Performance Engineering of SDL/MSC Systems

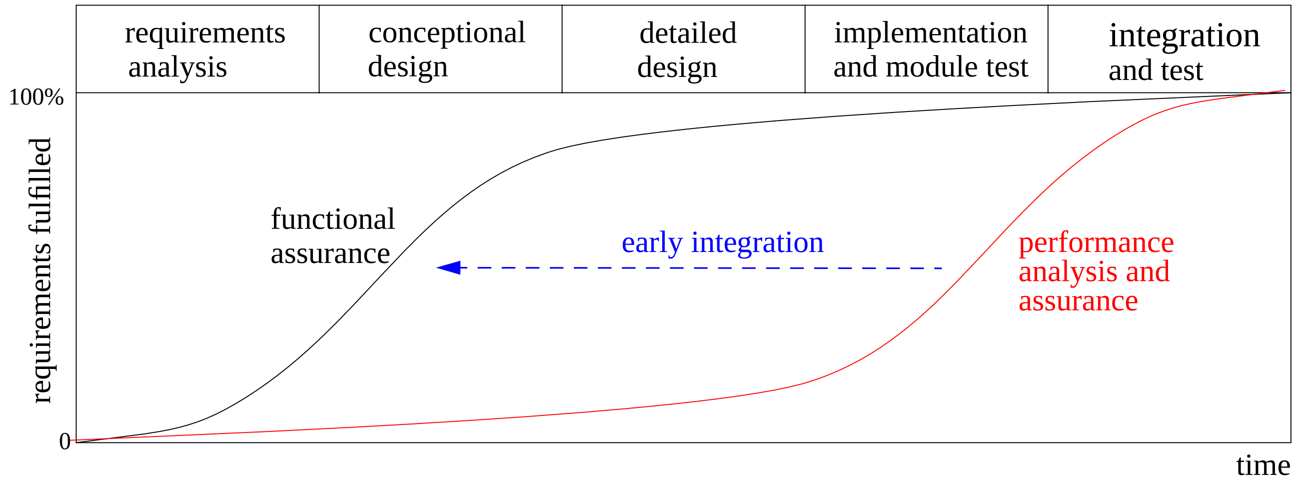
Andreas Mitschele-Thiel
Universität Erlangen-Nürnberg
mitsch@informatik.uni-erlangen.de
http://www7.informatik.uni-erlangen.de/~mitsch

Bruno Müller-Clostermann
Universität GH Essen
bmc@informatik.uni-essen.de
http://www.cs.uni-essen.de/Fachgebiete/SysMod/

1. Principles and Techniques of Performance Engineering
2. Introduction into SDL and MSC
3. SDL/MSC-Based Performance Engineering
4. Tools for SDL- and MSC-Based Performance Engineering
5. Concluding Remarks
6. Further Readings

1. Principles and Techniques of Performance Engineering

Problem Statement



late consideration of performance aspects

- **late detection and correction of performance problems**
  - **high cost for redesign**
  - **gradual destruction of system architecture**

1. Principles and Techniques of Performance Engineering

Performance Engineering - What’s that?

Some Definitions:

- Methods and techniques to effectively derive efficient systems
- Integration of performance issues in the systems engineering process

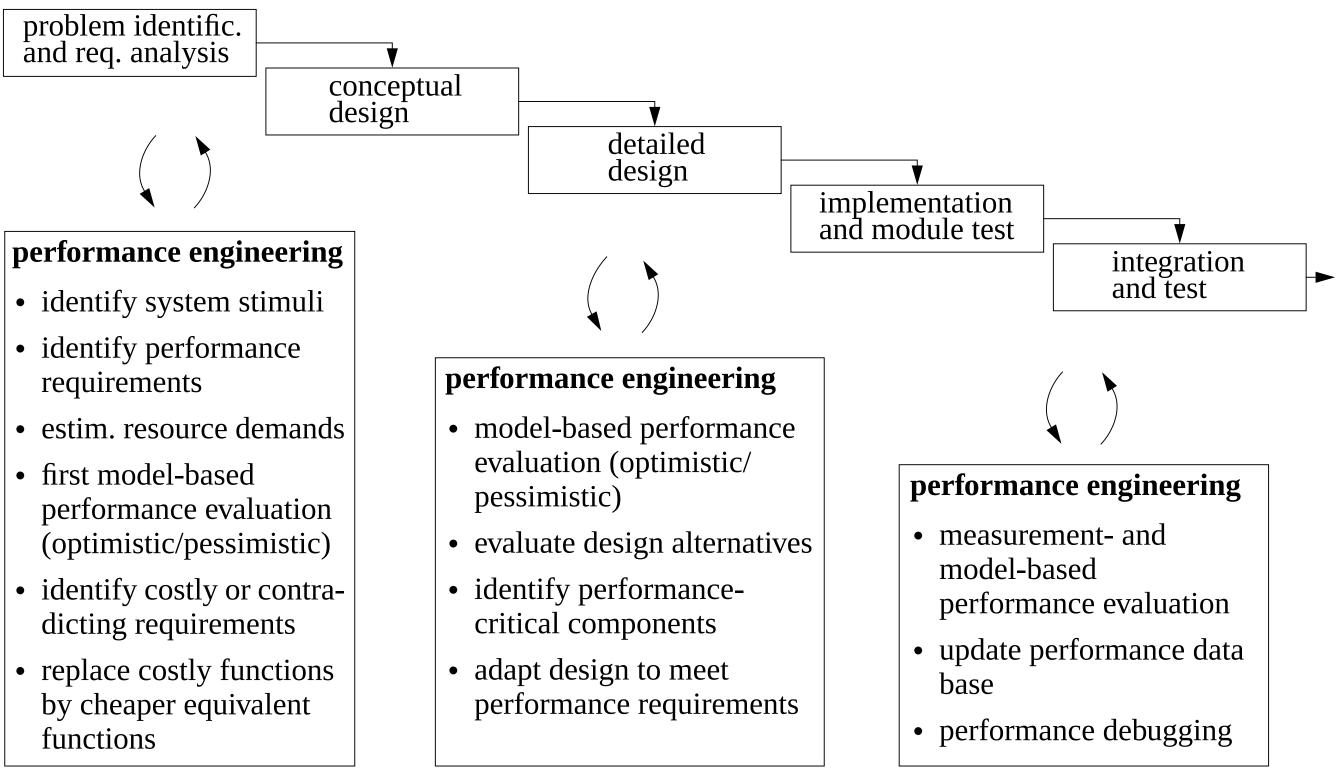
Two Important Subtasks of Performance Engineering:

- derive performance measures (performance evaluation)
  - performance modelling
  - performance measurements
- control the systems engineering process to develop efficient and cost-effective systems
  - identify performance-critical parts
  - deal with performance critical-parts appropriately (design and implementation)

Literature: Connie Smith, Performance Engineering of Software Systems, Addison Wesley, 1990

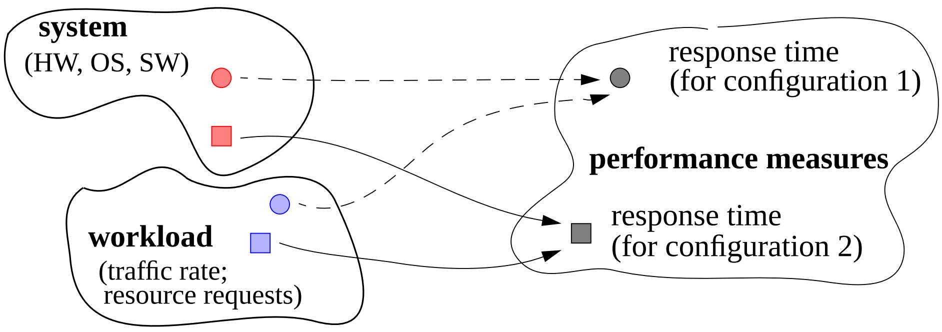
1. Principles and Techniques of Performance Engineering

Performance Engineering in the Development Process



1. Principles and Techniques of Performance Engineering

Performance Measures as a Function of ‘System’ and ‘Workload’



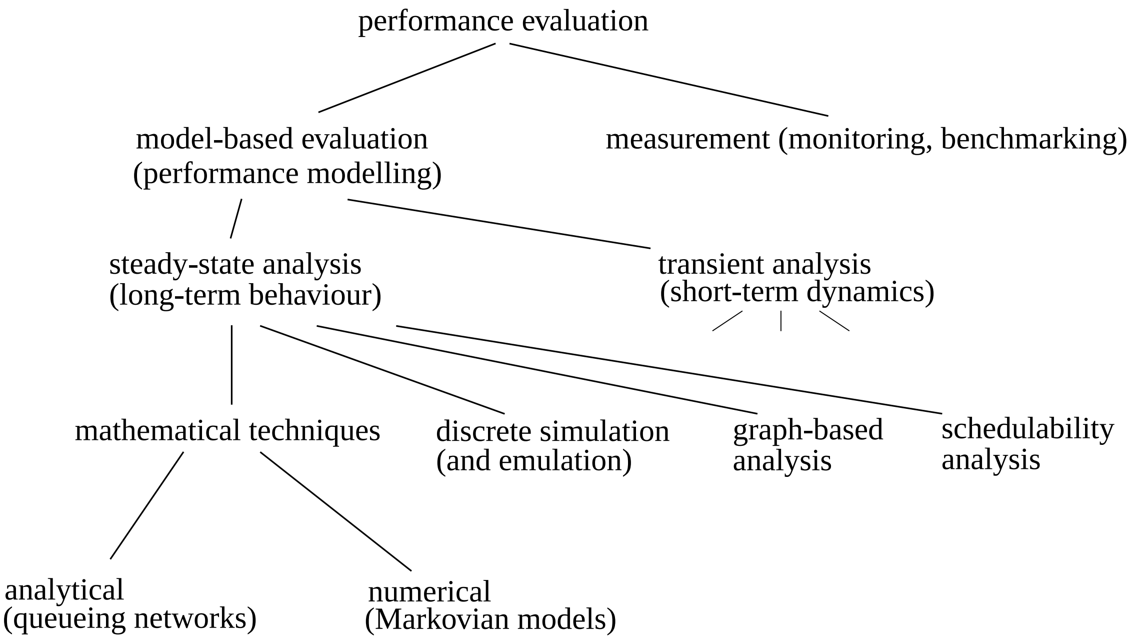
We have an abstract function

f(system, workload) --> performance measures

- system may include hardware, operating system, overhead and application software
- workload includes traffic characteristic (external stimuli) and resource requests
- system and workload may be real objects or abstract descriptions
- function f is derived employing techniques and tools from the field of performance evaluation

1. Principles and Techniques of Performance Engineering

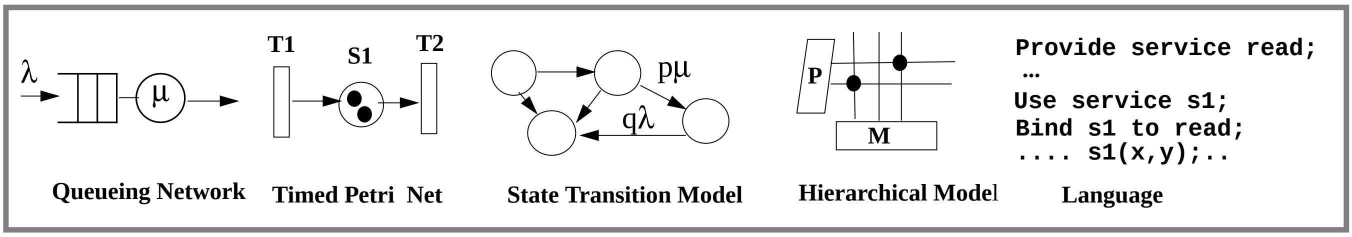
A Classification of Performance Evaluation Techniques



Literature: Raj Jain, The Art of Computer System Performance Analysis, Wiley, 1991

1. Principles and Techniques of Performance Engineering

Paradigms and Notions for Specification of Performance Models



Transformation from high level specification to low level solution techniques

Back transformation of results

Techniques for Model Solution

Analytical Techniques	Numerical Techniques	Discrete Simulation
mean value analysis, product form techniques Prob $(n_1, ..., n_M)=\Pi P(n_i)$, "fast" solutions	Numerical Solution of Markov Chains, i.e. solve global balance equations $pQ=0$, $\Sigma p_i=1$	`While not EventList.empty and RunConditionsHold do ... excute current ... ... schedule new events .. ... update statistics ... ....`

1. Principles and Techniques of Performance Engineering

Model Input

1. Devices/stations and their characteristics (including HW, OS, Application, Overhead)
2. Workload intensity (# stimuli/time, traffic arrival rate)
3. Workload type (eg. interactive)
4. Demands at the resources (type and amount of work to be done)

Solution Techniques

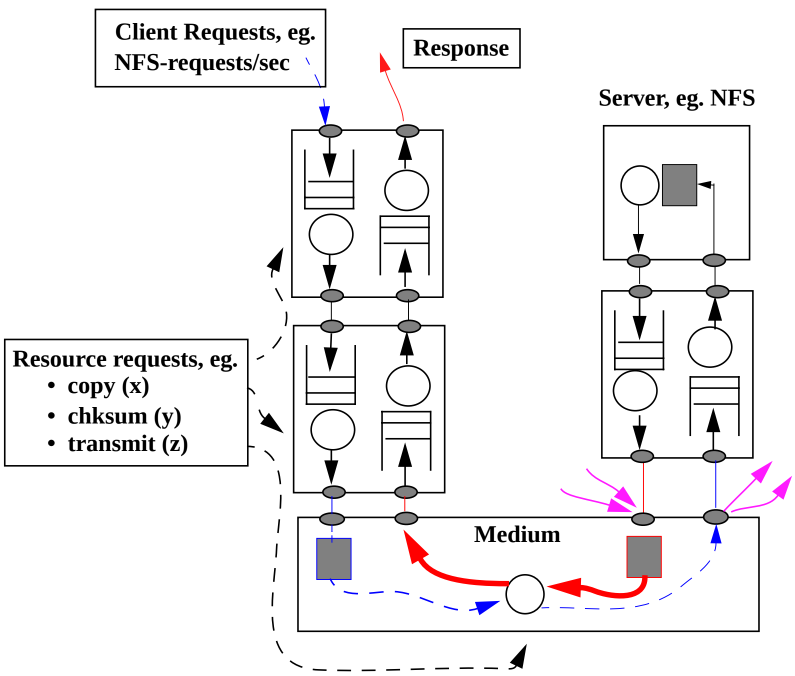
Analytical, numerical or discrete simulation depending on ....

5. level of detail,
6. required performance measures,
7. available tools, and more ....

Model Output

8. Throughputs (KB/sec, NFS-requests/sec)
9. Utilization of medium and server
10. Response times, Wait time
11. Queue lengths

Example: Queueing Network Model of a C/S-System



1. Principles and Techniques of Performance Engineering

Steps of Performance Modelling

1. Understand the object of your investigation, either existing or under design, as well as possible (hard task).
2. Predict the workload imposed on the system and build a workload model (very hard task).
3. Build a performance model, i.e. map your “mind model” into a meaningful or “equivalent” performance model (hard task).
4. Transform the performance model to an executable assessable model (can be done automatically).
5. Execute/analyse the model and derive performance measures (easy task, automatic).
6. Check whether your performance measures do meet your performance goals.
7. Modify your system design and restart.

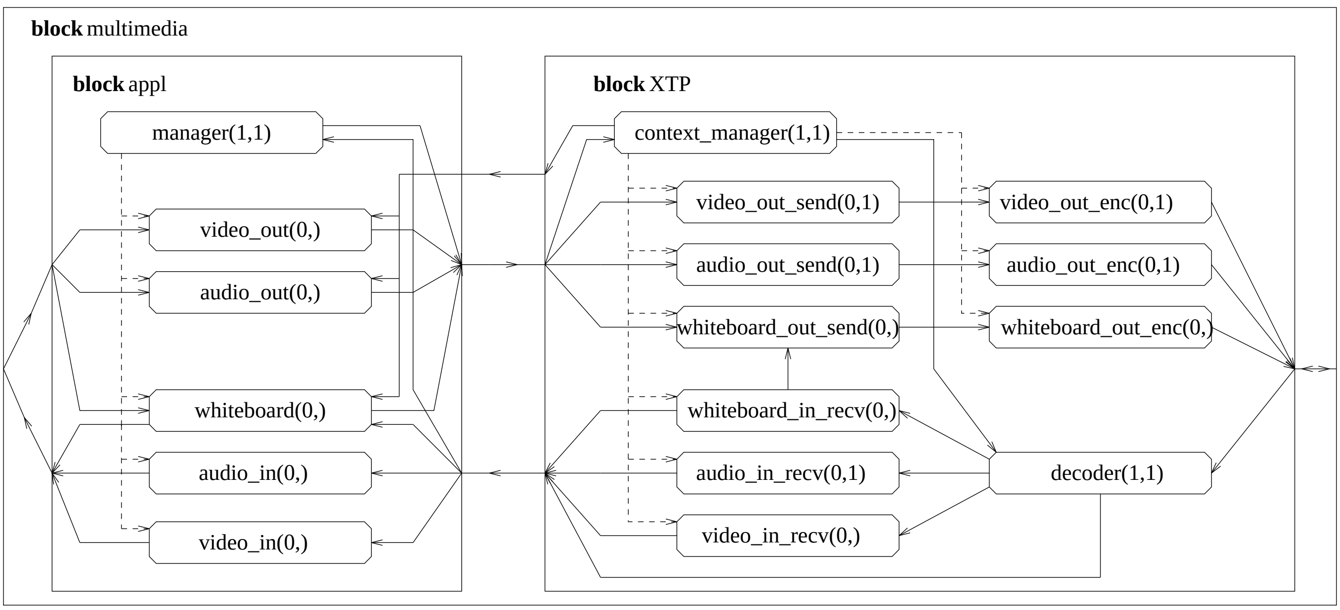
Preview with respect to the SDL/MSC context

- Step 1 is supported by SDL/MSC.
- Step 2 is supported by MSC (use cases) and SDL signal list.
- Step 3 is supported by SDL/MSC methodology including implementation design.

2. Introduction into SDL and MSC

Specification and Description Language (SDL, ITU-T Z.100)

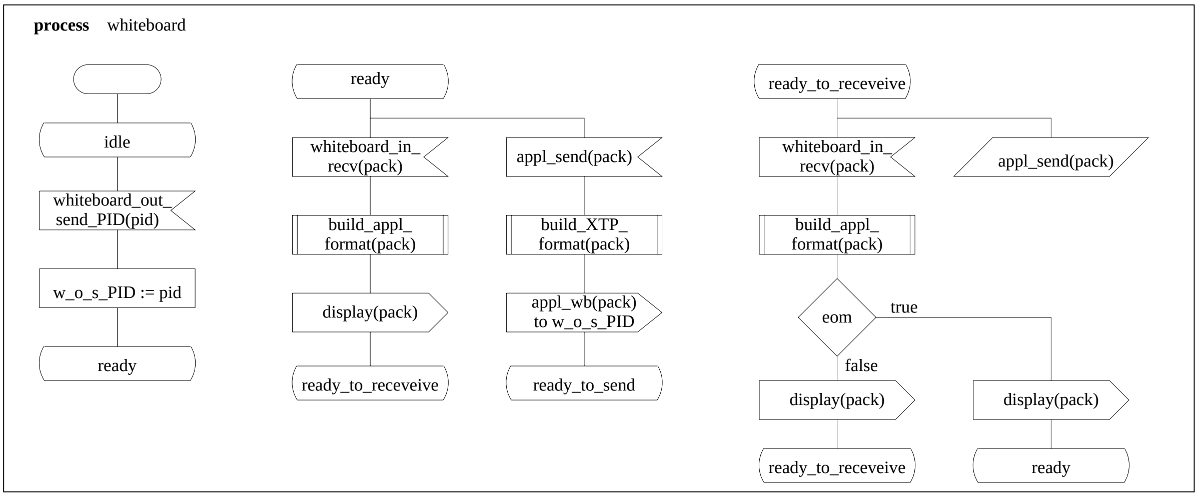
Block Diagram



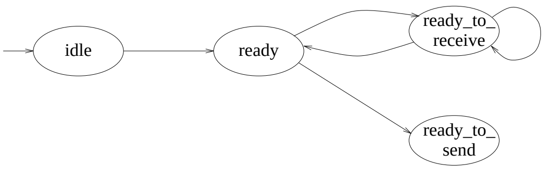
2. Introduction into SDL and MSC

Specification and Description Language (SDL)

Process Diagram



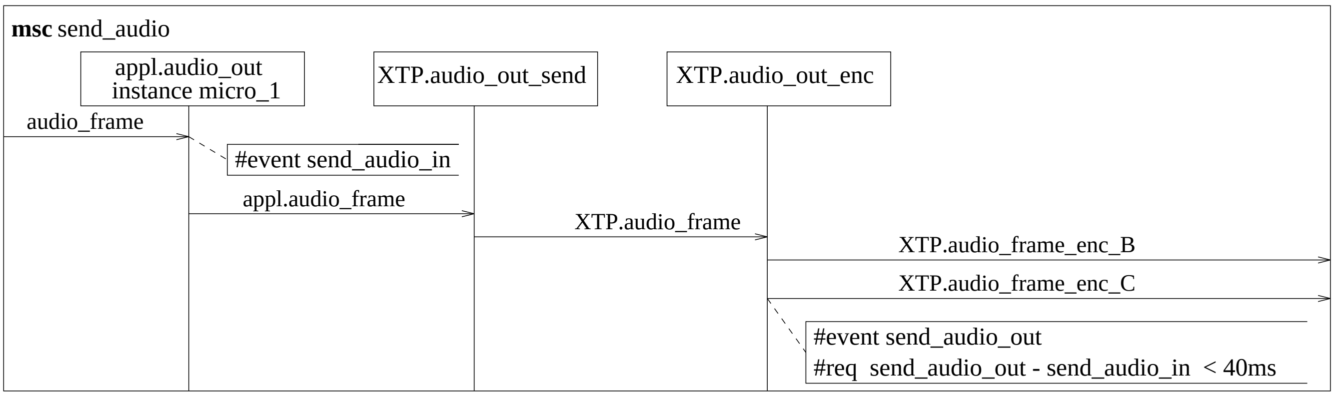
FSM Description



2. Introduction into SDL and MSC

Message Sequence Chart (MSC, ITU-T Z.120)

Sequence Diagram

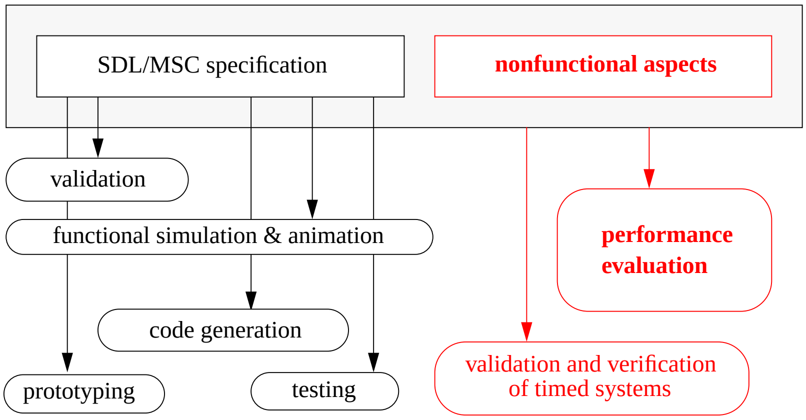


3. SDL/MSC-Based Performance Engineering

Purpose of SDL/MSC

A system specified with standard SDL/MSC may serve as a basis for

- verification and validation,
- functional simulation and animation,
- code generation, prototyping,
- testing, and more.



3. SDL/MSC-Based Performance Engineering

Additional Information Needed for SDL/MSC-Based Performance Engineering

SDL and MSC

- do not cover nonfunctional aspects (time, delay, cost)
- abstract from implementation details (model ideal world, not real physics with delays, limited resources and errors)

Information not covered by standard SDL and MSC

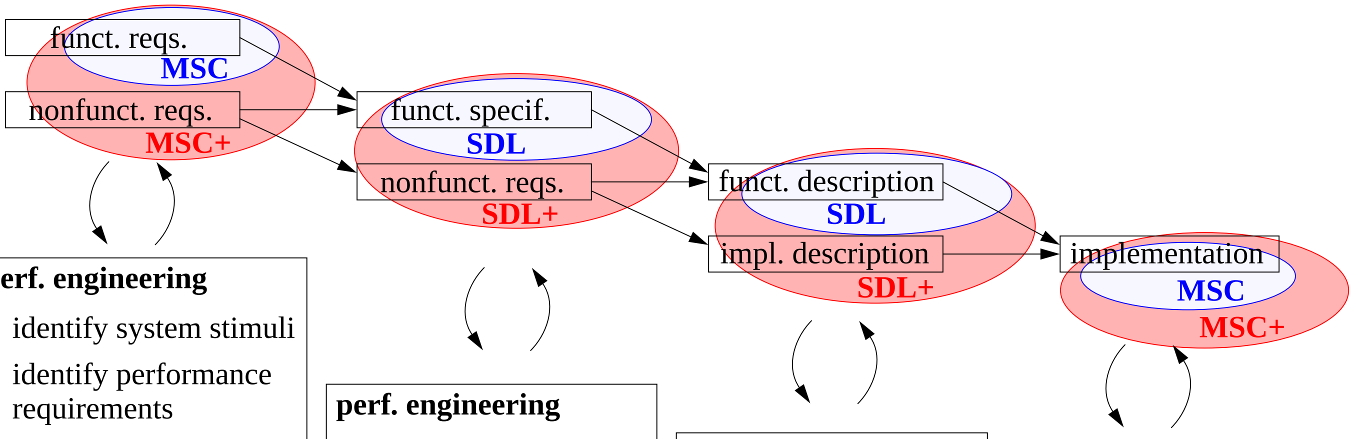
- system stimuli (arrival process)
- available (limited) resources: processors, links, memory, strategies to handle contention
- resource demands: computation and communication cost, memory
- implementation decisions: SW configuration (code generation strategy, etc.), mapping on hardware, HW/SW partitioning
- performance sensors
- performance requirements

Prerequisites for SDL/MSC-based Performance Engineering

- identification and formal description of missing information
- association of information with functional information given by SDL/MSC specification (question: where and how to specify the added information?)

3. SDL/MSC-Based Performance Engineering

Performance Engineering with SDL/MSC



perf. engineering
- identify system stimuli
- identify performance requirements
- est. resource demands
- first model-based performance evaluation (optimistic/pessimistic)
- identify costly or contradicting reqs.
- replace costly functions by cheaper equivalent functions

perf. engineering
- model-based performance evaluation (optimistic/pessimistic)
- evaluate design alternatives
- identify performance-critical components
- adapt design to meet performance reqs.

perf. engineering
- detailed model-based performance evaluation (optimistic/pessimistic)
- evaluate implementation alternatives
- adapt implementation to meet perf. reqs.

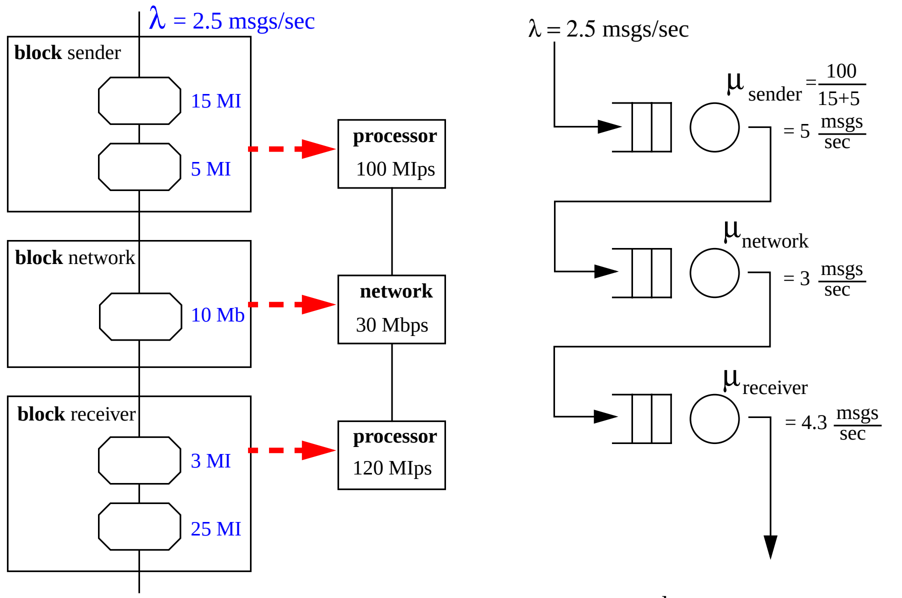
perf. engineering
- measurement- and model-based perf. evaluation
- update performance data base
- performance debugging

3. SDL/MSC-Based Performance Engineering

Performance Modelling with SDL/MSC

Example 1: **Queueing Model Derived from an SDL Specification**

SDL specification & workload & machines & mapping = performance model



$$\text{processor service rate } \mu = \frac{\text{processor speed}}{\text{processor load per msg}}$$

Analytical queueing network analysis provides mean values for
- delays, utilizations,
- queue lengths and population

for certain model classes, here:

utilization $\rho_{network}$ = 83,3%
total delay T = 2.96 sec
total population N = 7.4 msgs

Restrictions are:
- no general distributions,
- no synchronizations,
- no blocking, etc.

For these cases,
- **numerical Markov analysis** or
- **discrete simulation**

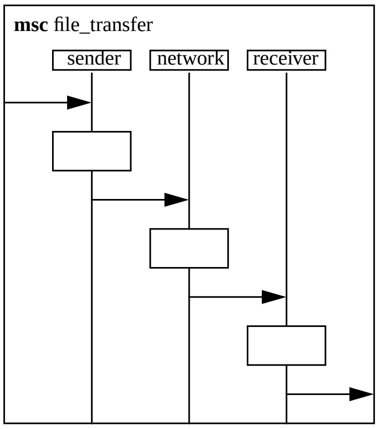
may be applied.

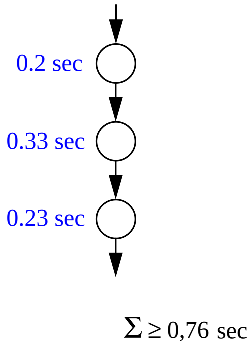
3. SDL/MSC-Based Performance Engineering

Performance Modelling with SDL/MSC (cont’d)

Example 2: Critical Path Analysis with MSC

MSC specification **&** **workload (resource reqs.)** **=** **performance model**





Simple task graph analysis provides

- response time (optimistic case) for different workload scenarios (MSCs)
- with deterministic or exponential service times

More detailed analysis of superimposed workloads with

- **derivation and analysis of schedule**
- **schedulability analysis**
- **simulation of a set of MSCs (pessimistic case)**

Extensions to the simple performance model:

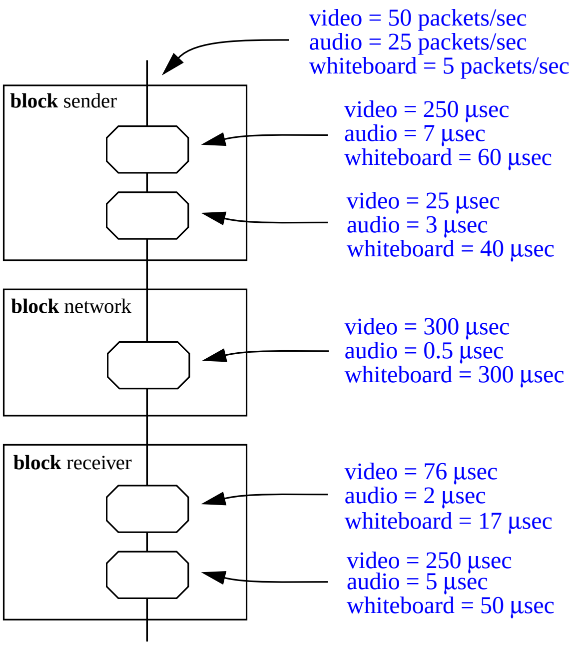
- add mapping on resources and resulting sequentialisation constraints
- add scheduling strategy and superimpose several workload scenarios

3. SDL/MSC-Based Performance Engineering

Performance Modelling with SDL/MSC (cont’d)

Example 3: Bottleneck Analysis with Process Graph Derived from SDL Specification

SDL specification **&** **workload** **(& machines & mapping) =** **performance model (load per resource or per SDL unit)**



load of block sender
= video + audio + whiteboard
= 50 * (250 + 25)
+ 25 * (7 + 3)
+ 5 * (60 + 40) µsec
= 14.5 msec

Process graph analysis provides

- load imposed on the resources (optimistic/ pessimistic)
- load of SDL processes, blocks, channels, etc.
- identification of performance-critical workload scenarios
- identification of bottleneck servers or heavily loaded SDL units

4. Tools for SDL- and MSC-based Performance Engineering

Overview of Approaches

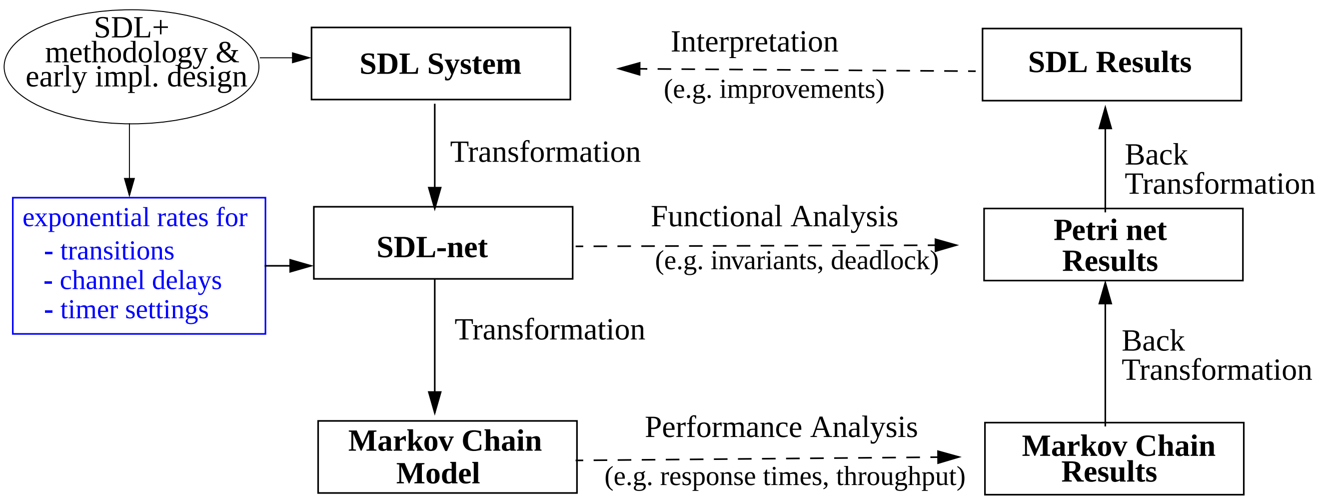
- Petri Nets Approach: SDL to **SDL-net**, SDL to Queueing Petri Nets (QPN)
- Defining blocks as executing machines: **SPECS** (SDL Performance Evaluation of Concurrent Systems)
- Execution of generated code on emulated target hardware (**SPEET**: SDL Performance Evaluation Tool)
- Mapping SDL to full-fledged performance evaluation environments: (**HIT**, **OPNET**-Modeller)
- Mapping resource requests to machines: **QUEST** and the language Queueing SDL (**QSDL**)
- Coupling of SDL specification with simulation tool (**Easy-Sim**: Geode-SDL and SES Workbench)
- MSC-based performance evaluation and optimization (**DO-IT** Toolbox, HW/SW-Codesign Project)
- Building LQN performance models from traces (**Model Builder**)

Classification of Approaches

Specification of Performance Aspects			Performance Modelling and Analysis
What?	**Where?**	**How?**	**Which Technique?**
system stimuli	SDL	annotational (by comments)	analytic queuing networks
resource demands (time durations)	MSC	language extension	Petri nets (numeric or simulation)
machine	SDL/MSC	implicitly	general simulation model
mapping	separately		code derived from specification
perf. requirements			graph model (task/program graph)
			coupling with simulation tool

4. Tools: SDL-net - Mapping SDL to Stochastic Petri Nets

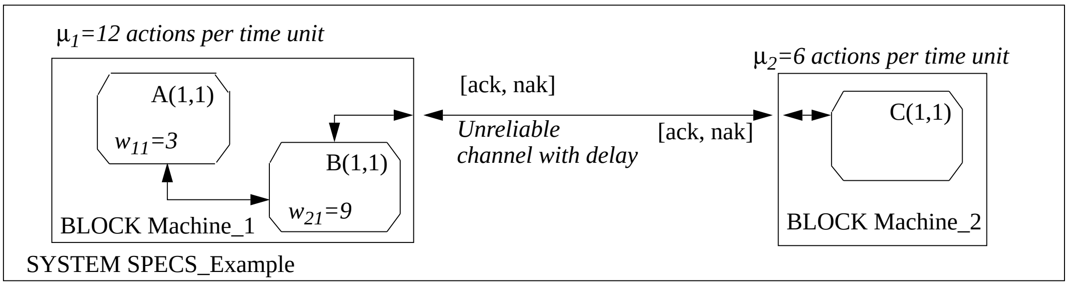
Specification of Performance Aspects			Performance Modelling and Analysis
What?	**Where?**	**How?**	**Which Technique?**
time durations (exponential)	separately	Petri net level	Petri nets & numerical Markov analysis



Literature: H. M. Kabutz, Doctoral Thesis, University of Cape Town, 1997

4. Tools: SPECS - SDL Performance Evaluation of Concurrent Systems

Specification of Performance Aspects			Performance Modelling and Analysis
What?	**Where?**	**How?**	**Which Technique?**
machines=blocks workload=weights channel delays & reliability	separately	GUI to specify and construct the environment	code derived from specification is executed /simulated on a "virtual machine"



- Processes A and B in the same block execute in a multitasked way according to their weights w_{11} and w_{12}

Literature: M. Bütow, M. Mestern, C. Schapiro, P.S. Kritzinger: Perf. Modelling with SDL, FORTE/PSTV '96

4. Tools: Hardware emulation with SPEET

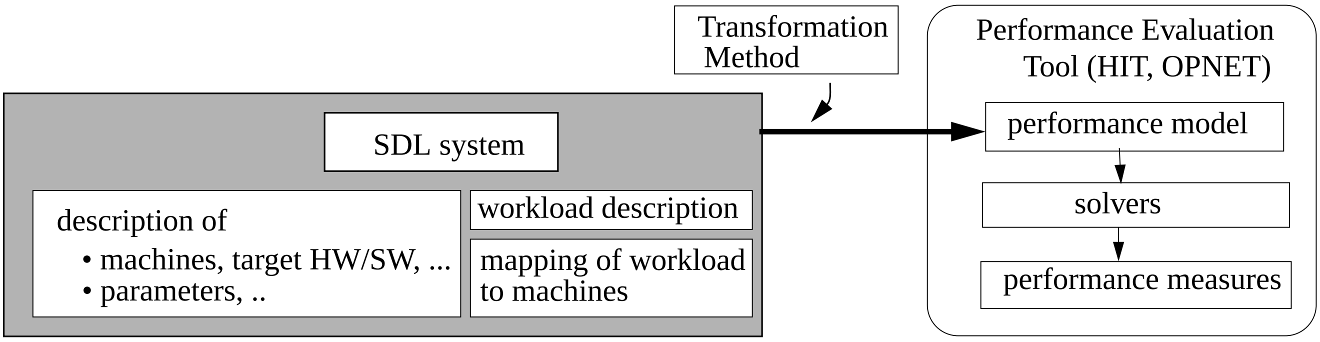
Specification of Performance Aspects			Performance Modelling and Analysis
What?	**Where?**	**How?**	**Which Technique?**
transmission models workload generators →	separately	simulation and emulation environment	- code derived from specification is executed on emulated hardware, - parallel simulation
perf. requirements →	MSC with time constraints		

- Simulation and emulation of several formal specifications at the same time.
- Systems can be triggered by traffic load generators and can be interconnected with transmission links
- Detailed workload models and the exact modelling of (existing) hardware by emulation.

Literature: M. Steppler, M. Lott: SPEET - SDL Performance Evaluation Tool, Proc. SDL Forum '97

4. Tools: Usage of Modeling Environments (HIT and OPNET)

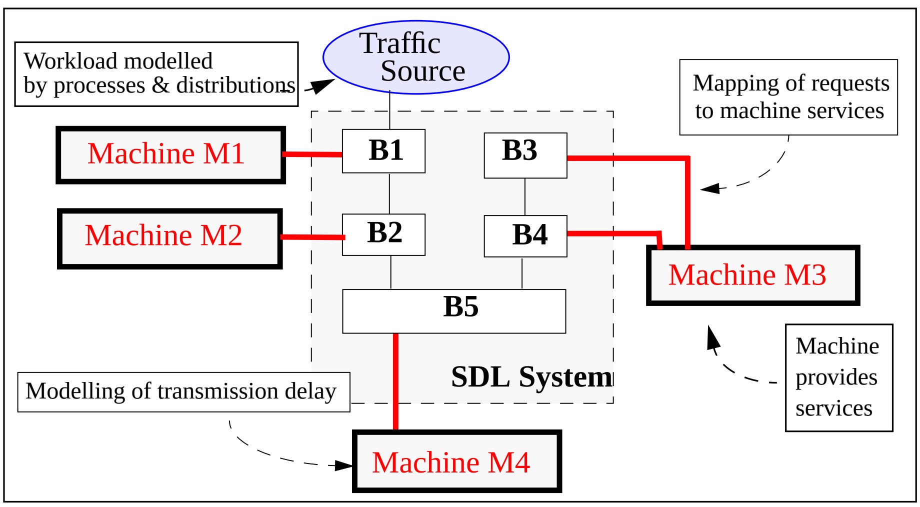
Specification of Performance Aspects			Performance Modelling and Analysis
What?	**Where?**	**How?**	**Which Technique?**
machines workload mapping	separately, then transformed to a performance model (in HIT or OPNET)		techniques supplied by perf. tool (here: discrete simulation)



Literature: E. Heck: The Integration of SDL with HIT, Ph.D. Thesis, Universität Dortmund, Inf. IV, 1996
J. Martins, J.-P. Hubaux, T. Saydam, S. Znatny: Integrating OPNET and SDL, ICC 1996

4. Tools: QUEST and the Extension of SDL to QSDL

Specification of Performance Aspects			Performance Modelling and Analysis
What?	**Where?**	**How?**	**Which Technique?**
machines workload mappings	in SDL, yielding a QSDL-System	annotational “pragmas” for machines & requests	discrete simulation
performance requirements →		temporal logic	verification of timed systems by model checking

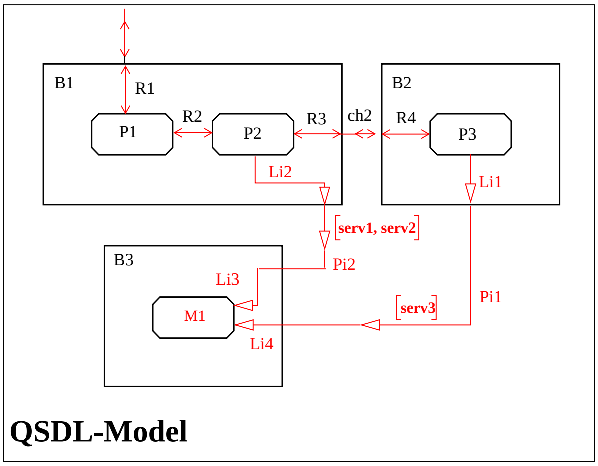


The extension to QSDL
Extensions to describe time and resources are added as comments. QSDL consists of SDL with comments*).

The tool QUEST
QUEST translates QSDL--systems to performance models and excutes simulations or verifies performance requirements

**) Since Oct. 1998 QUEST works with annotations instead of a language extension. This allows simple exchange of QSDL-Models between QUEST and other SDL-tools.*

4. Tools: QUEST and the Extension of SDL to QSDL (cont'd)



Features of QSDL

- **machines providing services**
- **processes request services**
- **links (connecting processes to machines)**
- **pipes (connecting blocks enclosing machines)**
- **enhanced timing (process awake, delayed output)**
- **distribution functions**

```
SYSTEM Example1;
   BLOCK B1 referenced;  /*.... same for B2 and B3 ... */
   /*## MACHINESERVICE serv1, serv2, serv3;  ##*/
   /*## PIPE Pi2 FROM B1 TO B2 WITH serv1, serv2;  ENDPIPE; ##*/
   /*## PIPE Pi1 FROM B2 TO B3 WITH serv3; ENDPIPE; ##*/
ENDSYSTEM Example1;        <-- Defining Pipes between Blocks

BLOCK B1;
   PROCESS P1 REFERENCED; /* ... same for P2 ... */
   /*## LINK Li2 FROM P2 TO ENV WITH serv1, serv2;  ##*/
ENDBLOCK B1;        <-- Defining a Link to the Block Boundary

BLOCK B2; ..... ENDBLOCK B2;

BLOCK B3;
   /*## MACHINE M1 referenced;  ##*/
ENDBLOCK B3;
                                      Definition of Machine M1
/*##     MACHINE M1;          <--
         SERVER  1;                   /* Number of Processors */
         DISCIPLINE FCFS;             /* Service  Strategy */
         OFFERS   serv1 : 0.07,       /* Services and Speeds */
                  serv2 : 47.11,
                  serv3 : 0.815;
         ENDMACHINE M1;               ##*/

PROCESS P2 (1,1);     <-- Request of Service serv1 offered by M1
   /*## REQUEST serv1(some_amount);  ##*/
ENDPROCESS P2;
```

Literature:

M. Diefenbruch, et al.: The QUEST-approach for the Perf. Eval. of SDL-Systems, FORTE/PSTV '96
J. Hintelmann, et al.: Perf. Analysis of TCP´s Flow Control Mechanisms using QSDL, SDL Forum '97

4. Tools: Coupling SDL with the SES Workbench

Specification of Performance Aspects			Performance Modelling and Analysis
What?	**Where?**	**How?**	**Which Technique?**
refinement of - timer - channel - task component exchange	separately	SES workbench	coupling of GEODE-SDL with the SES workbench (simulation environment)

- Functional aspects of the SDL description are modelled in the SDL environment
- Aspects related to time and non-ideal features of hardware are modelled by the SES workbench.
- SES workbench provides components to ....
  - ... generate timeouts after a specified time,
  - ... model communication links (delay and error) and
  - ... model processing delay incurred by SDL tasks and processes.
- The coupling is implemented by routing messages typically exchanged between the application-specific code and the SDL runtime support system through the SES workbench.

Literature: Chr. Schaffer, R.J. Raschhofer, A. Simm; EaSy-Sim: A Tool Environment for the Design of Complex, Real-Time Systems, EUROCAST'95

4. Tools: DO-IT / HW/SW-Codesign Project

Specification of Performance Aspects			Performance Modelling and Analysis
What?	**Where?**	**How?**	**Which Technique?**
system stimuli	→ MSC	annotational (comments in SDL and MSC)	- general simulation
service requests	→ MSC		- task and process graph analysis
machine	→ SDL		- real-time analysis (schedulability)
mapping	→ SDL		
perf. requirements	→ MSC		

- early and systematic integration of performance aspects in the systems engineering process
- automization of the design and implementation process employing model-based optimization techniques
- MSC-based performance evaluation techniques
- derivation of mixed HW/SW implementations (HW/SW codesign)

Literature: Mitschele-Thiel, et al.: DO-IT Toolbox, FORTE/PSTV ‘96
Henke, et al.: Derivation of Efficient Implementations from SDL Specifications, SDL Forum ‘97
Faltin, et al.: Annotational Extension of MSCs to Support Performance Engineering, SDL Forum‘97
Mitschele-Thiel, Slomka: Methodology for HW/SW Codesign of RT-Systems, CONSYSE ‘97

4. Tools: Building LQN Performance Models from Traces

Specification of Performance Aspects		Performance Modelling and Analysis
What?	**Where? How?**	**Which Technique?**
processors & scheduling task allocation workload (arrival rates) resource demands (costs) perf. requirements (deadlines)	skeletal LQN model is completed to a performance model via a textual interface	response delays at any level are derived by analytic and simulative techniques

- Step 1: SDL excecution traces are transformed into angio traces
- Step 2: Identify the type of of messages in the trace (synch., asynch., reply, forwarding)
- Step 3: Identify the different services provided by each process
- Step 4: Find the precedence relationship between activities in each service
- Step 5: Map the software architecture model into an LQN submodel
- .... Merge the submodels, complete it to a performance model and solve with the LQN toolset

Literature: Automated Performance Modeling from Scenarios and SDL Designs of Distributed Systems,
H. El-Sayed, D. Cameron, M. Woodside, PDSE ‘98

5. Concluding Remarks

A joint formal description serves as basis for several engineering activities, i.e. to deal with

- functional aspects and
- nonfunctional aspects

of the system under development.

Merits of SDL/MSC-Based Performance Engineering:

- validation of larger systems
- inherent consistency between the functional and the performance model
- automatic derivation of performance models from SDL or/and MSC specification
- small additional overhead for performance evaluation
- early detection of performance problems and potential performance bottlenecks
- major savings of time and money in later development phases and for later system releases
- no corruption of the systems architecture due to ‘performance-hacking’ (future-save development)

What’s next?

- Case studies and application to real world problems
- Better integration with SDL methodology (including implementation design)
- Stabilization and distribution of tools, training of staff
- Improved cooperation with industry, tool builders, and standardization bodies

6. Further Readings

Systems Engineering with SDL:

- R. Brœk, Ø. Haugen. Engineering Real Time Systems, Prentice Hall, 1993. (Good book on the design and implementation of systems based on SDL)
- A. Olsen, O. Faergemand, B. Moeller-Pedersen, R. Reed, J.R.W. Smith. Systems Engineering Using SDL-92. North Holland, 1994. (Good reference book on SDL)

SDL/MSC-based Performance Evaluation and Performance Engineering:

- A. Mitschele-Thiel, B. Müller-Clostermann, R. Reed (Eds.). Proceedings of the Workshop on Performance and Time in SDL and MSC. Report IMMD VII-1/98, University of Erlangen, Germany, Febr. 1998. (Provides an up-to-date collection of papers on performance evaluation tools)
- A. Mitschele-Thiel, B. Müller-Clostermann. Performance Engineering of SDL/MSC Systems. To appear in Computer Networks and ISDN Systems, Elsevier, 1998. (Provides an overview on SDL- and MSC-based performance evaluation tools)
- A. Mitschele-Thiel. Performance Evaluation of SDL Systems. Proceedings of the 1st Workshop of the SDL Forum Society on SDL and MSC, Informatik-Bericht Nr. 104, Humboldt-Universität zu Berlin, June 1998. (Discussion of the issues involved with the integration of performance and time aspects into the SDL Z.100 standard)