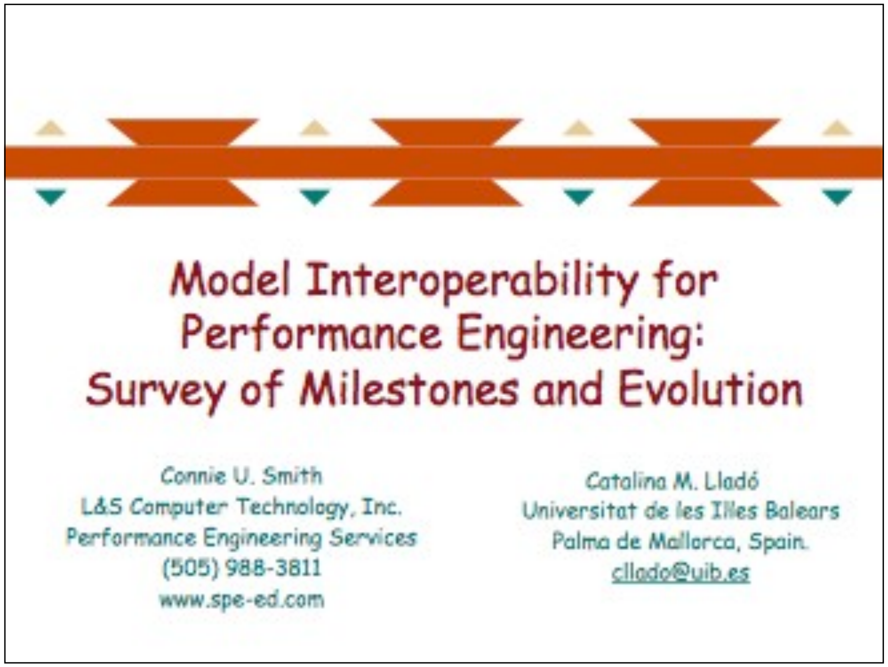
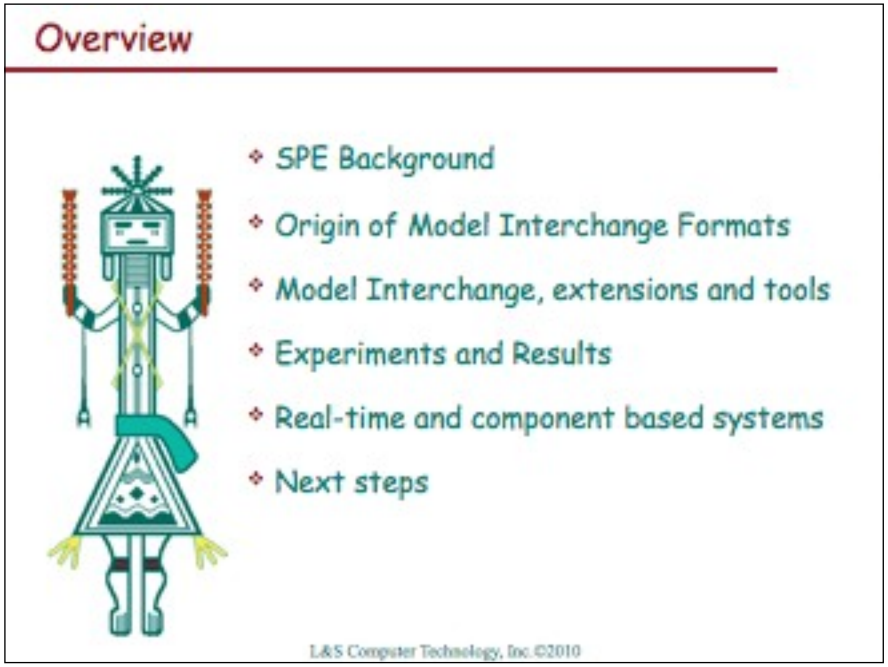


Model Interoperability for Performance Engineering: Survey of Milestones and Evolution

Connie U. Smith
L&S Computer Technology, Inc.
Performance Engineering Services
(505) 988-3811
www.spe-ed.com

Catalina M. Lladó
Universitat de les Illes Balears
Palma de Mallorca, Spain.
cllado@uib.es



Overview

- SPE Background
- Origin of Model Interchange Formats
- Model Interchange, extensions and tools
- Experiments and Results
- Real-time and component based systems
- Next steps

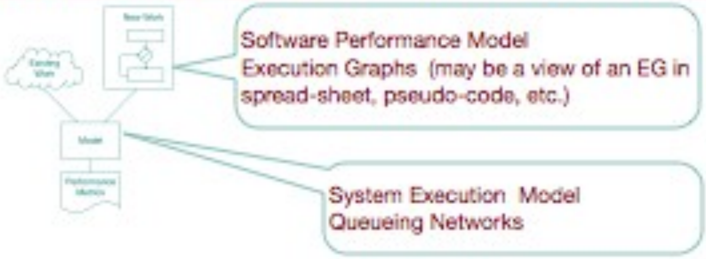
Part 1: Short SPE Background

Software Performance Engineering (SPE) Goal

- Early, model-based assessment of software decisions to determine performance impact

- Architecture
  - has the most significant influence on performance
  - most difficult to change

System Versus Software Modeling Tools



Software Performance Model
Execution Graphs (may be a view of an EG in spread-sheet, pseudo-code, etc.)

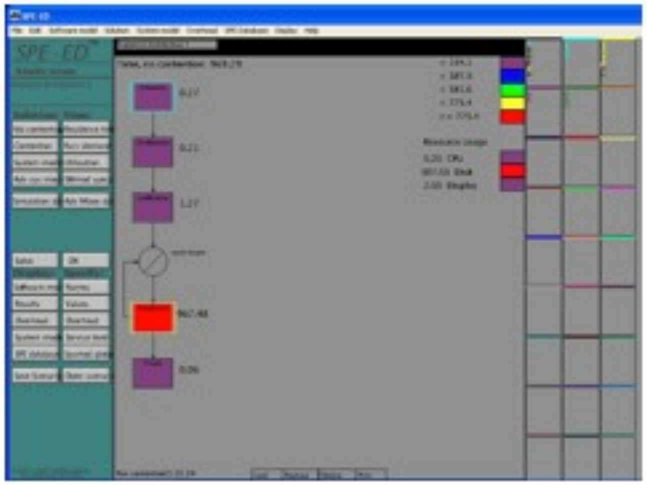
System Execution Model
Queueing Networks

System	Software
Requires more modeling expertise	Requires less modeling expertise
Device usage, overall response time and throughput	Time and resource requirements of processing steps and overall
Useful to evaluate hardware changes	Useful to evaluate software alternatives

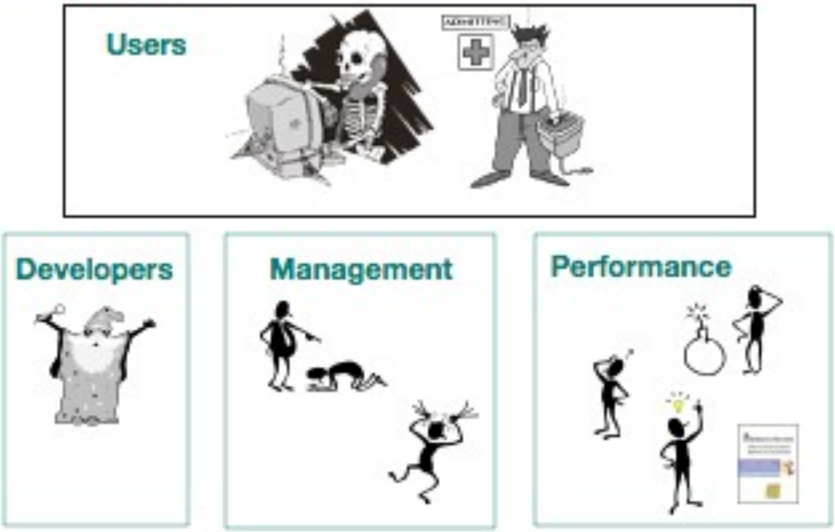
A combination is best.

SPE·ED

Tool for performance engineers



SPE Reality



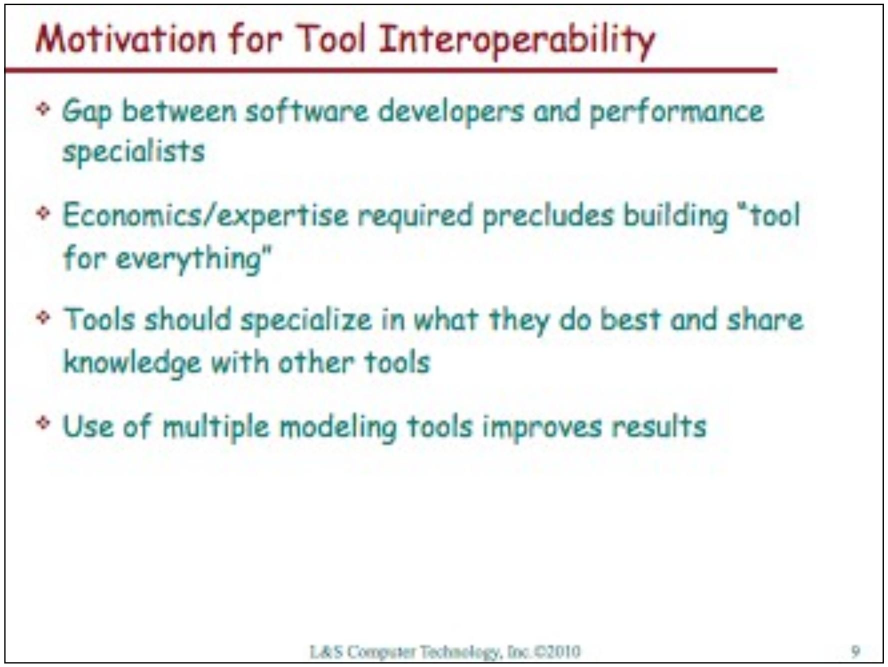
Users

Developers

Management

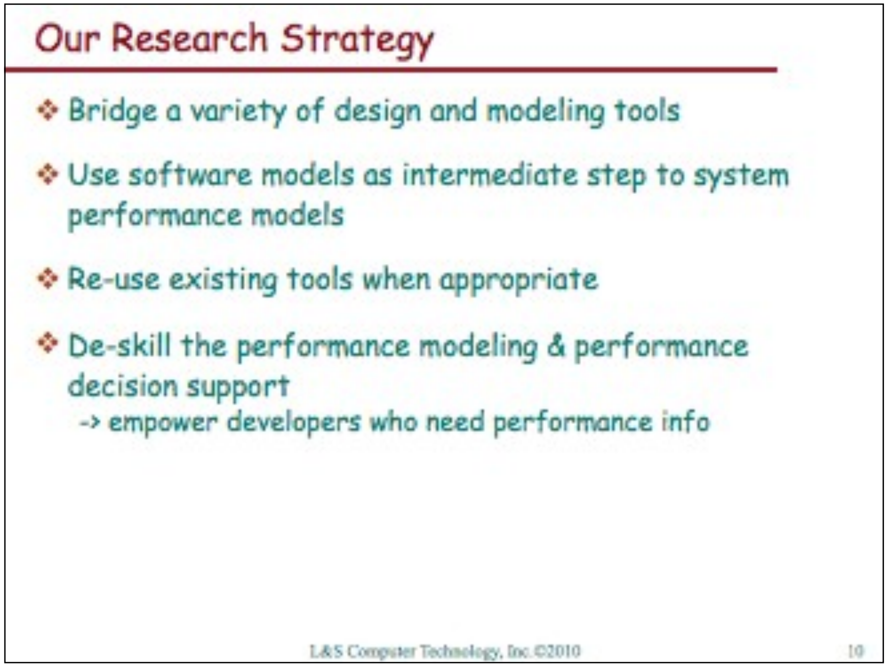
Performance

Part 2: Origin of Model Interchange Formats



Motivation for Tool Interoperability

- Gap between software developers and performance specialists
- Economics/expertise required precludes building "tool for everything"
- Tools should specialize in what they do best and share knowledge with other tools
- Use of multiple modeling tools improves results



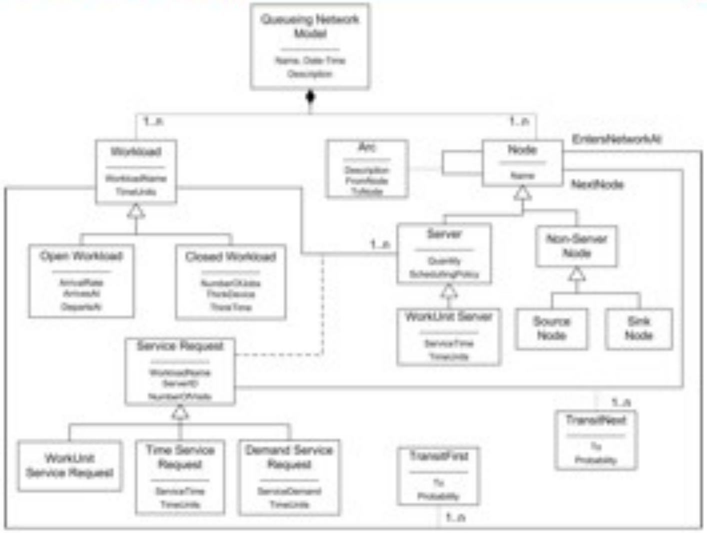
Our Research Strategy

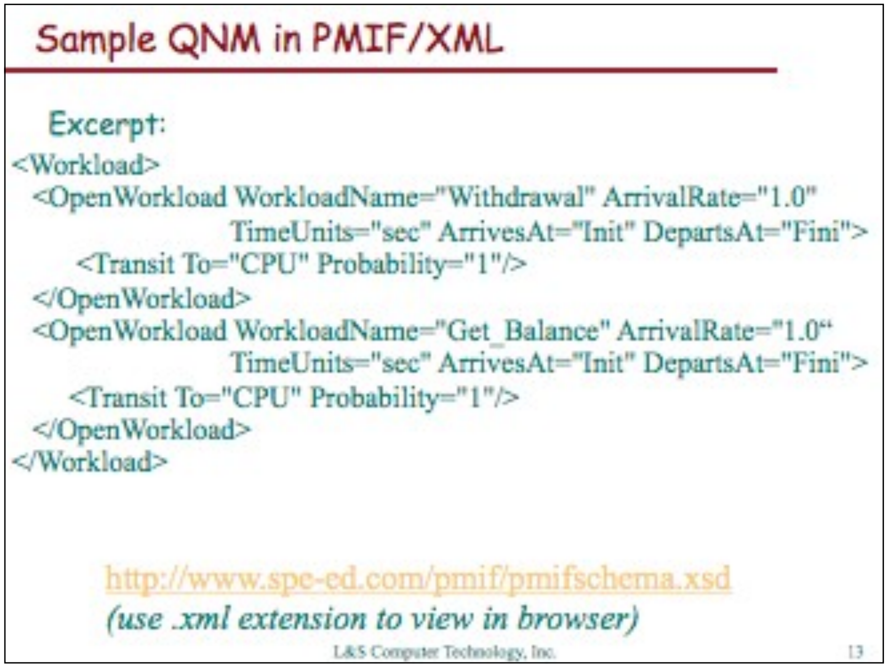
- Bridge a variety of design and modeling tools
- Use software models as intermediate step to system performance models
- Re-use existing tools when appropriate
- De-skill the performance modeling & performance decision support
  -> empower developers who need performance info

Research Results – System Model Interchange

- Performance Model Interchange Format (PMIF) (Smith & Williams – Tools 1997 panel, JSS 1999)
- New version of the PMIF specification (PMIF 2.0) (Smith & Lladó Qest 2004)
  - XML implementation
  - Prototype proof of concept
    - Export from SPE·ED (export interface)
    - Import to Qnap (file translation)
    - Export from Qnap (syntactical and lexical analyzer) – CLEI 2005
- Latest version Performance Evaluation 2010

PMIF Meta-model





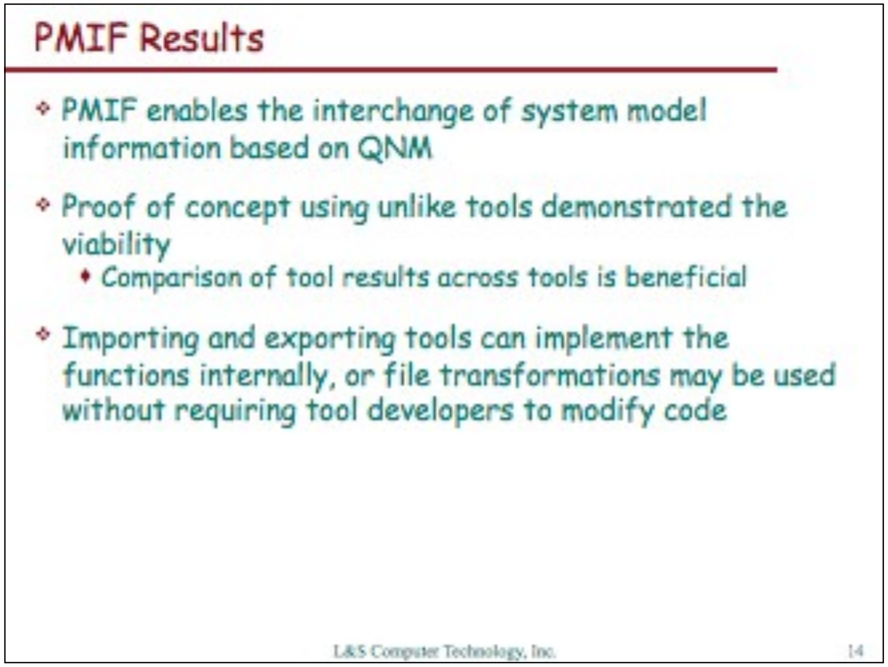
Sample QNM in PMIF/XML

Excerpt:

```
<Workload>
  <OpenWorkload WorkloadName="Withdrawal" ArrivalRate="1.0"
                    TimeUnits="sec" ArrivesAt="Init" DepartsAt="Fini">
      <Transit To="CPU" Probability="1"/>
  </OpenWorkload>
  <OpenWorkload WorkloadName="Get_Balance" ArrivalRate="1.0“
                    TimeUnits="sec" ArrivesAt="Init" DepartsAt="Fini">
     <Transit To="CPU" Probability="1"/>
  </OpenWorkload>
</Workload>
```

http://www.spe-ed.com/pmif/pmifschema.xsd

(use .xml extension to view in browser)



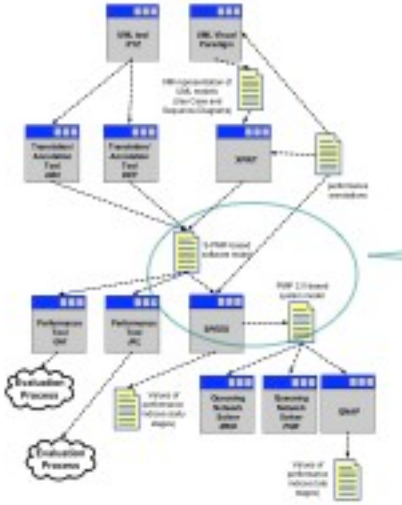
PMIF Results

- PMIF enables the interchange of system model information based on QNM
- Proof of concept using unlike tools demonstrated the viability
  - Comparison of tool results across tools is beneficial
- Importing and exporting tools can implement the functions internally, or file transformations may be used without requiring tool developers to modify code

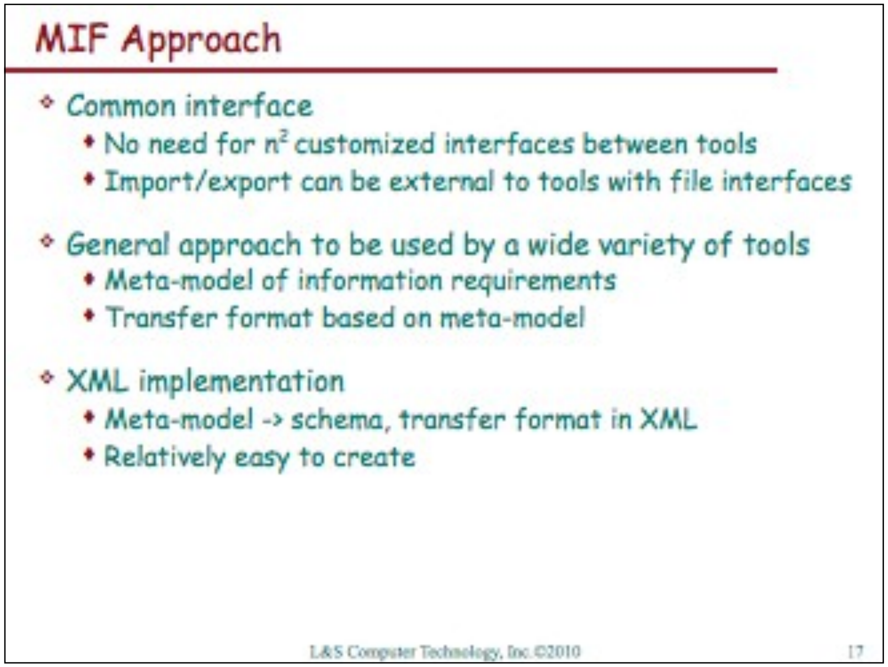
Research Results - Software Model

- Interchange between design tools and software performance modeling tools
- SPE Meta-Model (Williams & Smith, Tools 1995)
  - Defines information requirements for the interchange
- S-PMIF (Cortellessa, di Marco, Lladó, Smith, Williams WOSP 2005)
  - XML schema, implementation, proof of concept
  - Poseiden Visual Paradigm -> XPIRIT -> SPE-ED

UML Design Models -> Performance Models

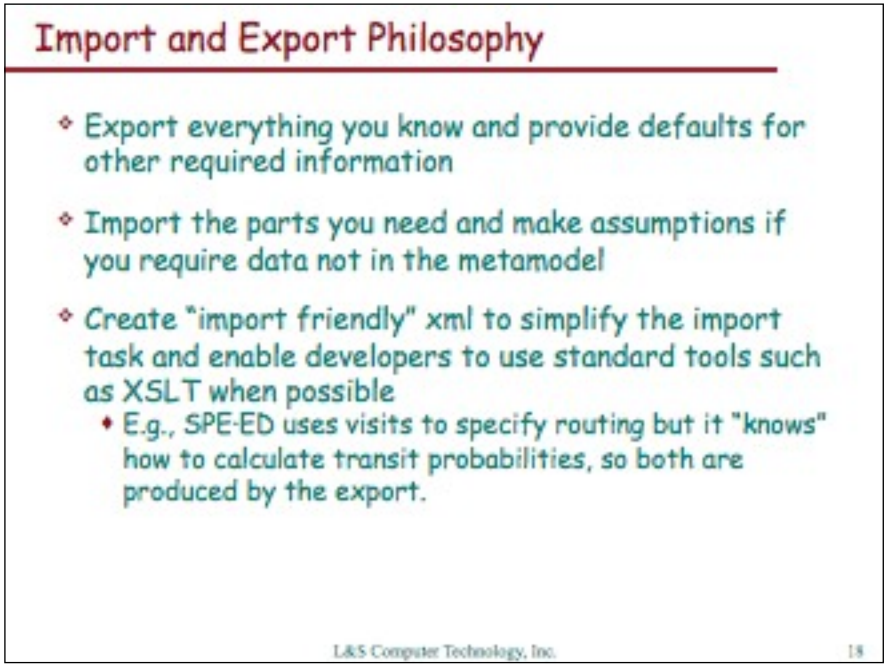


Model Interchange Formats (MIFs) streamline model interoperability process



MIF Approach

- Common interface
  - No need for n^2 customized interfaces between tools
  - Import/export can be external to tools with file interfaces
- General approach to be used by a wide variety of tools
  - Meta-model of information requirements
  - Transfer format based on meta-model
- XML implementation
  - Meta-model -> schema, transfer format in XML
  - Relatively easy to create



Import and Export Philosophy

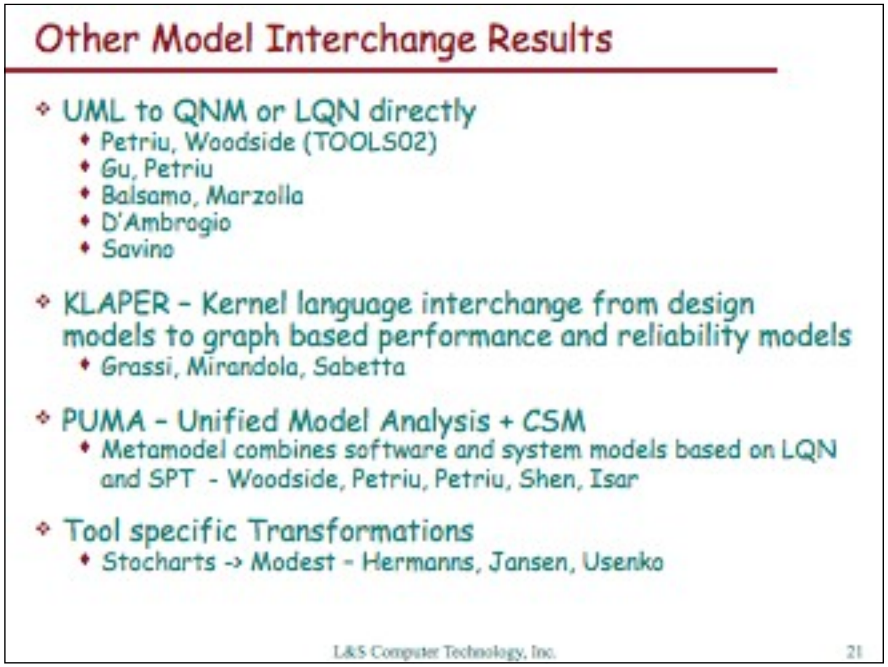
- Export everything you know and provide defaults for other required information
- Import the parts you need and make assumptions if you require data not in the metamodel
- Create "import friendly" xml to simplify the import task and enable developers to use standard tools such as XSLT when possible
  - E.g., SPE-ED uses visits to specify routing but it "knows" how to calculate transit probabilities, so both are produced by the export.

Part 3: Model Interchange: Extensions and Tools

Initial PMIF Extensions

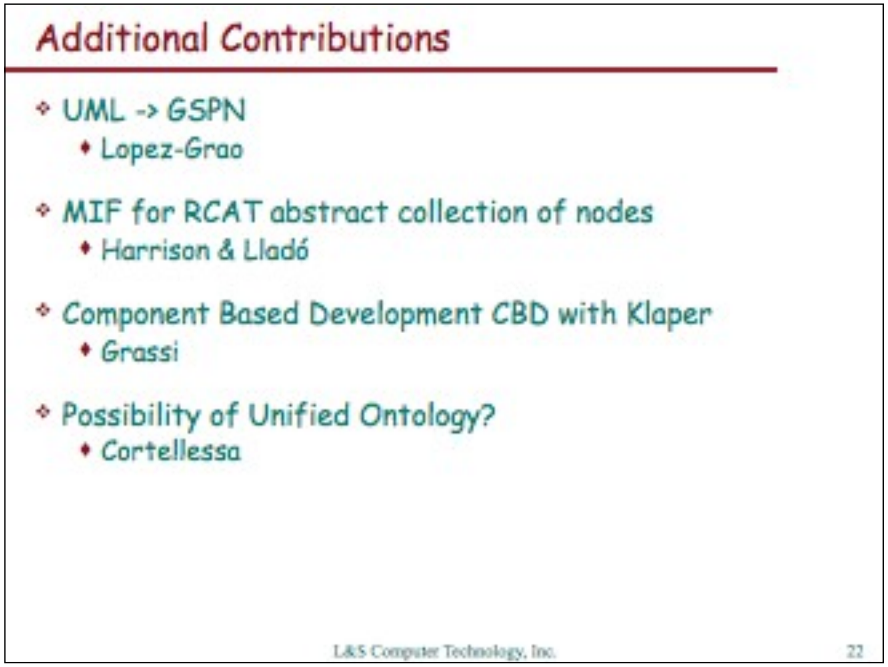
- Web Service implementation (WOSP 2005)
- Numerous, reproducible case studies

- Semantic Validation - ICSEA 2006, tool at Qest 2006
  - PMIF import tools: only one validation code
  - PMIF export tools: to check that they generate correct models
  - Web Service, developed, installed and maintained once for all its users



Other Model Interchange Results

- UML to QNM or LQN directly
  - Petriu, Woodside (TOOLS02)
  - Gu, Petriu
  - Balsamo, Marzolla
  - D'Ambrogio
  - Savino
- KLAPER – Kernel language interchange from design models to graph based performance and reliability models
  - Grassi, Mirandola, Sabetta
- PUMA – Unified Model Analysis + CSM
  - Metamodel combines software and system models based on LQN and SPT - Woodside, Petriu, Petriu, Shen, Isar
- Tool specific Transformations
  - Stocharts -> Modest – Hermanns, Jansen, Usenko



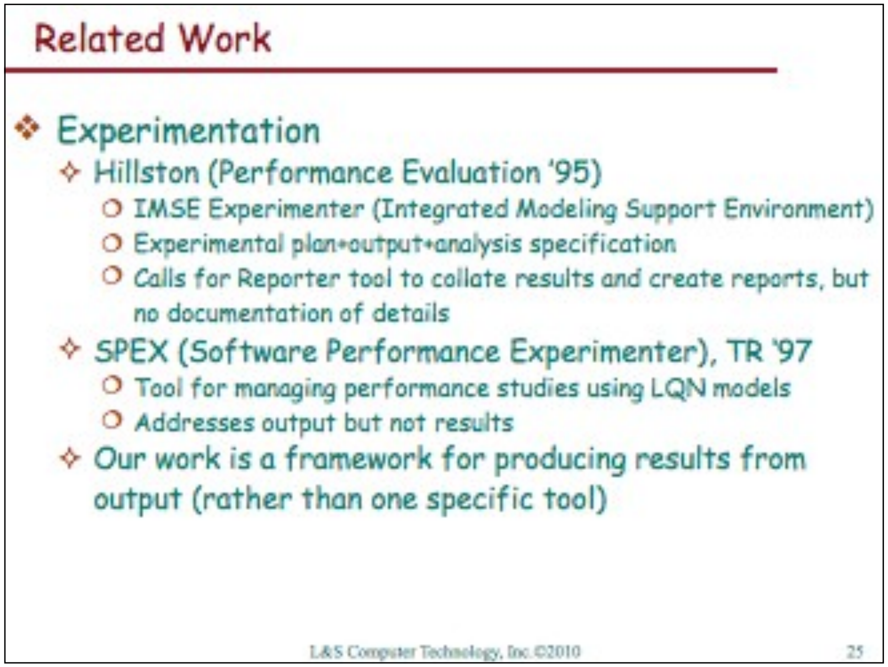
Additional Contributions

- UML -> GSPN
  - Lopez-Grao
- MIF for RCAT abstract collection of nodes
  - Harrison & Lladó
- Component Based Development CBD with Klaper
  - Grassi
- Possibility of Unified Ontology?
  - Cortellessa

Summary: Model Interchange Formats

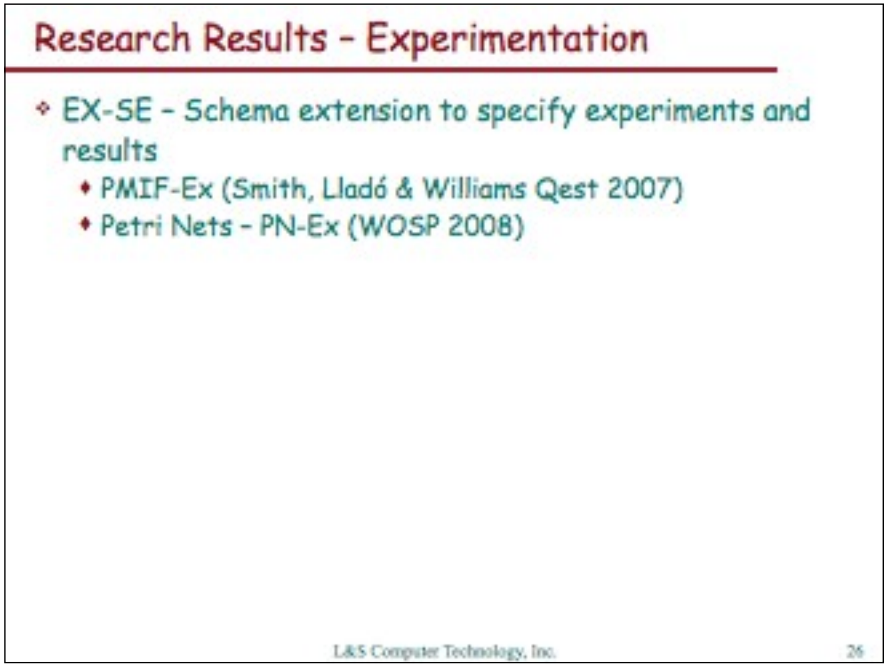
- Now possible to transfer among various performance modeling tools:
  - PMIF - Queueing network models
  - LQN
  - UML
  - Petri nets
- Allows diverse tools to exchange information IF
  - They provide an import and export interface
  - OR they read/write model specifications from/to a file
- Limitation: MIF specifies a model and set of parameters for one run

Part 4: Experiments and Results



Related Work

- Experimentation
  - Hillston (Performance Evaluation '95)
    - IMSE Experimenter (Integrated Modeling Support Environment)
    - Experimental plan+output+analysis specification
    - Calls for Reporter tool to collate results and create reports, but no documentation of details
  - SPEX (Software Performance Experimenter), TR '97
    - Tool for managing performance studies using LQN models
    - Addresses output but not results
  - Our work is a framework for producing results from output (rather than one specific tool)



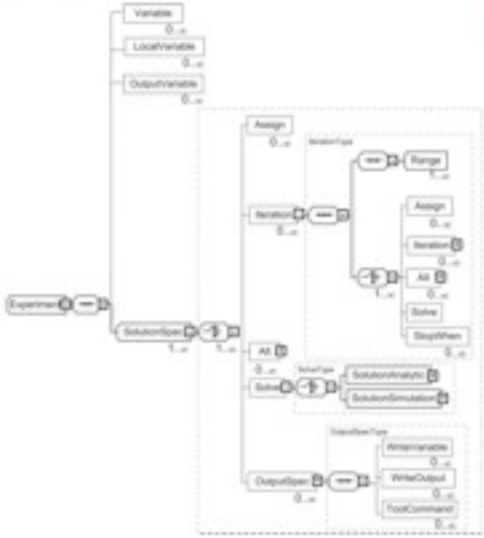
Research Results – Experimentation

- EX-SE – Schema extension to specify experiments and results
  - PMIF-Ex (Smith, Lladó & Williams Qest 2007)
  - Petri Nets – PN-Ex (WOSP 2008)

Experiment Schema Extension (Ex-SE)

- Defines a set of model runs and the desired output
- Specify performance studies independent of a given tool paradigm
- Illustrated with an instance of the Ex-SE for PMIF: PMIF-Ex
  - Applies to other modeling paradigms,
  - Can be used in stand-alone mode,
  - Can specify measurements as well as model studies.
  - Developed experiment examples,
  - Implemented a prototype
  - Demonstrated use with other model paradigms

Schema



- Include Experiment schema into host schema (e.g., PMIF)
- Specialize for terminology and attributes to change

Example Experiment Specification

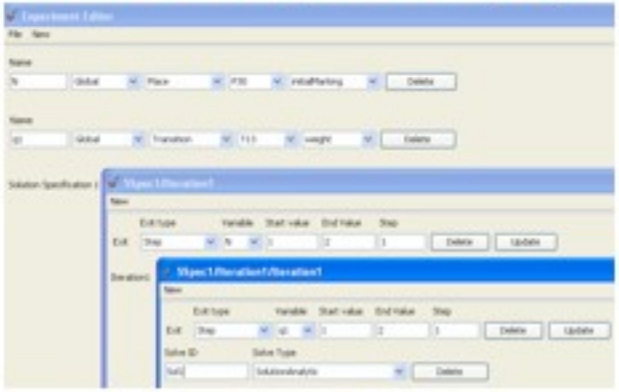
```
<Assign VariableName="SchedType" Value="PS" />
<Iteration>
   <Range VariableName="NForms" Start="18" End="36" Step="9" />
   <Range VariableName="NApply" Start="32" End="64" Step="16" />
  <Range VariableName="NStore" Start="50" End="100" Step="25" />
  <Range VariableName="NConvert" Start="30" End="60" Step="15" />
  <StopWhen Test="UCPU &gt;= 99 or SumTFormTApp &lt;=TotTput" />
  <Assign VariableName="SumTFormTApp" Value="TotTput"/>
  <Solve SolutionID="RunPSAnalytic">
   <SolutionAnalytic/>
  </Solve>
  <Assign VariableName="SchedType" Value="FCFS" />
  <Solve SolutionID="RunFCFSSim" >
   <SolutionSimulation StartInterval="20000" StopTime="100000" />
  </Solve>
```

Compatibility

- Append the ExperimentType to the host schema, and change Variable type specifications to match
  - E.g., define a variable for WorkloadName and Attribute to change its ArrivalRate; attribute name must match host schema
- Examples of compatible schemas:
  - S-PMIF - software performance model interchange format
  - LQN - layered queueing network XML definition
  - GPMIF - performance model interchange format compatible with Reverse Component Agent Theory (RCAT)
  - PNML - Petri Net Markup Language
  - eDSPN - Petri Net interchange format, used by TimeNet
- Could be used with measurement experimenter, e.g., DECALS

Validation: Published Model Study

- Jain experiment demonstrated:
  - Typical experiments can be specified and evaluated.
  - Value of comparing results
  - An automated comparison of multiple experiments is useful



Name: N | Global | Place | P30 | initialMarking

Name: q1 | Global | Transition | T11 | weight

Solution Specification :

	Exit type	Variable	Start value	End Value	Step
Exit	Step	N	1	2	1

Iteration1:

	Exit type	Variable	Start value	End Value	Step
Exit	Step	q1	1	2	1

Solve ID: Sol1 | Solve Type: SolutionAnalytic

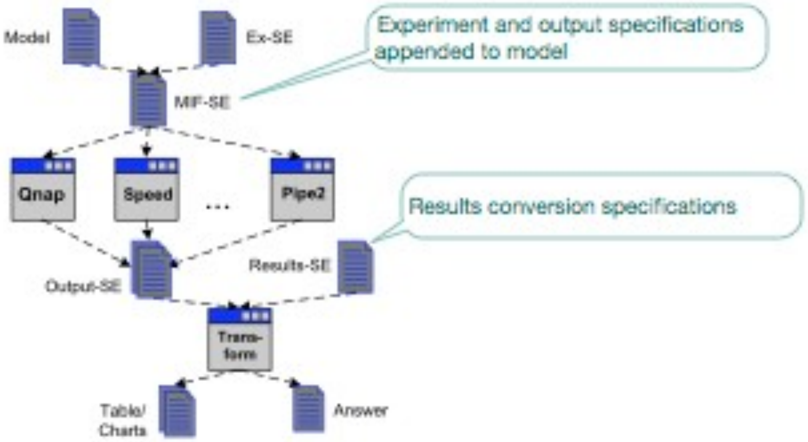
Assessment – Output -> Results

- Performance modeling tools produce numerical data
  - Output: Response times, utilizations, throughput, queue lengths, etc.
  - Users need a meaningful view/report of results
- Identified performance modeling Use Cases
- Surveyed output and results used in practice
  - Typical tables, charts
- Developed modeling-paradigm independent schema
  - Prototype implementation for QN
  - Proof of concept

Approach

- Produce tables and charts for publication and presentation
- Streamline specification of common results
- Allow for creation and update
- Xls (Excel and OpenOffice) and LaTex formats
- Allow for easy extension
- Visualization techniques are evolving
  - Include tool output reports with ToolCommand in the experiment specification
  - Interoperability with visualization tools

Overview of Extensions

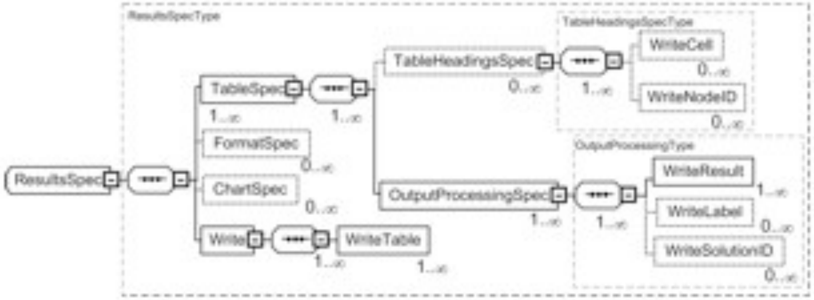


Experiment and output specifications appended to model

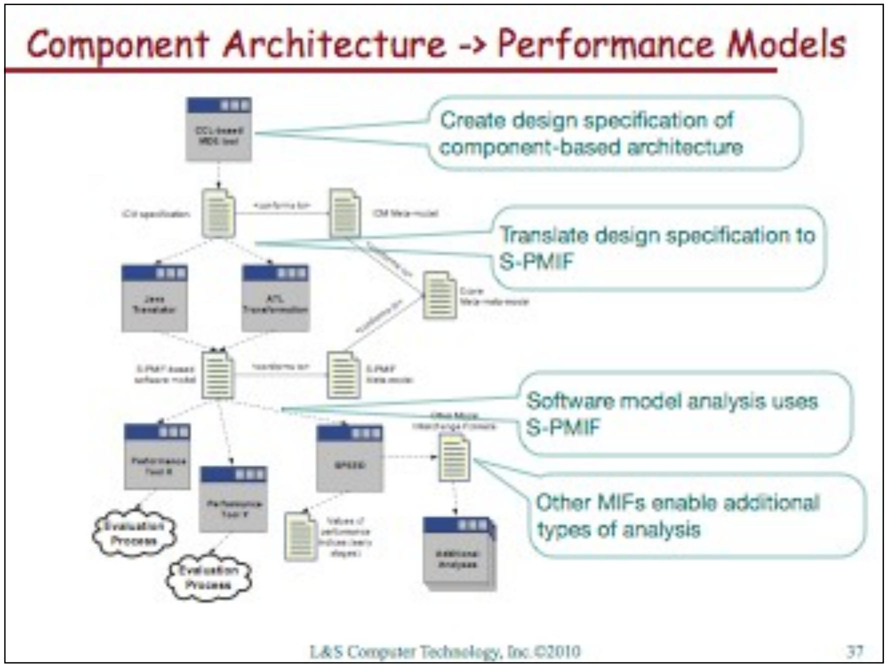
Results conversion specifications

Model Transformation Approach

- Results Schema Extension (Results-SE)



Part 5: Real-time and component based systems



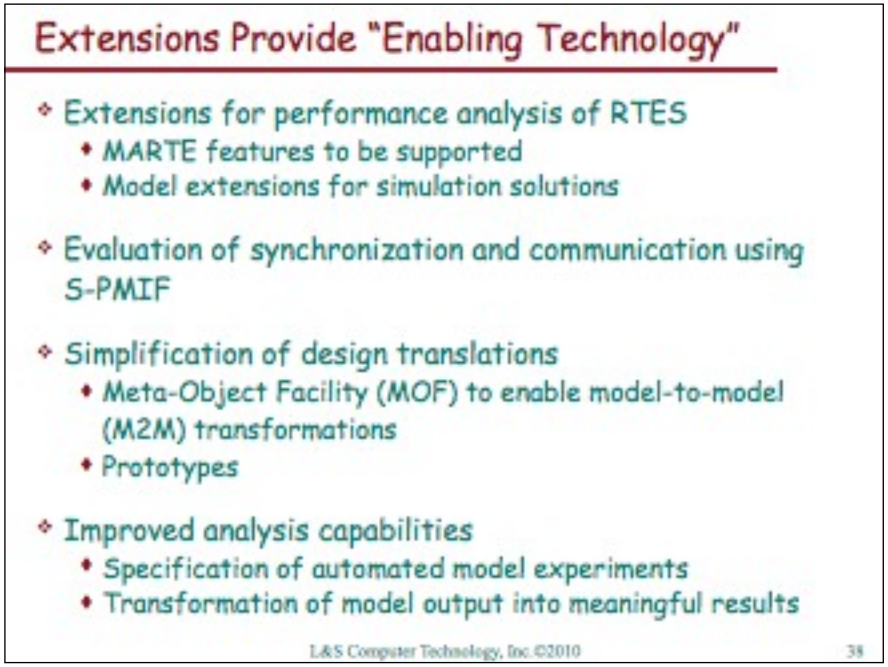
Component Architecture -> Performance Models

Create design specification of component-based architecture

Translate design specification to S-PMIF

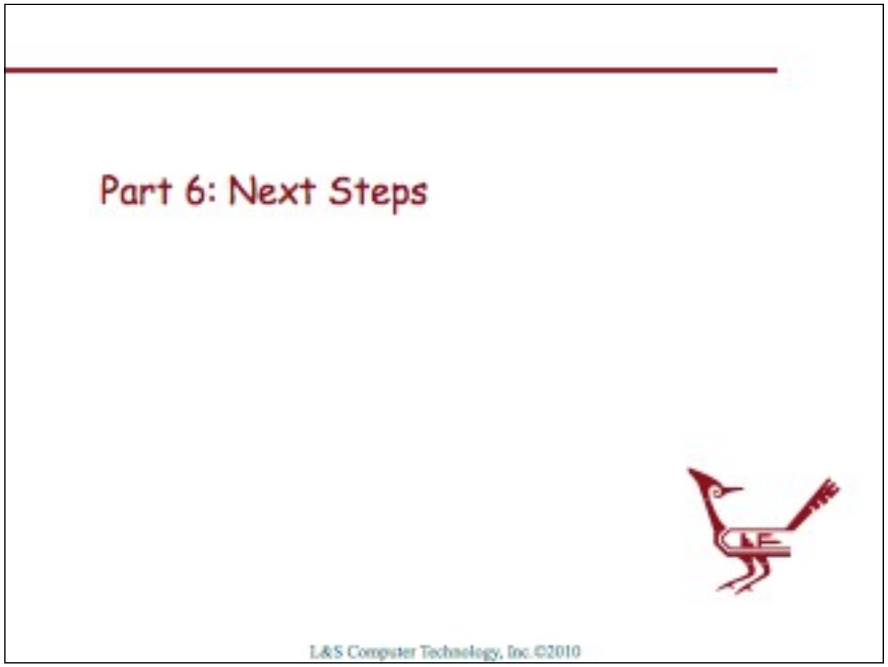
Software model analysis uses S-PMIF

Other MIFs enable additional types of analysis

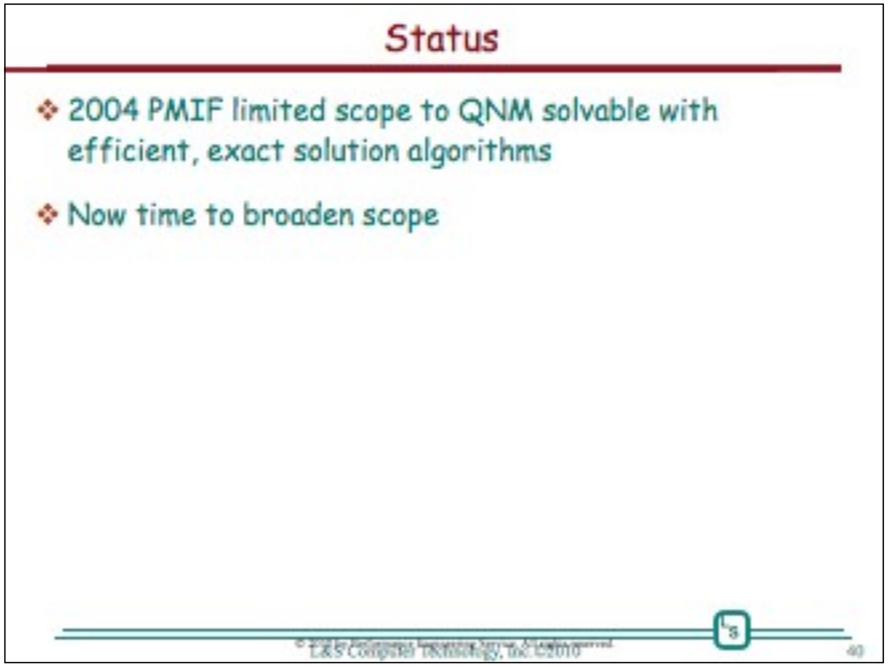


Extensions Provide "Enabling Technology"

- Extensions for performance analysis of RTES
  - MARTE features to be supported
  - Model extensions for simulation solutions
- Evaluation of synchronization and communication using S-PMIF
- Simplification of design translations
  - Meta-Object Facility (MOF) to enable model-to-model (M2M) transformations
  - Prototypes
- Improved analysis capabilities
  - Specification of automated model experiments
  - Transformation of model output into meaningful results



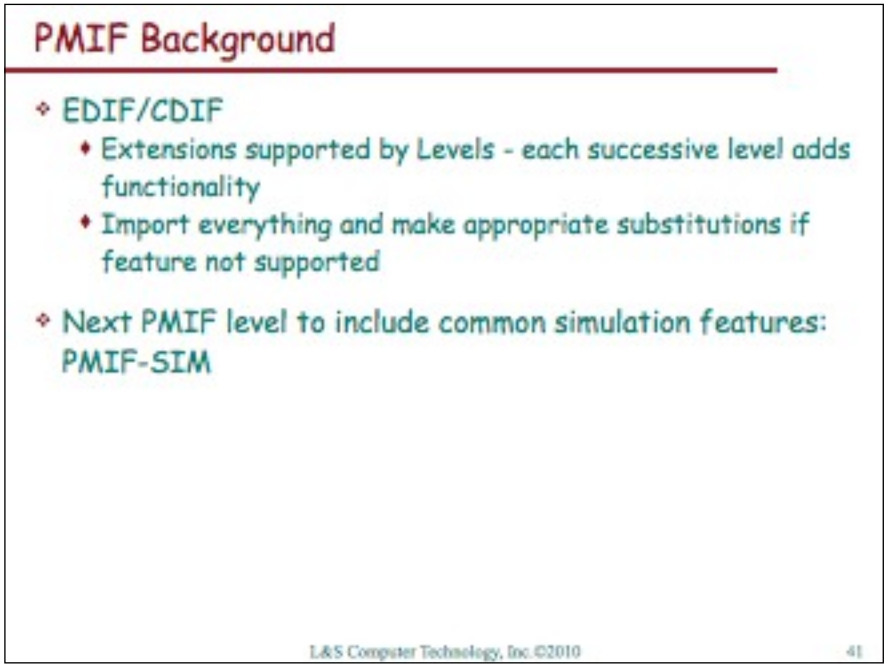
Part 6: Next Steps



Status

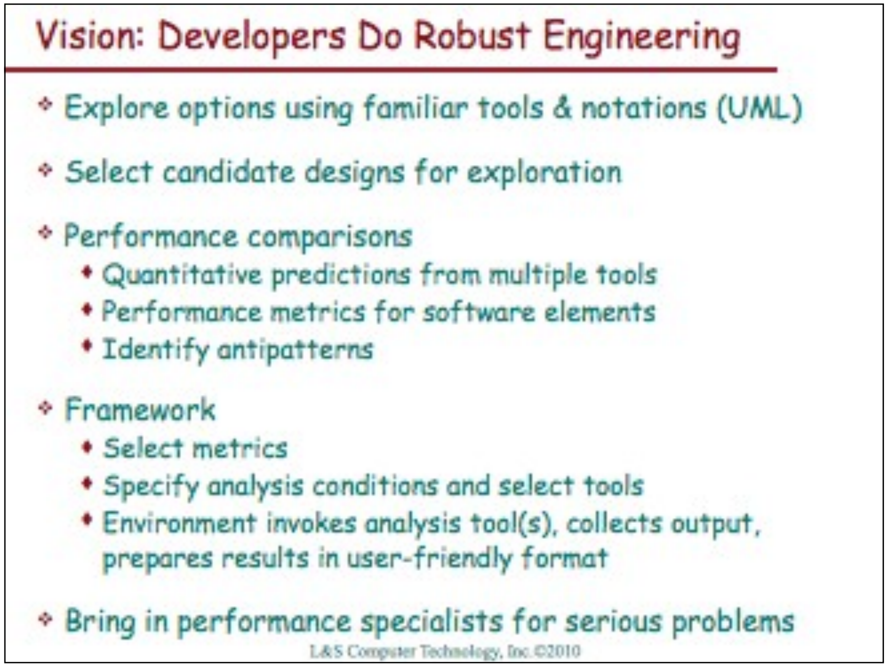
- 2004 PMIF limited scope to QNM solvable with efficient, exact solution algorithms
- Now time to broaden scope

© 2010 by Performance Engineering Services. All rights reserved.



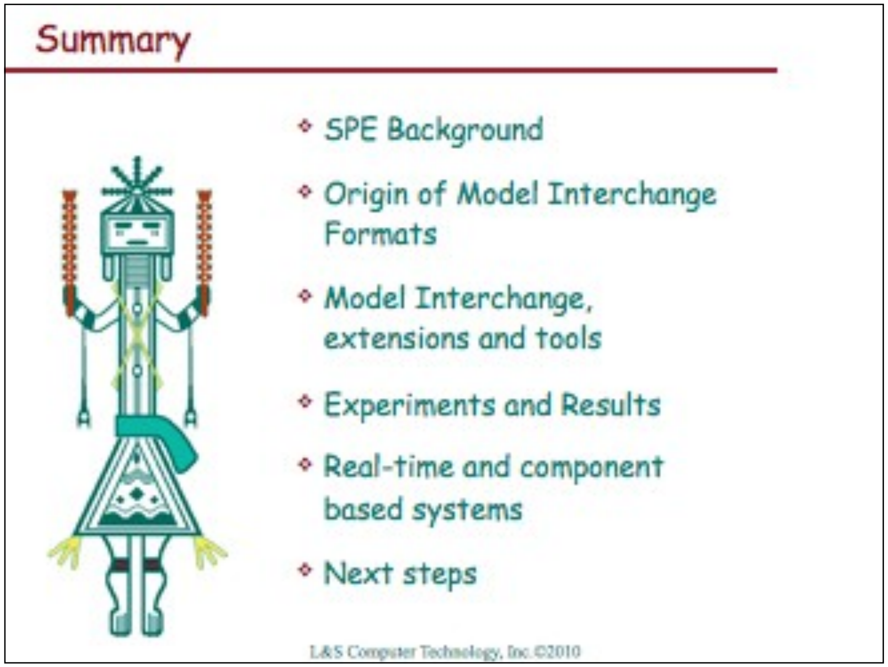
PMIF Background

- EDIF/CDIF
  - Extensions supported by Levels - each successive level adds functionality
  - Import everything and make appropriate substitutions if feature not supported
- Next PMIF level to include common simulation features: PMIF-SIM



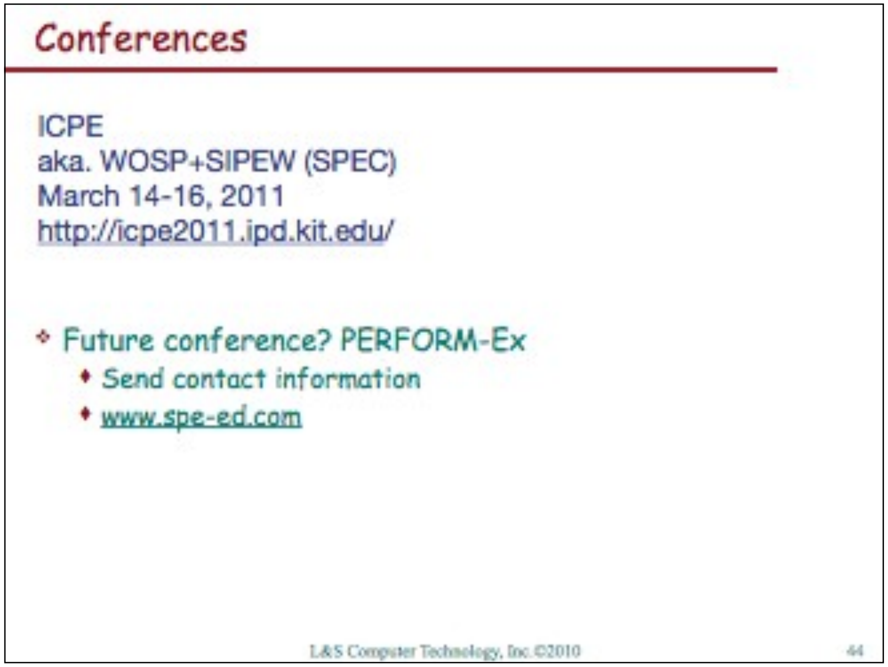
Vision: Developers Do Robust Engineering

- Explore options using familiar tools & notations (UML)
- Select candidate designs for exploration
- Performance comparisons
  - Quantitative predictions from multiple tools
  - Performance metrics for software elements
  - Identify antipatterns
- Framework
  - Select metrics
  - Specify analysis conditions and select tools
  - Environment invokes analysis tool(s), collects output, prepares results in user-friendly format
- Bring in performance specialists for serious problems



Summary

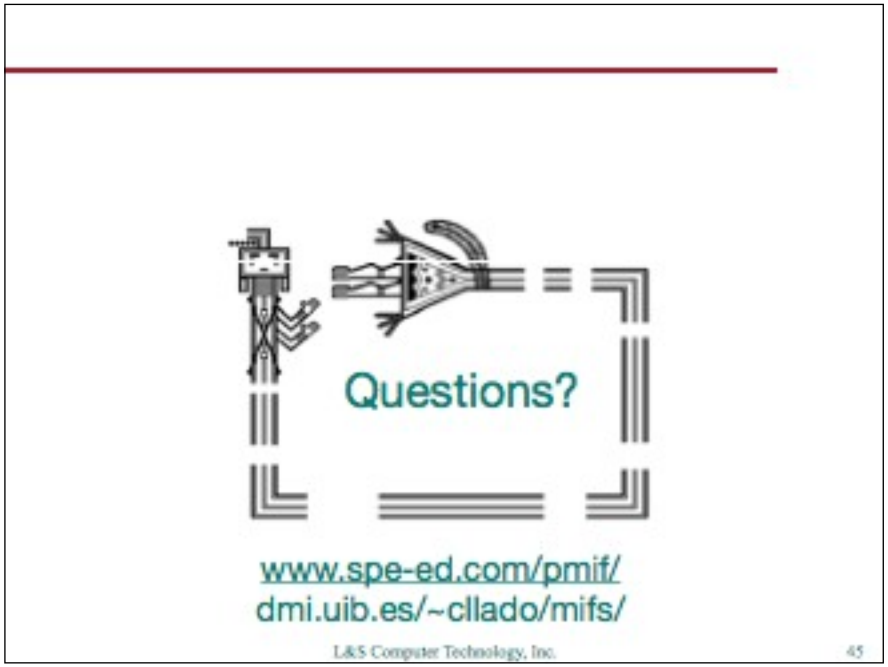
- SPE Background
- Origin of Model Interchange Formats
- Model Interchange, extensions and tools
- Experiments and Results
- Real-time and component based systems
- Next steps



Conferences

ICPE
aka. WOSP+SIPEW (SPEC)
March 14-16, 2011
http://icpe2011.ipd.kit.edu/

- Future conference? PERFORM-Ex
  - Send contact information
  - www.spe-ed.com



Questions?

www.spe-ed.com/pmif/
dmi.uib.es/~cllado/mifs/