W. Dulz. Performance evaluation of sdl/msc-specified systems. Technical report, Universitat Erlangen-Nurnberg, 1996.

ftp.informatik.uni-erlangen.de/local/inf7/papers/Dulz/Framework-for-Performance-Evaluation-of-SDL-and-MSC-specified-Systems.ps.gz

Performance Evaluation of SDL/MSC-specified Systems *

Winfried Dulz
University of Erlangen-Nürnberg
IMMD VII
Martensstr. 3, 91058 Erlangen, Germany
dulz@informatik.uni-erlangen.de

Keywords

Software engineering, specification standards, performance analysis, performance monitoring.

Abstract

SDL and MSC are standardized specification techniques for the development of reactive real-time systems and protocol architectures. Whereas MSC scenarios only allow to specify the dynamic system behavior in form of event/time diagrams, SDL specifications are used to describe static and functional aspects of the system as well.

In this paper a framework is presented which demonstrates the relationship between formally specified SDL systems and appropriate performance analysis respectively monitoring techniques driven by annoted MSC scenarios. It is further discussed how to analyse some important performance measures for SDL-specified systems, i.e. the utilization of processor and channel components, which correspond to the workload characteristics of given MSC scenarios.

1 Introduction and Motivation

Due to the ever increasing complexity of parallel and distributed systems appropriate computer aided system engineering (CASE) techniques must be used to design safe and high performance systems. CASE tools have to support all phases in the software development and life cycle, namely requirement analysis, design, specification, implementation, test and monitoring of the real system.

Standardized formal description techniques (FDTs) like LOTOS (ISO 8807) which is based on an process algebra approach, ESTELLE (ISO 9074) and SDL (ITU Z.100), both describing the system behavior by means of extended finite state automata [Hog89], provide a unifying theoretical basis for the construction of dedicated CASE tools. Modern object-oriented specification languages like SDL'92 also allow the hierarchical decomposition of systems by reusing less complex subsystems with simpler functional and temporal properties.

*This work is supported by the German Research Council (DFG) in scope of the Special Research Area 182 "Multiprocessor and Network Configurations"

There exist, however, only few results concerning the development of methods and tools for studying the performance of systems which are specified and implemented by using an FDT approach ([DHHMC95], [BB93], [HHMC91]). Performance evaluation starting from an abstract system model either can be achieved by mathematical analysis and simulation techniques in early system design phases or by monitoring parts of the real system in the integration and test phase. Because most modern FDT environments support the automatic generation of executable code from a given formal system specification, monitoring tools can reuse these system specifications as a formal monitoring model as well [DDL95].

In the next section it is first motivated why SDL and MSC are good suited for specifying and developping distributed real-time systems. The presentation of a general framework to evaluate the performance of SDL/MSC-specified systems follows next. Then it is shown how the analysis of the mean performance characteristics of processor and channel components can be achieved by using annotated SDL/MSC specifications. Finally the main results are summarized and a short outlook to future work will conclude the paper.

2 SDL and MSC

The Specification and Description Language (SDL, ITU-Standard Z.100) and Message Sequence Charts (MSC, ITU-Standard Z.120) are widely used FDTs that have been proved to be very useful for developping reactive real-time systems and protocols in telecommunication applications. SDL and MSC can be characterized by the following features.

MSCs are derived from informal textual user descriptions during the requirements analysis phase and specify the dynamic behavior of the system. In the ITU recommendation Z.120 [CCI92b] two equivalent MSC representations are standardized. On the one hand MSC/GR defines a graphical syntax with intuitive graphical elements that are best suited for human interactions via graphical editors and interfaces. On the other hand MSC/PR is a high-level programming notation which can be used by MSC tools. The primary use of MSCs are requirements definition, generation of SDL skeletons, behavior validation of SDL specifications and selection of test case scenarios. The main elements of an MSC diagram are vertical instance axes, that represent the causal relationship between actions, sending and receiving of messages, starting and stopping of timers, creation and deletion of dynamic instances and reaching of well-defined conditions that are represented by hexagonal boxes.

In the following the prototypical connection-oriented Inres protocol will be used to demonstrate the main building blocks of SDL/MSC system specifications. After a timer-controlled connection establishment phase is started by the initiator, a secure connection-oriented data exchange between the inititiator and responder entities via an unsecure medium service follows, until the responder will close the connection at some time or other.
The MSC ConnectionSetup in Figure 1 specifies a succesful connection establishment between the initiator and responder instances of the Inres protocol by means of exchanging protocol data units via the communication medium. An additional timer construct ensures the correct timing conditions between sending and receiving of the CR and CC messages inside of the initiator entity.

SDL is an object-oriented specification language which is used to describe the static system architecture in form of system components such as blocks and channels on the one hand. On the

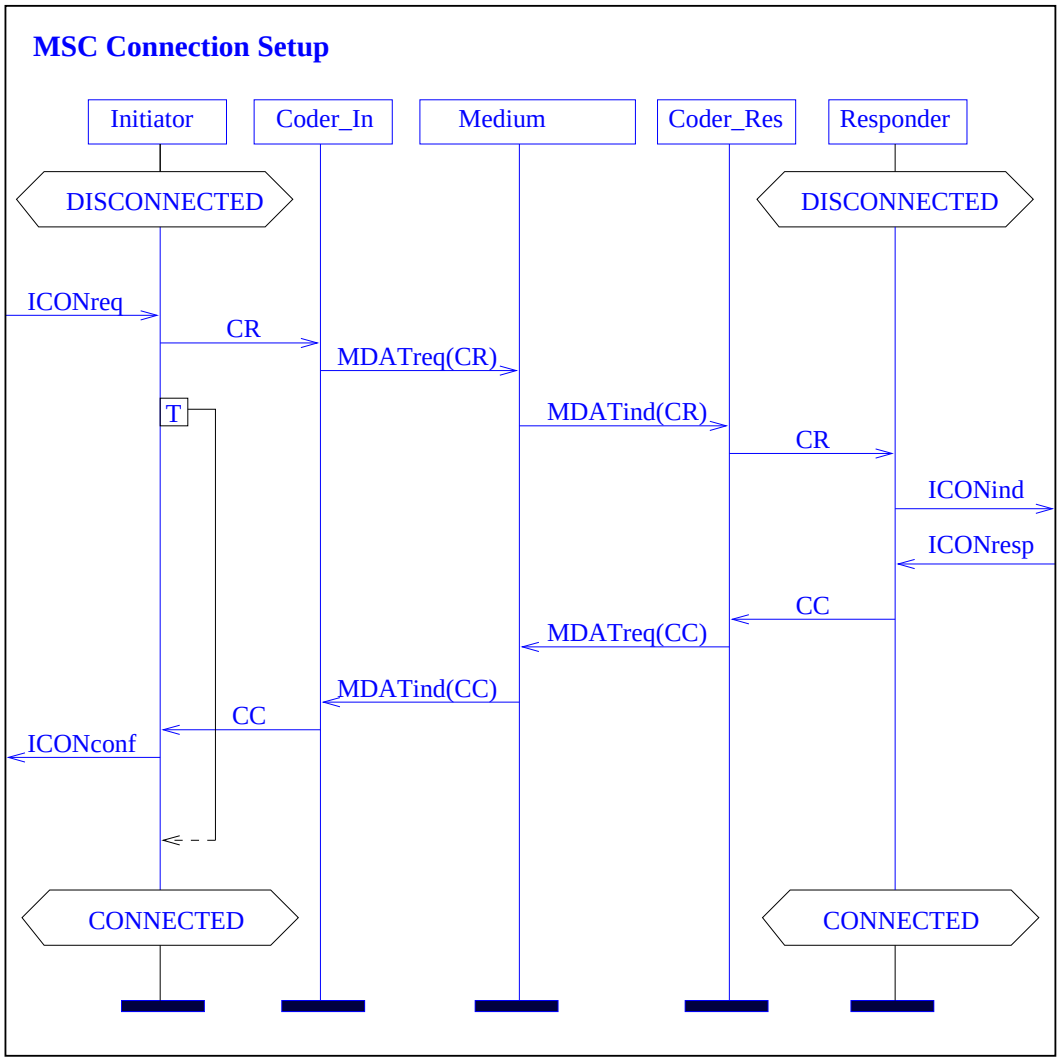

Figure 1: MSC for the successful connection setup

other other hand, dynamic elements like processes and signal routes can be specified, together with the description of their dynamic functional behavior. Just as for the MSCs there exist two equivalent representations for SDL as well, called SDL/GR and SDL/PR [CCI92a]. Whereas the first standardized notation is best suited for developping graphical user interfaces the second variant is better applied to SDL tools which perform the functional simulation and validation or compilation of executable systems ([SDT93]).
In Figure 2 a non-standard white-box decomposition of the Inres system is shown. Blocks represented as rectangles are connected by bidirectional channels and contain octagonal process components that are connected by signal routes. Channels and signal routes are augmented with signal list that contain names of signals which can be exchanged via these links.

Most SDL development systems offer a variety of functionalities and tools, such as graphical edititing, functional simulation or validation and compilation of executable object code for a given run-time environment. The implementation of high performance real-time applications demands, however, that quantitative requirement constraints must also be considered. These are among others

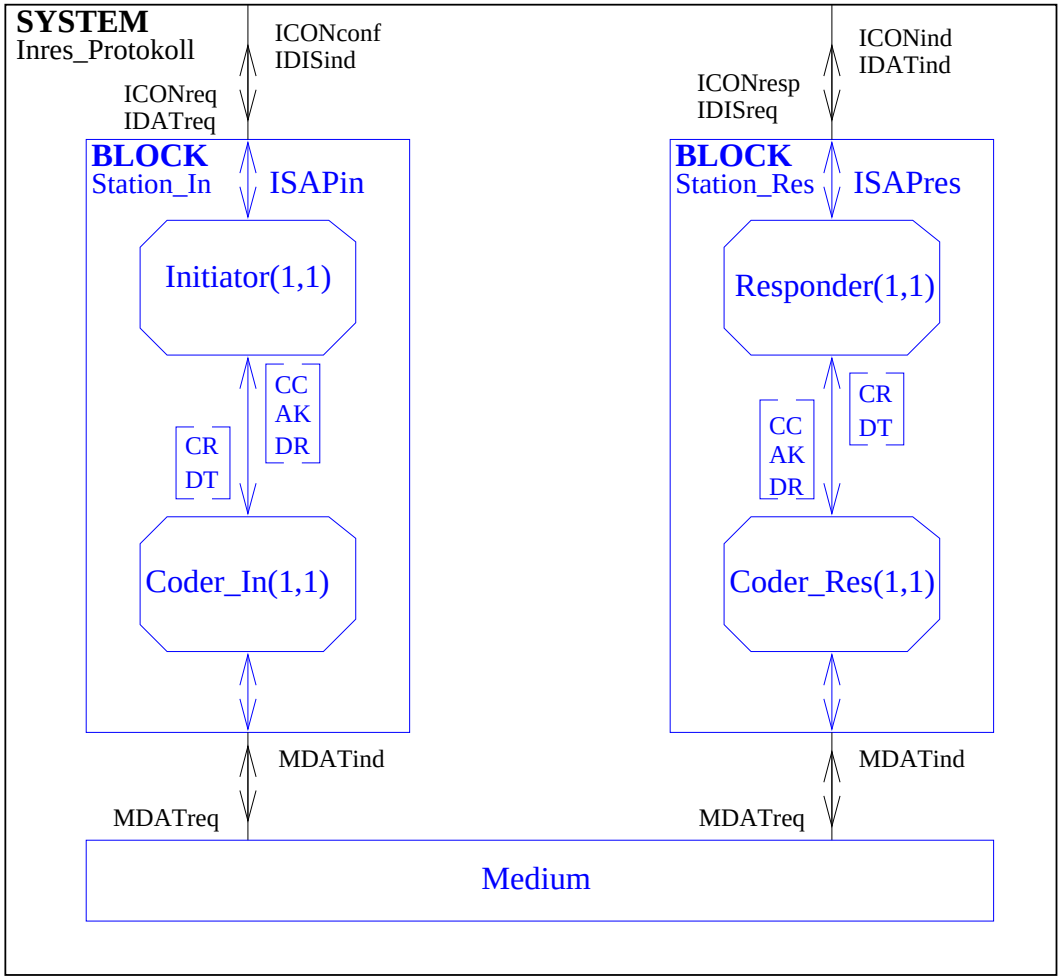

Figure 2: SDL specification of the Inres system

- **time constraints**: guaranteed reaction of the system within a well-defined time limit
- **performance constraints**: guaranteed response time, respectively throughput characteristics for a given workload scenario.

In the next section we will therefore discuss a general framework that can be applied to evaluate the performance of SDL/MSC-specified systems.

3 Performance Evaluation based on SDL/MSC-Specifications

System implementations which are based on an FDT approach will in general possess an improved functional correctness compared to usual developping techniques. Nevertheless, it is necessary to ensure the correct performance behavior in early design phases too. For instance, a base station that has to handle calls for a given number of mobile stations must not be saturated under normal workload conditions. We therefore propose the following framework for the specification-based performance evaluation of SDL/MSC-specified systems (see also Figure 3):

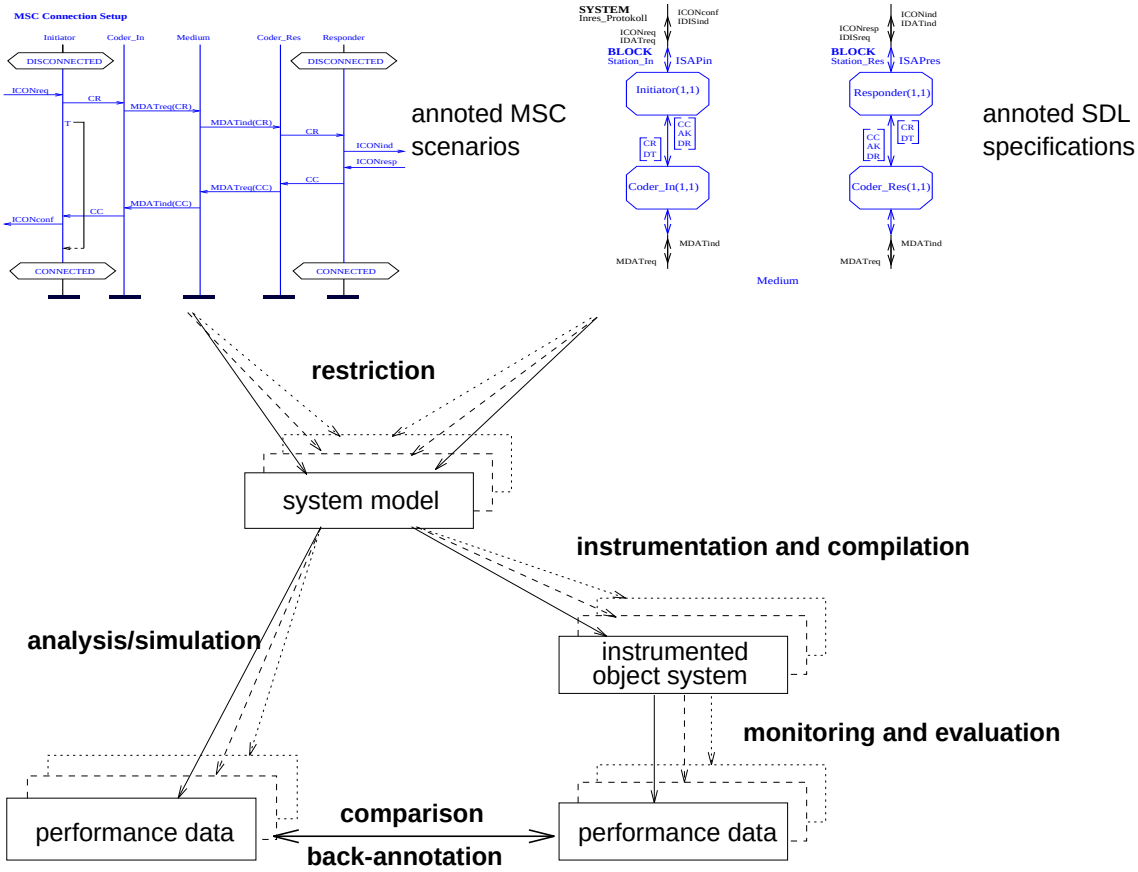

Figure 3: General model for the performance evaluation of SDL/MSC-specified systems

- Well-defined MSC sequences which constitute **MSC scenarios** are used to describe all dynamic aspects of the system under development and are infered during the requirement analysis phase from informal documents and user constraints. Each MSC specifies causal relations with respect to communication events, start and stop of process instances or timer actions and is annotated by its relative frequency, the mean arrival rate and lengths of messages and the mean service time of process components. It is therefore possible to consider MSC scenarios as **workload** definitions which affect the corresponding SDL-specified system.

- SDL specifications describe the static structure of the system and dynamic properties of active system components, such as processes or signal routes as well. We can therefore interpret the SDL specification as a **server model** that consists of active and passive resources which may be visited by workload elements such as actions or messages from the MSC scenarios.

- The restriction of the SDL system dynamics with respect to a given MSC scenario will yield a **system model** which may be evaluated by analytical or simulative techniques, depending on the complexity of the system model and allows the calculation of the characteristic performance parameters, e.g. throughput, utilization or response time.

- By modifying either the MSC specifications to examine heavy load situations or changing the SDL specification to increase or decrease the parallelism of concurrent system com-

ponents it is possible to compare different SDL specifications and to find the best performance solution for a given MSC scenario.

- When using SDL CASE tools it is possible to translate the best SDL specification in big parts automatically into a run-time target system.

- During the execution of the run-time SDL system a **specification-based monitoring** approach will produce event traces with respect to the given MSC scenario. The evaluation and analysis of the measured event trace finally yield insight into the dynamic system beavior and verifies wether the calculated performance indices of analysis, respectively simulation tools are correct [DDL95]. The final back-annotation of measured performance values into the annotated SDL/MSC specifications will eventually lead to better performance predictions during later analysis, respectively simulation steps.

In the next section its is shown how the analysis of mean performance characteristics concerning processor and channel components can be achieved by using annotated SDL/MSC specifications.

4 Analysis of Performance Parameters

To calculate the performance characteristics of a given SDL/MSC system model additional attributes have to be considered and must be annotated to the original specifications. These performance annotations characterize among other things

- arrival rates of MSC scenarios
- relative frequencies of the MSCs within one MSC scenarium
- lengths of messages which are exchanged between MSC instances
- bandwidth of SDL channels,
- service times of SDL processes and finally
- speedup factor of processor elements.

To generate system models additional processor declarations and mapping statements which describe the binding of SDL blocks and processes to processor elements must also be supplied in corresponding annotations [Hop93].

4.1 Calculation of Processor Utilizations

As discussed in the previous framework section a selected subset of all possible MSC scenarios constitutes the systems workload. In the following only messages and actions including timer operations are considered. We will therefore only **focus on static system structures** and neglect dynamic process creations and deletions which are considered in ongoing investigations. It is further assumed that the **relative processor performance** compared to a **norm** processor

type can be expressed by means of a linear **speedup factor**, which is similar to the scalable performance approach described in [BEO95] to facilitate the evaluation process, i.e

$$\mathrm{S}_p \quad = \quad \frac{\text{given processor performance}}{\text{norm processor performance}} \tag{1}$$

where S_p: speedup factor of processor p compared to the norm processor, $(p = 1, \ldots, |\mathrm{P}|)$, P: set of all processor elements in the system.

Because a single SDL process may be an instance within more than one MSC or MSC scenario the service time required for each process and all possible MSCs must be calculated first, i.e.

$$\mathrm{T}_{i,j,k} \quad = \quad \sum_{a \in \mathrm{A}_{i,j,k}} t_a \quad + \sum_{a \in \mathrm{S}_{i,j,k}} t_s \quad + \sum_{a \in \mathrm{R}_{i,j,k}} t_r \tag{2}$$

and $\mathrm{T}_{i,j,k}$: service time requirement of process instance k within MSC j of the i-th MSC scenario, t_a: service time of executing action a, t_s: service time of sending message s, t_r: service time of receiving message r, $\mathrm{A}_{i,j,k}$: set of all actions of process k within MSC j of the i-th MSC scenario, $\mathrm{S}_{i,j,k}$: set of sended messages of process k within MSC j of the i-th MSC scenario and $\mathrm{R}_{i,j,k}$: set of received messages of process k within MSC j of the i-th MSC scenario.

In the second step the calculated service times in (2) are weighted with respect to the arrival rate of the corresponding MSC scenarios and the relative occurence of the MSCs within each MSC scenario. This will yield the **normalized utilization** of the norm processor relative to each process instance of the considered MSC scenarios, i.e

$$\rho_{norm,k} \quad = \quad \sum_{i \in \mathrm{M}} \quad \left(\sum_{j \in \mathrm{M}_i} \mathrm{T}_{i,j,k} * \omega_j \right) \quad * \quad \lambda_i \tag{3}$$

where $\rho_{norm,k}$: normalized utilization relative to process instance k, ω_j: relative frequency of MSC j, λ_i: arrival rate of MSC scenario i, M: set of all possible MSC scenarios for the SDL system under investigation, M_i: set of MSCs of MSC scenario i.

By summarizing all normalized service time requirements for those processes which are mapped to a certain processor in the system model and weighting the result with the speedup factor relative to the norm processor it is possible to calculate the **relative utilization** for a given application. To obtain the **effective utilization** an additional factor for considering background activities that are not related to MSC scenarios must be added. This will finally yield

$$\rho_p \quad = \quad \sum_{k \in \mathrm{X}_p} \quad \rho_{norm,k} \quad * \quad \mathrm{S}_p \quad + \quad \rho_p^b \tag{4}$$

where ρ_p: effective utilization of processor p $(p = 1, \ldots, |\mathrm{P}|)$, ρ_p^b: background utilization of processor p, X_p: set of all SDL processes which are mapped to processor p.

By analyzing the calculated utilization values in (4) possible overload situations for each processor element may be detected. In such a case the entries of the **processor/load matrix** indicates which service is responsible for overloading the processor, i.e.

$$\varphi_{p,i} \quad = \quad \sum_{k \in \mathrm{X}_p} \quad \left(\sum_{j \in \mathrm{M}_i} \mathrm{T}_{i,j,k} * \omega_j \right) \quad * \quad \mathrm{S}_p \tag{5}$$

where $\varphi_{p,i}$ is the **normalized workload** of processor p induced by MSC scenario i (see also Figure 4).

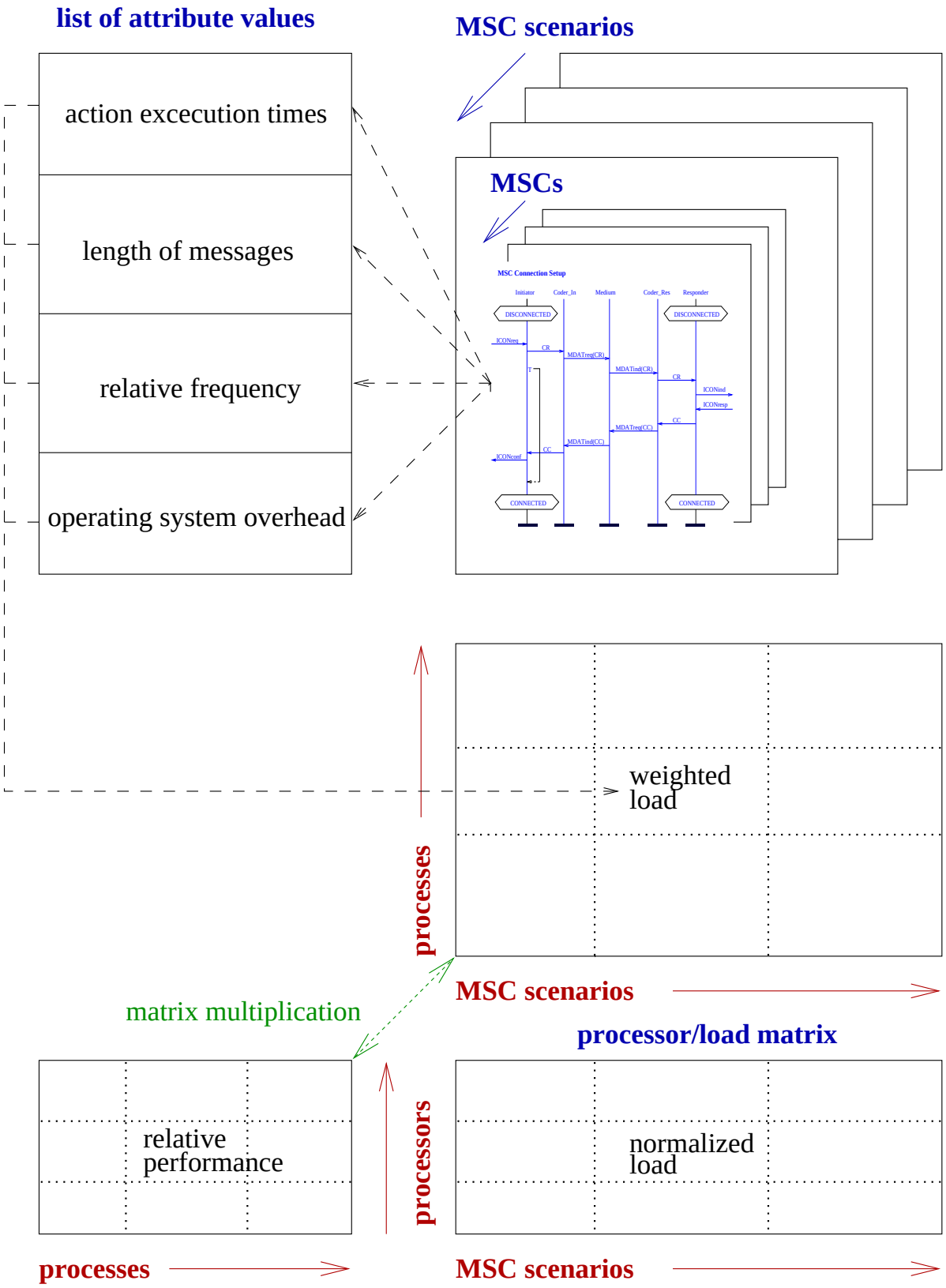

Figure 4: Calculation of the processor/load matrix

4.2 Calculation of Channel Utilizations

The exchange of messages with a given length between MSC process instances via signal routes will physically affect SDL channel components which possess a limited bandwidth. In the first step the **total length** of all messages that will be exchanged via signal routes has to be calculated according to

$$\mathrm{L}_{i,j,r} = \sum_{m \in \mathrm{M}_{i,j,r}} l_m \tag{6}$$

where $\mathrm{L}_{i,j,r}$: total length of messages which are transmitted via signal route r in MSC j of the i-th MSC scenario $(r = 1, \ldots, |\mathrm{R}|)$, l_m: length of message m, $\mathrm{M}_{i,j,r}$: set of all messages, messages which are transmitted via signal route r in MSC j of the i-th MSC scenario, R: set of all signal routes in the SDL system.

In analogy to (3) it is possible to calculate the total bandwith requirements for each SDL signal route by weighting the calculated total length in (6) with respect to the relative occurence and arrival rate of the corresponding MSC and MSC scenarios, i.e.

$$\psi_r = \sum_{i \in \mathrm{M}} \left(\sum_{j \in \mathrm{M}_i} \mathrm{L}_{i,j,r} * \omega_j \right) * \lambda_i \tag{7}$$

where ψ_r is the total bandwidth requirement of signal route r.

Further the **effective bandwith requirement** for each SDL channel is obtained in analogy to (4) by considering all connected signal routes and additional background communications which are not related to MSC scenarios, which yields

$$\psi_s = \sum_{r \in \mathrm{Y}_s} \psi_r + \psi_s^b \tag{8}$$

where ψ_s: effective bandwith requirements for channel s $(s = 1, \ldots, |\mathrm{S}|)$, ψ_p^b: background bandwith requirements for channel s, S: set of all SDL channels, Y_s: set of all SDL signal routes which are connected to channel s.

Finally the **channel utilization** ρ_s is calculated to

$$\rho_s = \frac{\psi_s}{\mathrm{C}_s} \tag{9}$$

and C_s is the channel capacity of channel s. By analyzing the calculated utilization values in (9) possible overload situations for each channel element may also be detected. In such a case the entries of the **channel/load matrix** indicates which service is responsible for overloading the channel, i.e.

$$\varphi_{s,i} = \sum_{r \in \mathrm{Y}_s} \left(\sum_{j \in \mathrm{M}_i} \mathrm{L}_{i,j,r} * \omega_j \right) \tag{10}$$

where $\varphi_{s,i}$ is the required amount of data to be communicated via channel s induced by MSC scenario i.

5 Concluding Remarks

By considering performance evaluation techniques in early stages of a systems lifecycle it is possible to reduce the overall development costs. This is due to the fact that the detection of performance bottlenecks in the real system will often lead to a complete redesign. Methods which allow to predict the performance behavior of formally specified systems are therefore needed.

In this paper a general framework to calculate some basic performance indices for SDL/MSC-specified systems, i.e. the utilization of SDL processor and channel components relative to the given MSC scenarios was presented. In [DDL95] it was further demonstrated how performance characteristics of real SDL/MSC implementations can be measured when an MSC-based monitoring approach is used.

Recent investigations will also consider analysis techniques which include the specification of dynamic SDL/MSC system components, i.e. creating and deleting of SDL processes and signal routes which may compete for a restricted number of system resources, such as processor, memory and channel elements.

References

[BB93] F. Bause and P. Buchholz. Qualitative and Quantitative Analysis of Timed SDL Specifications. In *Kommunikation in Verteilten Systemen*, pages 486–500, München, March 1993. Springer, Informatik aktuell.

[BEO95] T. Benner, R. Ernsts, and A. Österling. Scalable Performance Scheduling for Hardware-Software Cosynthesis. In *Proc. of the European Design Automation Conference, Brighton, Sept. 18-22*, 1995.

[CCI92a] CCITT. *Recommendation Z.100: Specification and Description Language SDL, Blue Book*. ITU General Secreteriat — Sales Section, Place des Nations, CH-1211 Geneva 20, 1992.

[CCI92b] CCITT. *Recommendation Z.120: Message Sequence Charts (MSC)*. ITU General Secreteriat — Sales Section, Place des Nations, CH-1211 Geneva 20, 1992.

[DDL95] P. Dauphin, W. Dulz, and F. Lemmen. Specification-driven Performance Monitoring of SDL/MSC-specified Protocols. In A. Cavalli and S. Budkowski, editors, *Proc. of 8th Int. Workshop on Protocol Test Systems*, pages 409–424, Sept. 1995.

[DHHMC95] Marc Diefenbruch, Elke Heck, Jörg Hintelmann, and Bruno Müller-Clostermann. Performance Evaluation of SDL Systems Adjunct by Queueing Models. In Rolv Braek and Amardeo Sarma, editors, *Proc. Seventh SDL Forum, Oslo, Norway*, pages 231–242, 1995.

[HHMC91] E. Heck, D. Hogrefe, and B. Müller-Clostermann. Hierarchical Performance Evaluation Based on Formally Specified Communication Protocols. *IEEE Trans. on Computers*, 40(4):500–513, April 1991.

[Hog89] Dieter Hogrefe. Estelle, LOTOS und SDL, 1989.

[Hop93] K. Hopmann. Lastanalysator für hierarchische Signalisiernetze, Diplomarbeit, IMMD7, Universität Erlangen-Nürnberg , 1993.

[SDT93] SDT. *SDT 2.3 User's Guide* , 1993.