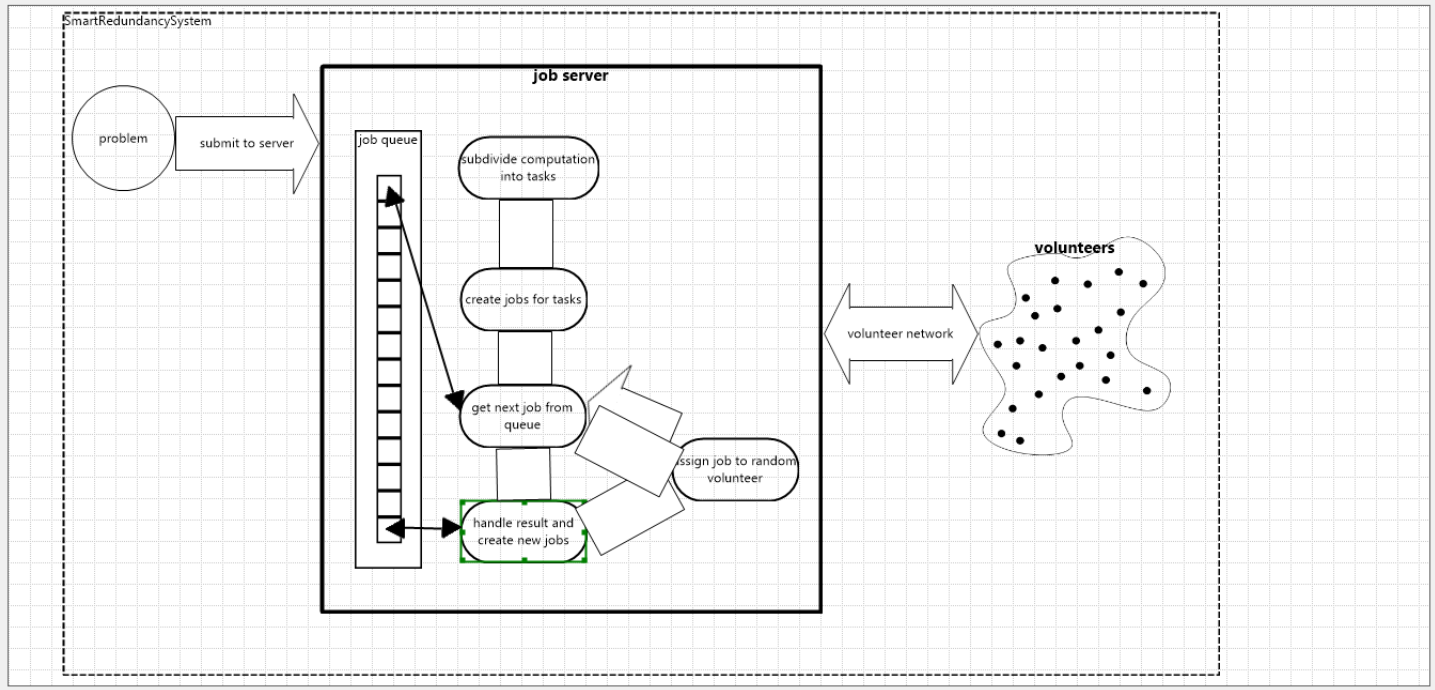
File Edit Run Window Help

100%

VolunteerComputing

Model Tree

- SmartRedundancy
  - SmartRedundancySystem
    - NewProblemSubmission
    - Problem
    - SimpleJobServer
      - AssignJob
      - Cost
      - CreateJobs
      - EndOfQueue
      - FinishedAssignment
      - FinishedCreate
      - FinishedGettingJob
      - FinishedHandleResult
      - FinishedSubdivide
      - FrontOfQueue
      - GetNextJob
      - HandleResult
      - KParameter
      - NewNextFunction
      - NumCorrect
      - NumIncorrect
      - NumTasks
      - PercentCorrect
      - ReceivedResults
      - SubdivideComp
      - TotalExecutionTime
      - UniformJob
      - UniformJobQueue
      - Vote
    - VolunteerNetwork
    - Volunteers



Text Editor

Run | Duration

```
SimpleJobServer.Cost = (SimpleJobServer.VolunteerNetwork.Volunteers.PoolSize*SimpleJobServer.VolunteerNetwork.Volunteers.VolunteerReliability +
 SimpleJobServer.VolunteerNetwork.Volunteers.PoolSize*SimpleJobServer.VolunteerNetwork.Volunteers.ProcessingPower)+(new Random()).NextDouble()/100 ;
if (!SimpleJobServer.ReceivedResults.ContainsKey(JobAssignment.JobId))
      SimpleJobServer.ReceivedResults.Add(JobAssignment.JobId, (Vote)SimpleJobServer.Create("Vote"));

double r = Random.Uniform();

if (r < SimpleJobServer.VolunteerNetwork.Volunteers.VolunteerReliability)
{
      SimpleJobServer.ReceivedResults[JobAssignment.JobId].Correct++;
}
else
{
```

Properties

Common	
Displayed Name	**handle result and create new**
Hidden	False
Name	**HandleResult**
Simulation Metatype	Method
Type	ServerFunction
Full Name	SmartRedundancySystem.Simp
Presentation Metatype	Shape
ID	92ab85d4-e9b3-4adc-a62d-141
Instance	
Shape Properties	**Rectangle, (Width=100, Heig**
Location	**131, 407**
Size	**120, 60**
Show Name?	True
Font	**Segoe UI, 9pt**
Simulation	
Resource Dependency	**VolunteerNetwork**
Resource Request	**Jobs**

Name

Sets the object's name (must be unique within the object's parent).

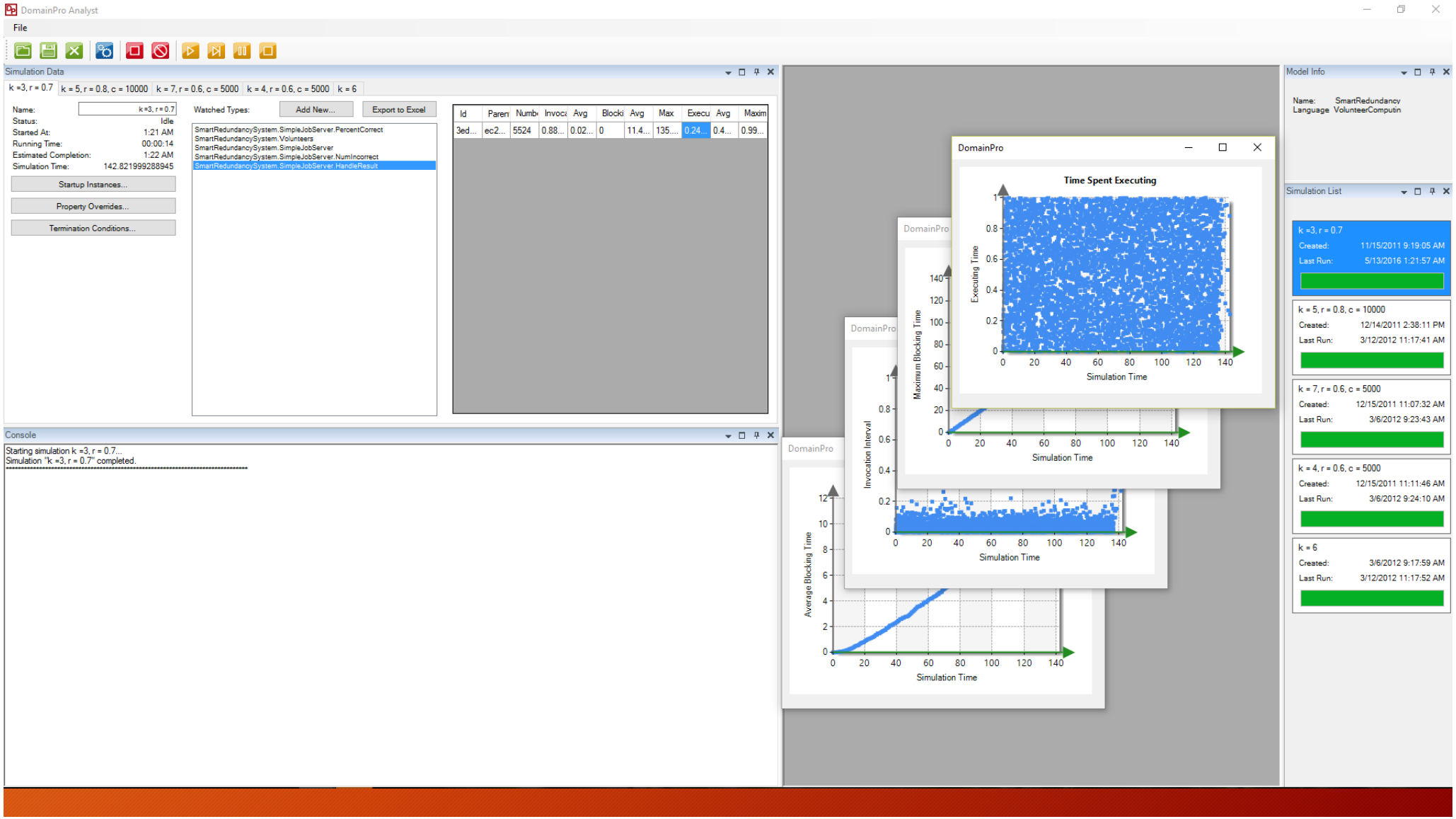
A Trade-Off Scenario in DCA

USC Viterbi
School of Engineering

- *Reliability* vs. *efficiency*
  - Two scenarios
  - Similar *reliability*
  - 6.5x more *efficiency* in Scenario #1

Required Confidence: 3
Node Reliability: 75 %
Generated Jobs: 5630

Required Confidence: 8
Node Reliability: 60 %
Generated Jobs: 36684


DomainPro Analyst
File
Simulation Data
k =3, r = 0.7
k = 5, r = 0.8, c = 10000
k = 7, r = 0.6, c = 5000
k = 4, r = 0.6, c = 5000
k = 6
Name: k =3, r = 0.7
Status: Idle
Started At: 1:21 AM
Running Time: 00:00:14
Estimated Completion: 1:22 AM
Simulation Time: 142.821999288945
Startup Instances...
Property Overrides...
Termination Conditions...
Watched Types:
Add New...
Export to Excel
SmartRedundancySystem.SimpleJobServer.PercentCorrect
SmartRedundancySystem.Volunteers
SmartRedundancySystem.SimpleJobServer
SmartRedundancySystem.SimpleJobServer.NumIncorrect
SmartRedundancySystem.SimpleJobServer.HandleResult
Id
Paren
Numb
Invoc
Avg
Block
Avg
Max
Execu
Avg
Maxim
3ed...
ec2...
5524
0.88...
0.02...
0
11.4...
135....
0.24...
0.4...
0.99...
Console
Starting simulation k =3, r = 0.7...
Simulation "k =3, r = 0.7" completed.
DomainPro
Time Spent Executing
Executing Time
Simulation Time
Maximum Blocking Time
Invocation Interval
Average Blocking Time
Model Info
Name: SmartRedundancy
Language VolunteerComputin
Simulation List
k =3, r = 0.7
Created: 11/15/2011 9:19:05 AM
Last Run: 5/13/2016 1:21:57 AM
k = 5, r = 0.8, c = 10000
Created: 12/14/2011 2:38:11 PM
Last Run: 3/12/2012 11:17:41 AM
k = 7, r = 0.6, c = 5000
Created: 12/15/2011 11:07:32 AM
Last Run: 3/6/2012 9:23:43 AM
k = 4, r = 0.6, c = 5000
Created: 12/15/2011 11:11:46 AM
Last Run: 3/6/2012 9:24:10 AM
k = 6
Created: 3/6/2012 9:17:59 AM
Last Run: 3/12/2012 11:17:52 AM

Contributions and Future Work

USC Viterbi
School of Engineering

- Contributions
  - End-to-end domain specific modeling
  - Automated generation of model interpreters
  - Discrete event simulation
  - Extensible architecture
- Future work
  - Automating model optimization
  - Isomorphic treatment of modeling, analysis, implementation
  - Additional case studies

An End-to-End Domain Specific Modeling and Analysis Platform

Arman Shahbazian, George Edwards, Nenad Medvidovic
Computer Science Department
University of Southern California, Los Angeles, USA