FMC-QE
A New Approach in Quantitative Modeling

Werner Zorn
Hasso Plattner Institute at the University of Potsdam
Prof.-Dr.-Helmert Str. 2-3, Potsdam, 14482, Germany
E-mail: zorn@hpi.uni-potsdam.de

Abstract—**Service requests are the origin of every service provisioning process and therefore the entities to be considered first. Similar to Physics and Engineering Sciences, service requests as generic model variables are 2-tuples of {value} and [unit]. Complex service requests build hierarchies with control structures for the server stations. With respect to the abstractions of quantitative modeling, degrees of freedom within these control structures can be used to transform models into equivalent ones, which make it possible to simplify the evaluation process. Service request specific server stations can be mapped to common multiplexed server stations, thereby extending the application domain to multiclass problems.**

The modeling and evaluation calculus FMC-QE uses a 3-dimensional representation space, which makes it possible to integrate paradigms and results from Queueing Networks as well as those from Timed Petri Nets. Quantitative models are represented by one or more trees of parameterized flow balance equations, being placed in a Tableau structure for evaluation. Due to hierarchical modeling and the simplicity of the corresponding formulas the computational complexity is minimal. This will be demonstrated by means of a well known problem.

I. INTRODUCTION

Quantitative evaluation techniques of state discrete systems can be roughly divided into those based on Queueing Theory and those on Time augmented Petri Net Theory as illustrated by Fig. 1. Models of the former class are represented by static networks of queueing server stations, whereas those of the latter use dynamic networks of places and transitions.

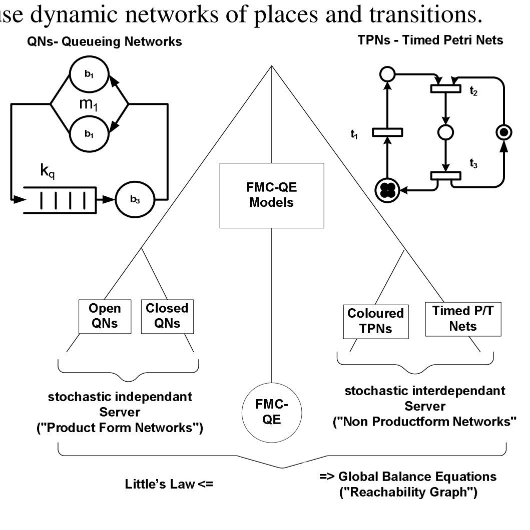

Fig. 1: Classification of Quantitative Methods

The new approach presented here tries to integrate the different views of the classical disciplines within one model, that allows to extend the class of problems to be efficiently handled, e.g. performance analysis of large software based systems.

The models presented in the following are based on FMC- the Fundamental Modeling Concepts, developed and applied in teaching and research at HPI ([12], [16], www.f-m-c.org), in which three different structures are used (see **Fig.**), namely:

1. Static structures, which describe the composition of active and passive components
2. Dynamic structures, which describe the causal ordering of the state transitions
3. Value structures, which describe the contents with their value ranges and relations

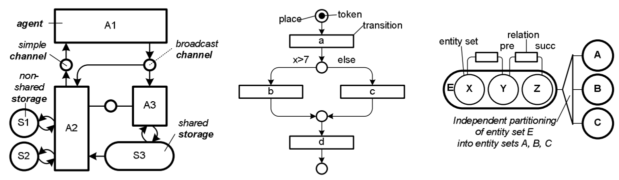

Fig. 2: The Three Plan Types Used in FMC

Each structure corresponds to a bipartite graph (Agent Channel Net, Petri Net, Entity Relation Diagram), referring to each other. We consider these structures to be the dimensions of a 3-dimensional representation space. Successful modeling with FMC of SAP's R/3 system in the 1990s has proven FMC's usability even for large software systems. We therefore base our model on FMC. It is called FMC-QE (QE for quantitative evaluation).

Quantitative Modeling for practical applications has always to be considered under the aspect of cost efficiency. This requires analyzing both the system and the kind of questions to be answered through modeling.

For the class of performance related questions, if one asks for the system's throughput and response times under specific load assumptions, there is a well known tradeoff between computational complexity and gain in accuracy of the measures to be evaluate (s. 2.)

Fig. 3 illustrates the well-known fact that the performance measures, throughput, and response time over number of customers for continuously working systems have a well defined asymptotic behavior, which is widely independent of the type of mathematical model applied.

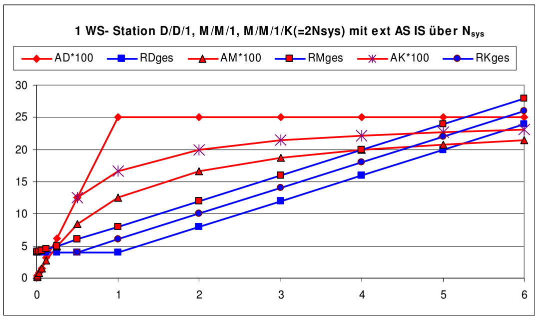

Fig. 3: Asymptotic Performance Behavior A,R over N

The results of deterministic and stochastic analysis (here D/D/1, M/M/1 and M/M/1/K) for a single queueing station differ only for small numbers of customers. Assuming, that this station represents the bottleneck component of a larger system, then the asymptotic behavior of the total system is determined accordingly.

Facing these facts the question is, whether and how the cost effectiveness of quantitative modeling can be increased, possibly using additional degrees of freedom in the chosen 3-dimensional modeling space. This goal should not exclude more detailed analysis down to the state space explorations at the Markov chain level, if the real problem requires that.

The approaches pursued so far, make use of the following four techniques:

1. reduction of the methodical complexity of open and closed systems by a common model
2. reduction of state space complexity by distinction of control and operational states
3. reduction of algorithmic complexity by transforming the model into an equivalent one
4. reduction of computational complexity by approximating product form solutions

II. Modeling Service Requests

As in the physical and engineering sciences, service requests are tuples of {value} and [unit]:

Service Requests

$SRq_i^{[bb]}$ Service Request of type i at hierarchical level [bb]

$N_i^{e[bb]}=\{N_i^{e[bb]}\}\ [N_i^{e[bb]}]$ unified Service Request of type i at level [bb]

$\{N_i^{e[bb]}\} = 1$ (by definition)

$N_i^{[bb]}=\{N_i^{[bb]}\}\ [N_i^{e[bb]}]$ as multiples of unified Service Request $N_i^{e[bb]}$

$n_i^{[bb]} = \{N_i^{[bb]}\}$ number of Service Requests of type i at level [bb]

Service Response

$SRs_i^{[bb]}$ Response to $SRq_i^{[bb]}$ (e.g. ACK, NAK(cause), unknown)

(only used in compositional structures (static plans))

The variables carry as attribute the hierarchical level [bb] with the values 1 for the root and 0 for the surrounding universe. The generative structure together with an example is shown in Fig. 4 and Fig. 5.

When dealing with hierarchies, two kinds of relations between their elements have to be carefully distinguished:

1. organizational hierarchies with super-/subordinate relations
2. abstraction hierarchies with compose/decompose relations

Complex service requests form hierarchies of kind 2), thereby decomposing into less complex ones (e.g., go shopping: buy milk, fill up the car, cut your hair). The decomposition is done by control service requests, the final service by operational service requests.

Hierarchical Traffic Transformation Flow Coefficients v_i allow requests in each control node to be replicated. Degrees of freedom within the control structures allow transformations into equivalent networks.

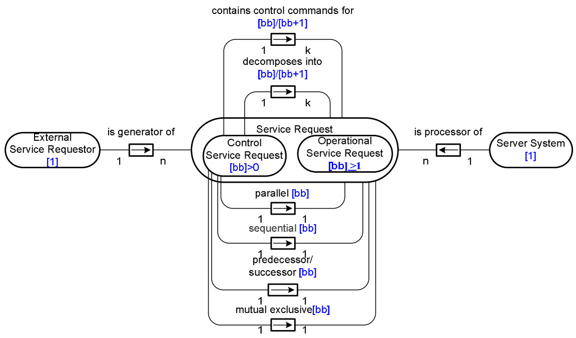

Fig. 4: Generative Structure for Hierarchical Service Request

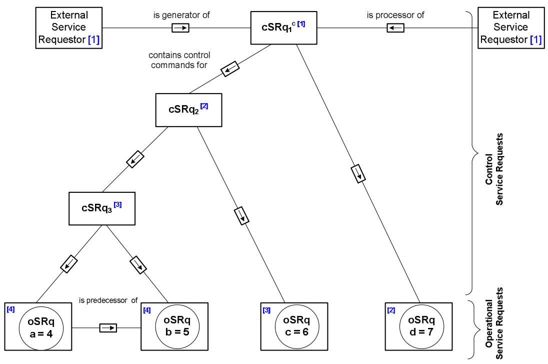

Fig. 5: Tree Structure of Hierarchical Service Requests

III. Modeling Server Stations

The structure of the server system is generated through a 1:1 mapping of the Service Request tree as shown by Fig. 5.

Hierarchical Server Stations, hSSt, process control server requests. Basic Server Stations, BSSt, process the operational server requests. The latter are those, which need service times, whereas the former execute "timeless". The sharing of the same servers as common resources by different BSSts is described later**.** Each Server Station Subtree can be replicated by means of a Server Station Multiplexing Coefficient $m_i^{[bb]}$. The common model of a queueing server station, shown in Fig. 6a) is refined to the models in Fig. 6 b) – d).

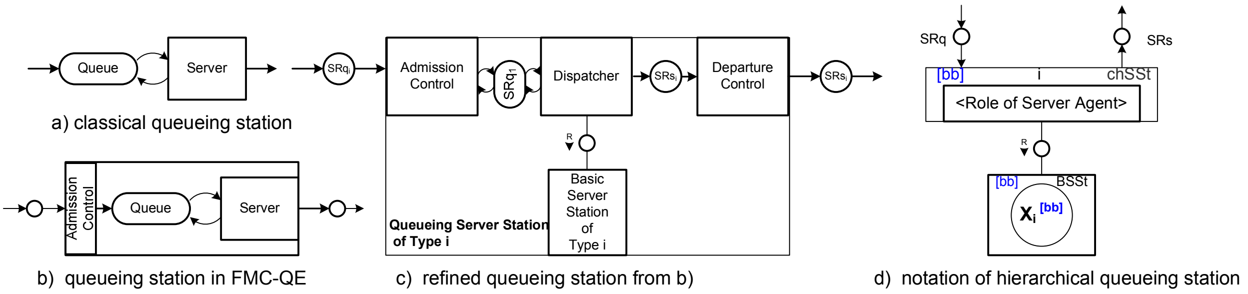

Fig. 6: modeling Server Stations

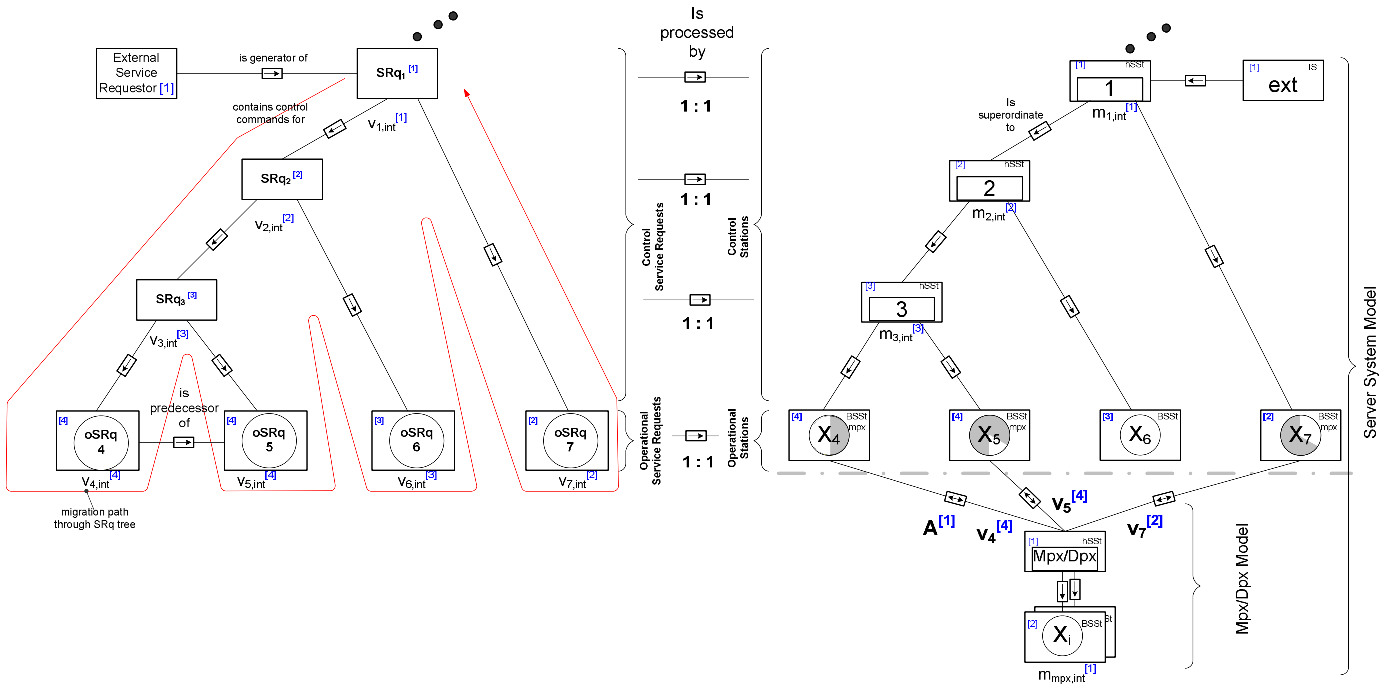

Fig. 7: Relation between Server Request's and Server System's Structure

IV. Modeling Transitions

It is a well known practice in design and implementation of IT systems to distinguish between control states and operational states of a system [19]. While the control state space of a single server is typically very small, at least comprising the 2 states "Idle" and "Busy", the operational state space is potentially infinite, either countable infinite for discrete operational items or uncountable infinite for continuous operational variables.

This concept can easily be applied in systems modeling through distinguishing between "black" control tokens for marking the control states and "colored" operational tokens representing the customers, carrying the type and amount of service required as attached values. The corresponding structural dynamic elements are shown in Fig. . Considering a whole network, there are the following definitions:

Control state: marking of places with control tokens

$$M^c = (M^c_1, \ldots M^c_i, \ldots M^c_k)$$

Operational state: number of customers

$$N = (N_1, \ldots N_i, \ldots N_k), \;\; N_i = (N_{i,s}, N_{i,q})$$

with Ni,s and Ni,q as numbers in server resp. in queue.

In a system with k queueing stations, the states are represented by values of variables to be found in the compositional structure for the server agent.

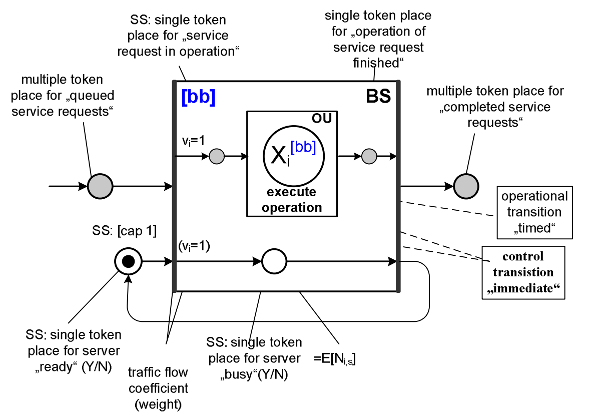

Fig. 8: Structure of Controlled Operational Transition T^{co}

The T^{co} embedded into the control structure of a BSSt with its queueing place and dispatching transition is show **7**.

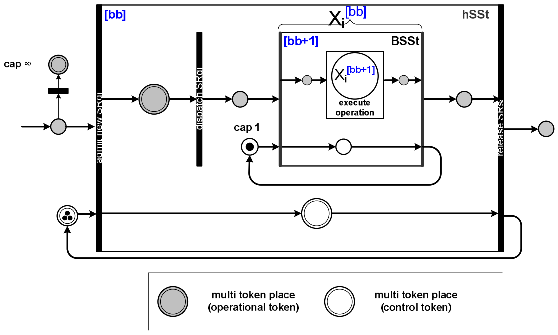

Fig. 9: Basic Server Station BSSt with Embedded T^{co}

The very common serial connection of server stations raises the questions of the appropriate interpretation of serial transitions, when the service request flow contains service responses after passing a T^{co}. The easiest understanding is a simplified model for a control flow, where each service request is directed to the subsequent station through the hierarchical station, when the service response is fired by the predecessor transition.

V. Fundamental Laws

The fundamental Laws applied are the so called:

1. Little's Law (LL)
2. Forced Traffic Flow Law (FTFL)

The former is the well known "black box" law from Queueing Theory to interrelate N, A and R for system in steady state. The FTFL however is a "shadow" law, mostly used, when visits of server stations are considered.

In this hierarchical context, the two laws can be looked at as complementary laws, where LL describes dynamic processes on one hierarchical layer, and FTFL describes processes between two adjacent hierarchical layers. Both together are able to describe processes within hierarchical systems. It is important to emphasize the fundamental difference between the very similar appearing flow balance equation

$\lambda_i = p_{j,i}\, \lambda_j$ (1)

and the FTFL equation

$\lambda_i = v_i\, \lambda$ (2)

Little's Law (LL)

$N_i^{[bb]} = A_i^{[bb]}\, R_i^{[bb]}$

Forced Traffic Flow Law (FTFL)

$A_i^{[bb]} = v_{i,int}^{[bb]}\, A_h^{[bb-1]}$ $h = \sup(i)$

$A_i^{[bb+1]} = N_i^{[bb+1]}/\Delta t$

$A_j^{[bb]} = N_j^{[bb]}/\Delta t$

Traffic Flow Transformation Coefficient

$v_{i,int}^{[bb]} = \{v_{i,int}^{[bb]}\}[v_{i,int}^{[bb]}]$

$\{v_{i,int}^{[bb]}\} = \{A_i^{[bb]}\}/\{A_h^{[bb-1]}\}$

$[v_{i,int}^{[bb]}] = [N_{i,int}^{e[bb]}]/[N_h^{e[bb-1]}]$

These Traffic Flow Transformation coefficients are sometimes used to calculate number of visits from probabilities. This can be visualized by means of diagrams as in Fig.:

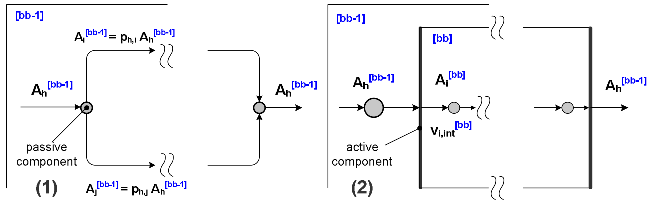

Fig. 10: Traffic Coefficient v_i vs. Probability $p_{i,j}$

Two facts are evident from Fig. 10:

1. Transformation (2) generates a different flow, which can only be done by some active component, represented in Fig. by a transition, whereas transformation (1) splits a flow into partial flows, where the joint flow always equals the unsplit one.
2. The overall steady state condition always requires the reverse transformation of (2), which corresponds to nested operations between two adjacent hierarchical layers.

Beside the two fundamental laws there are other relations, which form altogether the FMC-QE calculus (see Annex).

VI. Modeling Stationary Processes

Open Queueing Networks (OQN) and Closed Queueing Networks (CQN), the latter being the class to which time augmented Petri Nets (Stochastic Petri Nets SPNs, Generalized Stochastic Petri Nets GSPNs) belong, are considered and treated in a totally different way [15]. In OQN the arrival rate A is the independent parameter, where as in CQN the number N of customers is the independent parameter. The common abstraction however is the stationarity of processes, where each departing customer is replaced by an arriving one, either in a deterministic synchronized or a stochastic way. Both can be combined in a closed system, if the outside birth/death process is modeled by an external server with the two parameters A and N.

Fig. 11 illustrates the property, that customers cycle with a mean interarrival time through the serving system as well as through the outside world. The more customers want service, the longer is the roundtrip time. The more the arrival rate approaches the maximum throughput of the serving system, the more customers queue up there.

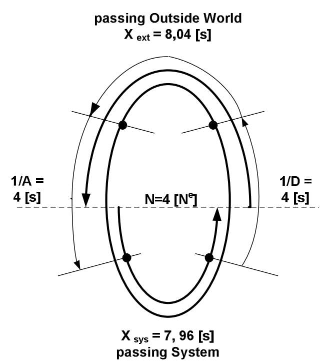

Fig. 11: Stationary Process with Parameters A and N

Fig. 11 shows the time X_{ext} spent outside as function of the free parameters, arrival rate A and total number of customers N. The external birth-death process is modeled as an infinite server, using the response time law for X_{ext}. It thus resembles the model for a dialogue system.

VII. The FMC-QE Tableau

The quantitative model of a server system is represented by a parameterized set of equations, which is tree structured. This tree of equations can be understood as the production tree of some program, which can be evaluated bottom up by a stack based interpreter. This tree can be stored in a table structure, called the FMC-QE Tableau. The interpreter used so far for the evaluation of the Tableau here was MS Excel. A more powerful interpreter is under development (see Conclusion and Outlook).

According to the 3-dimensional representation of models, the Tableau consists of:

1. Service Request Section
2. Server Section
3. Dynamic Evaluation Section

together with

4. Common Section
5. Global Parameter Section.

A description of the Tableau's syntax and semantic is given in the "FMC-QE Calculus" (see Annex), and will be illustrated by a simple example in the following section.

VIII. Example

Given the following open queueing network [3]:

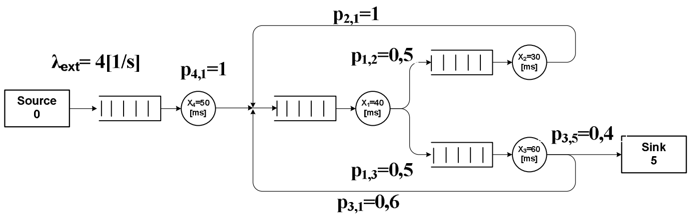

Fig. 12: Open Queueing Network from [3], p. 76 ff

Given:

λ_{ext} = 4 [1/s]

$\underline{\mu}$ = (25; 33,33; 16,67; 20) [1/s]

(<see QN drawing>)

Solution of the flow balance equation system

$\underline{\lambda} = \underline{\lambda}\, \mathbf{P}$

results in:

$\underline{\lambda}$ = (20;10;10;4) [1/s]

$\underline{\rho}$ = (0,8; 0,3; 0,6; 0,2)

$\underline{n}$ = (4; 0,429; 1,5; 0,25)

$\underline{R}$ = (0,2; 0,043; 0,15; 0,0625) [s]

N_{sys} = 6,179; R_{sys} = 1,545 [s]

The following Figures 13 - 15 represent the FMC-QE model, which is the basis for the Evaluation shown in Tables 1 and 2.

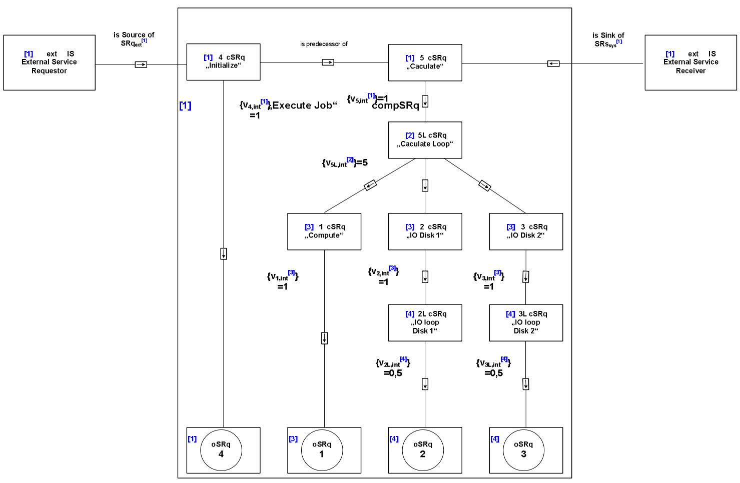

Fig. 13: Service Request structure for example of Fig. 13

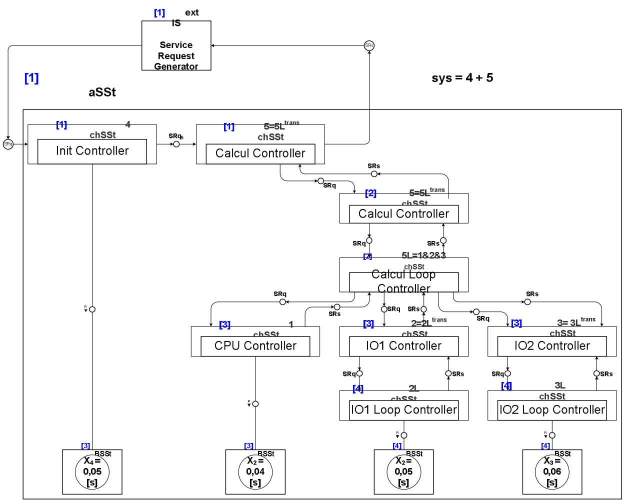

Fig. 14: Server System structure for example of Fig. 1

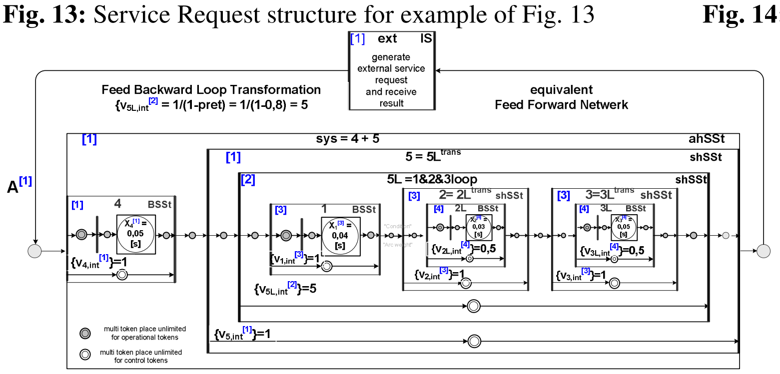

Fig. 15: dynamic control structure for example of Fig. 13

Global Parameter Section	
N_{total} [0]	7
$Amax^{[1]} = min(Bi_{,max})$	5,0
$A^{[1]}$ = f*Amax =	4,0
1≥f =	0,8
p_{ret}	0,8
$v_{sys,intern} = 1/(1-p_{ret})$	5

Table 1a: FMC- QE Tableau (Part I)

Common Se	Service Request Section						Server Section							H: $B_i^{[bb]}$=m	min(B_i/v_i) =	Dynamic Evaluation Section							
	level	SRQ unit					Server category									$v_i^{[bb]}$*A	A_i/B_i	m_i*U_i	$U_i^2/(1-U_i)$	$n_{i,q}+n_{i,s}$	$N_{i,q}/A_i$	$N_{i,s}/A_i$	N_i/A_i
i	[bb]	$[N_i^e]$	p_i	$v_{i,ext}$	$v_{i,int}$	v_i	name	kind	X_i	$m_{i,ext}$	$m_{i,int}$	m_i	: B_i =mi/(X	B_{max} =	$A_i^{[bb]}$=	U_i=	$n_{i,s}$=	$n_{i,q}$ =	n_i=	W_i=	X_i=	R_i=
														Bottleneck				M/M/1				
2L	4	[IO1 loop]	0,5	5	0,5	2,5	Disk 1l	B	0,03	1	1	1	33,3	13,3	10,0	0,30	0,30	0,13	0,43	0,01	0,030	0,043
2=2Ltran	3	[IO 1]	0,5	5	1	5	Disk 1	sH		1	1	1	66,7	13,3	20,0	0,30	0,30	0,13	0,43	0,01	0,015	0,021
3L	4	[IO2 loop]	0,5	5	0,5	2,5	Disk 2l	B	0,06	1	1	1	16,7	6,7	10,0	0,60	0,60	0,90	1,50	0,09	0,060	0,150
3=§Ltran	3	[IO 2]	0,5	5	1	5	Disk 2	sH		1	1	1	33,3	6,7	20,0	0,60	0,60	0,90	1,50	0,05	0,030	0,075
1	3	[Compute]	1	5	1	5	CPU	B	0,04	1	1	1	25,0	5,0	20,0	0,80	0,80	3,20	4,00	0,16	0,040	0,200
5L=1&2	2	alculate loo	1	1	5	5	Calculator	sH		1	1	1	25,0	5,0	20,0	0,80	1,70	4,23	5,93	0,21	0,085	0,296
5=5Ltran	1	[Calculate]	1	1	1	1	Calculator	sH		1	1	1	5,0	5,0	4,0	0,80	1,70	4,23	5,93	1,06	0,425	1,482
4	1	[init]	1	1	1	1	Initiator	B	0,05	1	1	1	20,0	20,0	4,0	0,20	0,20	0,05	0,25	0,01	0,050	0,063
sys=4&5	1	[exec Job]	1	1	1	1	System	aH		1	1	1	5,0	5,0	4,0	0,80	1,90	4,28	6,18	1,07	0,475	1,545
cust	1	[do Job]	1	1	1	1	Customer	IS *)	0,21	1	###	0,82	4,0	4,0	4,0	1,00	0,82	0,00	0,82	0,00	0,205	0,205
uni	0	[host]	1	1	1	1	Universe	aH		1	1	1	4,0	4,0	4,0	1,00	2,72	4,28	7,00	1,07	0,680	1,750

**) IS Server is never Bottleneck — N=A*R = 7,0 Check

Table 1b: FMC- QE Tableau (Part II)

Common Section	Service Request Section						Server Section								Dynamic Evaluation Section							
	level	SRQ unit		$v_i^{[bb]}=v_{i,out}^{[bb]}*v_{i,int}^{[bb]}$			Server category			$m_i^{[bb]}=m_{i,out}^{[bb]}*m_{i,int}^{[bb]}$				$B_i^{[bb]}/v_i^{[bb]}$	$v_i^{[bb]}$*A	$A_i^{[bb]}/B_i^{[bb]}$	$m_i^{[bb]}*U_i^{[bb]}$	$(n_{i,s}^{[bb]})/(1-n_{i,s}^{[bb]})$	$\rho_i/(1-\rho_i)$	$n_{i,q}^{[bb]}/A_i^{[bb]}$	$n_{i,s}^{[bb]}/A_i^{[bb]}$	$n_i^{[bb]}/A_i^{[bb]}$
i	[bb]	$[N_i^e]$	p_i	$v_{i,out}$	$v_{i,int}$	v_i	name	kind	X_i	$m_{i,out}^{[bb]}$	$m_{i,int}^{[bb]}$	$m_i^{[bb]}$=	$B_i^{[bb]}$	$A_{max,i}^{[1]}$ =	$A_i^{[bb]}$=	$U_i^{[bb]}$=	$n_{i,s}^{[bb]}$=	$n_{i,q}^{[bb]}$ =	$n_i^{[bb]}$=	$W_i^{[bb]}$=	$X_i^{[bb]}$=	$R_i^{[bb]}$=
														Bottleneck				M/M/1				
2L	4	[loop IO 1]	0,5	$v_2^{[3]}$	0,5	$v_{2L}^{[4]}$=	Disk 1l	B	0,03	$m_2^{[3]}$	1	$m_{2L}^{[4]}$=	$m_{2L}^{[4]}/X_{2L}^{[4]}$	$B_{2L}^{[4]}/v_{2L}^{[4]}$	$v_{2L}^{[4]}$* A	$A_{2L}^{[4]}/B_{2L}^{[4]}$	$m_{2L}^{[4]}*U_{2L}^{[4]}$	$(n_{2L,s}^{[4]})^2/(1-n_{2L,s}^{[4]})$	$n_{2L,s}^{[4]}+n_{2L,q}^{[4]}$	$n_{2L.q}^{[4]}/A_{2L}^{[4]}$	$n_{2L,s}^{[4]}/A_{2L}^{[4]}$	$n_{2L}^{[4]}/A_{2L}^{[4]}$
2=2L^trans	3	[IO 1']	1	$v_{5L}^{[2]}$	1	$v_2^{[3]}$=	Disk 1	sH		$m_{5L}^{[2]}$	1	$m_2^{[3]}$=	$m_2^{[3]}*B_{2L}^{[4]}/v_{2L,int}^{[4]}$	$B_2^{[3]}/v_2^{[3]}$	$v_2^{[3]}$* A	$A_2^{[3]}/B_2^{[3]}$	$m_2^{[3]}*U_2^{[3]}$	$(n_{2,s}^{[3]})^2/(1-n_{2,s}^{[3]})$	$n_{2,s}^{[3]}+n_{2,q}^{[3]}$	$n_{2,q}^{[3]}/A_2^{[3]}$	$n_{2,s}^{[3]}/A_2^{[3]}$	$n_2^{[3]}/A_2^{[3]}$
3L	4	[loop IO 2]	0,5	$v_3^{[3]}$	0,5	$v_{3L}^{[4]}$=	Disk 2l	B	0,06	$m_3^{[3]}$	1	$m_{3L}^{[4]}$=	$m_{3L}^{[4]}/X_{3L}^{[4]}$	$B_{3L}^{[4]}/v_{3L}^{[4]}$	$v_{3L}^{[4]}$* A	$A_{3L}^{[4]}/B_{3L}^{[4]}$	$m_{3L}^{[4]}*U_{3L}^{[4]}$	$(n_{3L,s}^{[4]})^2/(1-n_{3L,s}^{[4]})$	$n_{3L,s}^{[4]}+n_{3L,q}^{[4]}$	$n_{3L.q}^{[4]}/A_{3L}^{[4]}$	$n_{3L,s}^{[4]}/A_{3L}^{[4]}$	$n_{3L}^{[4]}/A_{3L}^{[4]}$
3=3L^trans	3	[IO 2']	1	$v_{5L}^{[2]}$	1	$v_3^{[3]}$=	Disk 2	sH		$m_{5L}^{[2]}$	1	$m_3^{[3]}$=	$m_3^{[3]}*B_{3L}^{[4]}/v_{3L,int}^{[4]}$	$B_3^{[3]}/v_3^{[3]}$	$v_3^{[3]}$* A	$A_3^{[3]}/B_3^{[3]}$	$m_3^{[3]}*U_3^{[3]}$	$(n_{3,s}^{[3]})^2/(1-n_{3,s}^{[3]})$	$n_{3,s}^{[3]}+n_{3,q}^{[3]}$	$n_{3,q}^{[3]}/A_3^{[3]}$	$n_{3,s}^{[3]}/A_3^{[3]}$	$n_3^{[3]}/A_3^{[3]}$
1	3	[Compute]	1	$v_{5L}^{[2]}$	1	$v_1^{[3]}$=	CPU	B	0,04	$m_{5L}^{[2]}$	1	$m_1^{[3]}$=	$m_1^{[3]}/X_1^{[3]}$	$B_1^{[3]}/v_1^{[3]}$	$v_1^{[3]}$* A	$A_1^{[3]}/B_1^{[3]}$	$m_1^{[3]}*U_1^{[3]}$	$(n_{1,s}^{[3]})/(1-n_{1,s}^{[3]})$	$n_{1,s}^{[3]}+n_{1,q}^{[3]}$	$n_{1,q}^{[3]}/A_1^{[3]}$	$n_{1,s}^{[3]}/A_1^{[3]}$	$n_1^{[3]}/A_1^{[3]}$
5L=1&2&3	2	[loop Calcu	1	$v_5^{[1]}$	5	$v_{5L}^{[2]}$=	Calculator	sH		$m_5^{[1]}$	1	$m_{5L}^{[2]}$=	$m_{5L}^{[2]}*\min(B_j^{[3]}/v_{j,int}^{[3]})$	$B_{5L}^{[2]}/v_{5L}^{[2]}$	$v_{5L}^{[2]}$* A	$A_{5L}^{[2]}/B_{5L}^{[2]}$	$n_{5L,s}^{[2]}=\Sigma n_{i,s}^{[3]}$	$n_{5L,q}^{[2]}=\Sigma n_{i,q}^{[3]}$	$n_{5L,s}^{[2]}+n_{5L,q}^{[2]}$	$n_{5L,q}^{[2]}/A_{5L}^{[2]}$	$n_{5L,s}^{[2]}/A_{5L}^{[2]}$	$n_{5L}^{[2]}/A_{5L}^{[2]}$
5=5L^trans	1	[Calculate]	1	$v_{sys}^{[1]}$	1	$v_5^{[1]}$=	Calculator	sH		$m_{sys}^{[1]}$	1	$m_5^{[1]}$=	$m_5^{[1]}*B_{5L}^{[2]}/v_{5L,int}^{[2]}$	$B_5^{[1]}/v_5^{[1]}$	$v_2^{[1]}$* A	$A_5^{[1]}/B_5^{[1]}$	$m_5^{[1]}*U_5^{[1]}$	$(n_{5,s}^{[1]})^2/(1-n_{5,s}^{[1]})$	$n_{5,s}^{[1]}+n_{5,q}^{[1]}$	$n_{5,q}^{[1]}/A_2^{[1]}$	$n_{5,s}^{[1]}/A_5^{[1]}$	$n_2^{[2]}/A_2^{[2]}$
4	1	[Initialize]	1	$v_{sys}^{[1]}$	1	$v_4^{[1]}$=	Initiator	B	0,05	$m_{sys}^{[1]}$	1	$m_4^{[1]}$=	$m_4^{[1]}/X_4^{[1]}$	$B_4^{[1]}/v_4^{[1]}$	$v_4^{[1]}$* A	$A_4^{[1]}/B_4^{[1]}$	$m_4^{[1]}*U_4^{[1]}$	$(n_{4,s}^{[1]})^2/(1-n_{4,s}^{[1]})$	$n_{4,s}^{[1]}+n_{4,q}^{[1]}$	$n_{4,q}^{[1]}/A_4^{[1]}$	$n_{4,s}^{[1]}/A_4^{[1]}$	$n_4^{[1]}/A_4^{[1]}$
sys=4&5	1	[exec Job]	1	1	1	$v_{sys}^{[1]}$=	System	aH		1	1	$m_{sys}^{[1]}$=	$m_{sys}^{[1]}*\min(B_j^{[1]}/v_{j,int}^{[1]})$	$B_{sys}^{[1]}/v_{sys}^{[1]}$	$v_{sys}^{[1]}$* A	$A_{sys}^{[1]}/B_{sys}^{[1]}$	$m_{sys}^{[1]}*U_{sys}^{[1]}$	$(n_{sys,s}^{[1]})^2/(1-n_{sys,s}^{[1]})$	$n_{sys,s}^{[1]}+n_{ext,q}^{[1]}$	$n_{sys,q}^{[1]}/A_{sys}^{[1]}$	$n_{sys,s}^{[1]}/A_{sys}^{[1]}$	$n_{sys}^{[1]}/A_{sys}^{[1]}$
cust	1	[do Job]	1	1	1	$v_{ext}^{[1]}$	Customer	IS *)	X_{ext}	1	$n^{[0]}-n_{sys}^{[1]}$	$n_{ext}^{[1]}$	$m_{ext}^{[1]}/X_{ext}^{[1]}$	--- **)	$v_{ext}^{[1]}$* A	$A_{ext}^{[1]}/B_{ext}^{[1]}$	$m_{ext}^{[1]}*U_{ext}^{[1]}$	0	$n_{ext,s}^{[1]}$+next,q$^{[1]}$	$n_{ext,q}^{[1]}/A_{ext}^{[1]}$	$n_{ext,s}^{[1]}/A_{ext}^{[1]}$	$n_{ext}^{[1]}/A_{ext}^{[1]}$
uni	0	[let do]	1	1	1	1	Universe	aH		1		1	$B_{ext}^{[1]}=B_{sys}^{[1]}$	$B_{uni}^{[0]}/v_{sys}^{[0]}$	$v_{uni}^{[1]}$* A	$A_{uni}^{[0]}/B_{uni}^{[0]}$	$n_{sys,s}^{[1]}+n_{ext,s}^{[1]}$	$n_{sys,q}^{[1]}+n_{ext,q}^{[1]}$	$n_{uni,s}^{[1]}+n_{uni,q}^{[1]}$	$n_{uni,q}^{[0]}/A^{[1]}$	$n_{uni,s}^{[0]}/A^{[1]}$	$n_{uni}^{[0]}/A^{[1]}$

$X_{ext}^{[1]}=N^{[0]}/A^{[1]}-R_{sys}^{[1]}$

Table 2: formulas from FMC.QE calculus used in Evaluation of Table 1

Fig. 13 shows the service request tree for the QN of Fig. 12 with service times from the corresponding server system tree (Fig. 14). shows the corresponding serialized dynamic control structure. The parameters of these three different models are mapped to the Tableau (see Table 1b) with one hierarchical equation (row) for each entity. The set of rules applied are named the FMC-QE Calculus (see Annex), which delivers the variables of interest within the Dynamic Evaluation Section.

The solution delivered by the Tableau of Table 1b calculus is equivalent to solving the flow balance equation system. While the equation system has to be solved for each set of parameter values, the Tableau has to be set up only once, provided tree structured service requests are modeled. Modeling Multiplexer Server Station

IX. Modeling Multiplexer Server Station

Modeling Multiplex/Demultiplex (Mpx/Dpx) server stations is of particular importance in every modeling approach. Fehler! Verweisquelle konnte nicht gefunden werden. shows the typical representation of the multiplexer service for three requesting systems over time.

There are three parameters to consider for each requesting basic server station:

1. Arrival Rates $A^{z}_{i}{}^{[1]}$, possibly from different sources z
2. Service Request specific Service Times $X^{z}_{i}{}^{[bb]}$
3. Service Request specific Traffic Flow Transformation coefficients $v^{z}_{i}{}^{[bb]}$

The basic idea is to partition the common resource multiplexer into parallel servers and to allocate each of these parallel servers to each of the requesting server stations. The parameter used for this partial allocation is the multiplexing coefficient $m_{i,\ int}$ which was already introduced for modeling real Parallel Servers (PS).

The corresponding dynamic structures are shown in Fig. 16:

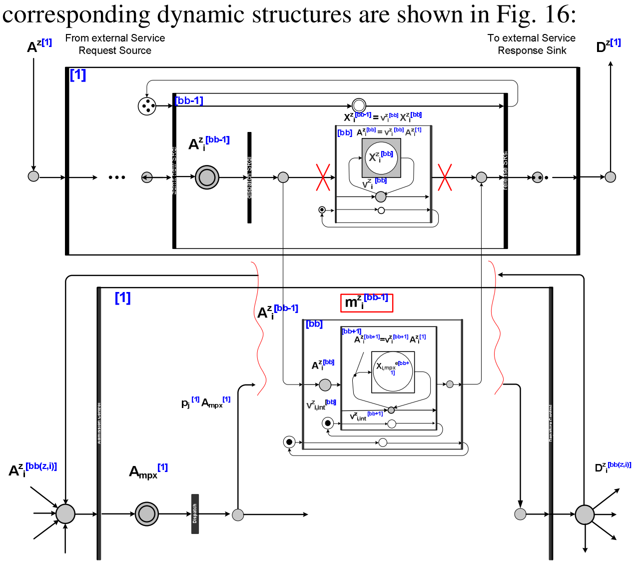

Fig. 16: Requesting Basic Server Station's View

Evaluation of $m^z_{i,int}{}^{[bb(i,z)]}$
$m^z_{i,int}{}^{[bb(i,z)+1]} = m^z_{mpx',i}{}^{[2]}$
$= v^z_{mpx',i}{}^{[3]}/v_{mpx}{}^{[4]}$
$= \Pi\, v^z_{mpx',i}{}^{[b]}/v_{mpx}{}^{[4]}$ for b = 1,2,3
$= \Pi\Sigma v^z_{mpx',i}{}^{[3]}/\Sigma v^s_{mpx',j}{}^{[3]}$ for all rSSt$^s_j{}^{[bb(j,s)]}$
with
$A_{mpx}{}^{[1]} = \Sigma A_j^{s[bb(j,s)-1]}$ for all rSSt$^s_j{}^{[bb(j,s)]}$
$v^s_{mpx'j}{}^{[3]} = (A^{s[1]}/A_{mpx}{}^{[1]}) * \{v^s_j{}^{[bb(j,s)]}\} * (X^s_j{}^{[bb(j,s)]}/X_{mpx}{}^{e[3]})$ for all (s,j)
$v^z_{mpx'i}{}^{[3]} = (A^{z[1]}/A_{mpx}{}^{[1]}) * \{v^z_i{}^{[bb(i,z)]}\} * (X^z_i{}^{[bb(i,z)]}/X_{mpx}{}^{e[3]})$ for all (s)
and
$v_{mpx}{}^{[4]} = v_{mpx}{}^{[3]} = \Sigma v^s_{mpx',j}{}^{[3]}$ for all rSSt$^s_j{}^{[bb(j,s)]}$

$m_{mpx,i}$ is simply exported back from the multiplexer's dispatcher back into the Tableaux of the requesting stations.

X. Conclusion and Outlook

The introduction of service requests as variables with values and units allows distinguishing among the different traffic flows and thereby modeling complex server systems hierarchically. Horizontally, Little's Law (LL) is applied to evaluate the equilibrium state for every single station within the hierarchy, whereas, vertically, the Forced Traffic Flow Law (FTFL) is applied to describe the transformation of service request flows between adjacent hierarchical levels along the paths from the root to the leaves. A set of service invariant transformation rules is given to transform an initial service request processing model into a tree structured one, which complies with the specification and evaluation scheme of the FMC-QE calculus.

For each class of external service requests the tree of parameterized flow balance equations is represented within a Tableau, where the three main sections correspond to the three dimensional description space of FMC [12],[16]. Service request and server station sections – together with the global section for the external source/sink – contain the structural and quantitative model parameters, whereas the dynamic evaluation section contains the well known formulas to evaluate the steady state variables together with performance measures of interest. Due to the interdependences within the hierarchical flow balance sheet, these redundancies can be used to check the model's correctness.

When stationary flows are modeled by means of mandatory external source/sinks, server systems are not classified into OPEN and CLOSED, but into those with unlimited and limited buffer resources. Solutions of the latter class are approximated by applying the well known type M/M/m/K formulas in the Tableau independently for each server station together with an estimate for the approximation error. Since basic server stations in FMC-QE models generally make use of shared multiplexer server stations, single class and multiclass problems are treated with the same methodology.

Once the model is set up and represented by the Tableau(x), the evaluation, due to the high level of abstraction of equilibrium and the simplicity of the corresponding formulas, requires only a minimum of time. The same holds for variations of quantitative model parameters, being computable with the same Tableau(x).

Many problems from literature ([3], [6] [8], [10], [17] a.o.) have been modeled and evaluated e.g.

Open and Closed Queueing Networks [4], [14]
Multiclass Problems [4]
Process synchronization [9], [14]

as well as a first

large software system based on SAP Netweaver [20]

with high accuracy and neglectable computing time.

Even though the interpreter, MS-Excel, which has been used so far was sufficient to demonstrate the methodology, a more powerful toolset is under construction, which will improve the ease of use and the power of the modeling as well as of the evaluation.

Acknowledgements

The author would like to thank Stephan Kluth, M.Sc., Dipl.-Ing. Flavius Copaciu and Tomasz Porzucek, M.Sc. for valuable discussions and support in finishing the paper, as well as Raveendra Babu M. Tech. for various computer evaluations. Special thanks to my most honorable colleague Siegfried Wendt for all his inspiring input from the fundamental modeling side.

References

[1] S. Balsamo. "Product Form Queueing Networks" In *"Performance Evaluation"* G. Haring, et al. (Eds.), LNCS 1769, Springer, Berlin, Heidelberg, 2000, pp. 377 – 401
[2] S. Bernardi and J. Campos. "On performance bounds for interval Time Petri Nets" in Proceedings of the *"First International Conference on the Quantitative Evaluation of Systems 2004",* Enschede/NL 27.-30.09.2004, IEEE Computer Society, USA, 2004
[3] G. Bolch. "Leistungsbewertung von Rechesystemen", Teubner, 1989
[4] G. Bolch, St. Greiner, H. de Meer and K.S. Trivedi. "Queueing Networks and Markov Chains" John Wiley & Sons, New York a.o., 1998
[5] C. Girault and R. Valk. "Petri Nets for Systems Engineering" Springer, Berlin Heidelberg a.o. 1998
[6] D. Gross and C.M. Harris. "Fundamentals of Queueing Theorie" 3rd edition John Wiley & Sons, New York a.o., 1998
[7] Performance Evaluation Group, Dipartimento di Informatica, Universita di Torino "Great SPN-User's Manual (version 2.0.2)", 2005
[8] M. Haas and W. Zorn. "Methodische Leistungsanalyse von Rechensystemen" Oldenbourg, München, 1995
[9] C. Heitmeyer and N. Lynch. "The Generalized Railroad Crossing: A Case Study" in *"Formal Verification of Real Time Systems",* Proceedings of IEEE Real Time Systems Symposium, San Juan, PR 1994
[10] R. Jain. "The Art of Computer Systems Performance Analysis" John Wiley & Sons, New York a.o., 1991
[11] F.P. Kelly. "Reversibility and Stochastic Networks", John Wiley & Sons, New York a.o 1979
[12] A. Knöpfel, B. Gröne and P. Tabeling. "Fundamental Modeling Concepts - Effective Communication of IT Systems", John Wiley & Sons, New York a.o 2006
[13] L. Kleinrock. "Queueing Systems" Vol. 1 and 2, John Wiley & Sons, New York a.o. 1975
[14] M.A. Marsan, G. Balbo, G. Conte, S. Donatelli and G. Franceschinis. "Modelling with Generalized Stochastic Petri Nets", John Wiley & Sons, New York a.o., 1995
[15] B. Schroeder. A. Wierman, M. Harchol-Balter. "Closed versus open system models and their impact on performance and scheduling" to appear in *NSDI '06*
[16] P. Tabeling. "Softwaresysteme und ihre Modellierung" Springer, Berlin, Heidelberg, 2006
[17] J. Wang. "Timed Petri Nets", Kluwer Academic Pub., Boston a.o., 1998
[18] S. Wendt. "Nichtphysikalische Grundlagen der Informationstechnik" Springer, Berlin Heidelberg a.o., 1991
[19] S. Wendt. "Operationszustand versus Steuerzustand- eine äußerst zweckmäßige Unterscheidung", Internal Paper, Universität Kaiserslautern, Germany, 1998
[20] M. Grund, J.Klimle; S. Kühn; N. Naumann, R. Reicherdt, K. Spichale „Case Study ERMF Frame“ Bachelor Project Report, Hasso- Plattner-Institute, University of Potsdam, 2007

ANNEX: FORMULAS OF THE FMC-QE CALCULUS

Global Parameter Section	
$A^{z[1]} = N^{z}_{ext}{}^{e[1]}/\Delta t = [N^{z}_{ext}{}^{e[1]}]/\Delta T \le A^{z}_{max}{}^{[1]}$	Arrival Rate of unified external Service Requests from Source z (< for stochastic distr.)
$A^{z}_{max}{}^{[1]} = \min(B^{z}_{max,i}{}^{[bb(i,z)]})$	Bottleneck Throughput of Server System (dependant parameter)
$\{N^{z}_{total}{}^{[0]}\} = n^{z}_{total}{}^{[0]}$	total number of unified Service Requests in Universe
$m^{z}_{ext}{}^{[1]} = \{N^{z}_{total}{}^{[0]}\} - \{N^{z}_{sys}{}^{[1]}\}$	number if customers within the Infinite Server Source/Sink z (dependant Parameter!)
Common Section	
i	Index of Service Request/Server/Transistion (unique identifier within Server System)
bb ε (0,1,...)	hierarchical level 0- highest ("universe"), 1- ext. Source/Sink || Server System
Service Request Section	
$N^{z\,e[bb]}_{i} = 1\,[N^{z\,e[bb]}_{i}]$	unified Service Request
$[N^{z\,e[bb]}_{i}]$	Service Request Unit (semantic identifier)
$v^{z}_{i,int}{}^{[bb]} = \{v^{z}_{i,int}{}^{[bb]}\}*[N_{i}^{e[bb]}]/[N_{sup(i)}{}^{e[bb-1]}]$	local Service Request transformation coefficient in multiples of $[N^{z}_{i}{}^{[bb]}]$
$v^{z}_{i}{}^{[bb]} = v^{z}_{i,int}{}^{[bb]} * v^{z}_{sup(i)}{}^{[bb-1]}$ (recursion leaf i ...root)	global Service Request transformation coefficient in multiples of $[N^{z[1]}]$
<origin>::= C||**S**	external Customer or System internal Service Request
Server Section	
<server station identifier>	Type of Server Station (semantic)
<hierarchical type >::=	Server Station type within Server Tree structure
hSSt: H	hierarchical Server Station ("node"):
sH (≡ shSSt)	superordinate hierarchical Server Station [bb-1]
aH (≡ ahSSt)	abstracted hierarchical Server Station [bb] (same as directly embedded Stations)
BSSt: B	Basic Server Station ("leaf")
mB ≡ mBSSt	multiplexd Basic Server Station
MSSt: M	Multiplexer/Demultiplexer Server Station
<server configuration type> ::=	local Server configuration
SS||PS||IS||mpxed	Single/Parallel/Infinite/Mpx
<capacity>:: 1 || K || ∞	max. number of SRq within Service Station ("storage places")
$X^{z}_{i}{}^{[bb]}$	Service Time of Basic Server Station type i for a unified SRq
$m^{z}_{i,int}{}^{[bb]}$	local multiplicity of servers within Service Station $SSt^{z}_{i}{}^{[bb]}$ (= multiplexing coefficient for SS)
$m^{zIS}_{ext}{}^{[1]} = \{N^{z}_{total}{}^{[0]}\} - \{N^{z}_{sys}{}^{[1]}\}$	average number of infirnite Customer Server Stations (IS) in external Source/Sink
$m^{z}_{i}{}^{[bb]} = m^{z}_{i,int}{}^{[bb]}\; m^{z}_{sup(i)}{}^{[bb-1]}$ (recursion leaf i... roo	total multiplicity of servers within Service Station $SSt^{z}_{i}{}^{[bb]}$
M: $m^{z}_{mpx(i),int}{}^{[bb(i,z)+1]} = m^{z}_{mpx,i}{}^{[2]}$	multiplexer partition for multiplexed Basic Server Station $mBSSt^{[bb]}$
Service Rate B_i	Service Rate for Unified Service Request
BSSt: $B^{z}_{i}{}^{[bb]} = m^{z}_{i}{}^{[bb]}*[N^{z\,e[bb]}_{i}]/X^{z}_{i}{}^{[bb]}$	for all subordinate BSSt/hSSt on level [bb+1]
mBSSt: $B^{z}_{i}{}^{[bb]} = B^{z}_{mpx(i)}{}^{[bb(i,z)+1]}$	Service Rate of multiplexed Basic Server Station $mpxBSSt^{[bb]}$
hSSt: $B^{z}_{i}{}^{[bb]} = \min(B^{z}_{sub(i)}{}^{[bb+1]}/v^{z}_{sub(i),int}{}^{[bb+1]})$	for all subordinate BSSt/hSSt on level [bb+1]
$B^{z}_{max,i}{}^{[1]} = B^{z}_{i}{}^{[bb]}/v^{z}_{i}{}^{[bb]}$	max. external Arrival Rate from Source z **(Bottleneck Determination)**
Dynamic Evaluation Section	
Arrival Rates Ai	
$A^{z}_{i}{}^{[bb]} = v^{z}_{i}{}^{[bb]} * A^{z[1]}$	Arrival Rate in SSt (z.i)
$U^{z}_{i}{}^{[bb]} = A^{z}_{i}{}^{[bb]}/B^{z}_{i}{}^{[bb]}$	Utilization of Server Station SSt (z.i)
Control State variable n_i for BSSt:	
except for System SRq's, which are ignored (e.g. ACK Service Requests within communication protocols)	
Control State variable $n_{i,s}$	number of SRq in Server of SSt (z,i)
$n^{z}_{i,s}{}^{[bb]} = m^{z}_{i}{}^{[bb]} * U^{z}_{i}{}^{[bb]}$	
Control State variable $n_{i,q}$	number of SRq in Queue of SSt (z,i)
$n^{z}_{i,q}{}^{[bb]} = 0$ **D/D/m**	
$n^{z}_{i,q}{}^{[bb]} = U^{z}_{i}{}^{[bb]}/(1-U^{z}_{i}{}^{[bb]})$ **M/M/m**	
$n^{z}_{i,q}{}^{[bb]} = U^{z}_{i}{}^{[bb]}/(2*(1-U^{z}_{i}{}^{[bb]}))$ **M/D/m**	
M/M/m/K a.o.	(formulas are a bit too complex to be integrated here)
Control State variables n_i for hSSt:	
$n^{z}_{i,s}{}^{[bb]} = \Sigma n^{z}_{sub(i,z),s}{}^{[bb+1]}$	for all (i,z) directly subordinate Server Stations at Laxer [bb+1]
$n^{z}_{i,q}{}^{[bb]} = \Sigma n^{z}_{sub(i,z),q}{}^{[bb+1]}$	for all (i,z) directly subordinate Server Stations at Laxer [bb+1]
common formulas:	
$n^{z}_{i}{}^{[bb]} = n^{z}_{i,s}{}^{[bb]} + n^{z}_{i,q}{}^{[bb]}$	number of SRq in SSt (z.i)
$W^{z}_{i}{}^{[bb]} = N^{z}_{i,q}{}^{[bb]}/A^{z}_{i}{}^{[bb]}$	Waiting Time for a unified SRq in Queue of SSt (z,i)
$X^{z}_{i}{}^{[bb]} = N^{z}_{i,s}{}^{[bb]}/A^{z}_{i}{}^{[bb]}$	Service Time for a unified SRq in Queue of SSt (z,i)
$R^{z}_{i}{}^{[bb]} = N^{z}_{i}{}^{[bb]}/A^{z}_{i}{}^{[bb]}$	Response Time for a unified SRq in Server Station SSt (z,i) ("Elapsed Time")
Correctness Check	
$N^{z}_{total}{}^{[0]} = A^{z}_{uni}{}^{[0]} * R^{z}_{uni}{}^{[0]}$	(based on the redundancy of the Tableau)