Formal approach to QoS Specification and Verification[*][†]

J. de Meer, A. Rennoch, J. Burmeister

Abstract

The Quality of Sercvice (QoS) of advanced communication services and applications needs to be defined precisely and also it is necessary to verify that QoS statements hold. For both purposes some means of QoS languages are required. In this paper we propose extentions for the formal description language LOTOS (ISO 8807) with concepts of time consumption, resource allocation and behaviour probabilities. A new action denotation for synchronization in real environments has been introduced and a formal semantics for the time phases during synchronization is defined. Furthermore, we propose LOTOS for the definition of configurations and measurement scenarios in QoS verification experiments.

1 Introduction

Currently high-speed and multimedia services and protocols based on broadband communications are of great evidence in industy and academic reasearch & development and standardization. The introduction of the new services and protocol features provide a new quality in communication. But this also implies a broader scope for their verification. While the verification of OSI protocols is restricted to conformance only, today's verification goals include also Performance and Quality considerations:

a) (functional) Conformance
b) Performance
c) Qualification

The verification of advanced services and protocols Conformance is achieved by testing and yields the correctness of tested functions of the entity under test:
$Conformance\ [SUT\ \tilde{}\ (standardized)\ functions] \rightarrow pass/fail\ (verdict)$

However, the correct performance of a tested entity cannot be expected. Thus investigations must be taken on a second level of verification, to measure the operation of the entity of concern. The result of this activity will be measures and metrics of the tested functions under certain conditions:
$Performance\ [SUT\ \tilde{}\ environment\ conditions] \rightarrow measures\&metrics$

But to assess the quality of a performed function you need different types of application loads, in case of a network, or in case of a service different usage. For example, the throughput of a connection might be suitable for data transportation, but will obviously not fit with the quality requirements of an unidirectional transmission of video streams. Hence the quality of a service can only be assessed by taking the performance of an application into account.
$Quality\ [SUT\ \tilde{}\ application\ requirements] \rightarrow qualified/not\ qualified$

In some cases expectations on the SUT quality for certain applications are well known, e.g. derived from simulations, earlier experiences or even standardization. This may lead to a justification of the SUT quality based on a comparison of the measurements with the expected quality parameter values:
$Quality\ [SUT\ \tilde{}\ (standardized)\ expectations] \rightarrow qualified/not\ qualified$

2 OSI/QoS related Framework

The first working draft of the QoS Framework [1] has been issued taking into account previous outline and input documents, [2, 3, 5]. It's the intention of the framework to esthablish a QoS model

[*] The IBM contact for this paper is Jacob Slonim, IBM CAS, 895 Don Mills Road, North York, Ontario, Canada M3C 1W3.

[†] This work is supported by the CEC RACE II project R2088 TOPIC (Tool Set for Protocol and Advanced Service Verification in IBC Environments)

for OSI which permits a functional specification of a service in consideration of real facilities (e.g. physical resources, transmission lines, computer hardware and software, ...).

The scope of the QoS model include the identification of different participants (with their roles) in QoS, a terminology for QoS (with a basic set of important QoS characteristics in OSI) and mechanisms for a dynamic control and administration of QoS (QoS management functions). The QoS characteristics are considered to be interfaces between communicating objects (e.g. service user and provider objects).

A major part of the actual working draft introduces the various characteristics of the QoS categories temporality, accuracy, accountability, robustness, protection, capacity, speed, manageability and others. W.r.t. the description of QoS mechanisms in different architectures basic modelling elements are introduced (objects, object types, interfaces, role, viewpoint). The different relationships between these elements are represented in the basic QoS model and described by the ODP viewpoints. The basic model has been applied for OSI.

Tutorial material for an actual realisation of the proposed QoS framework for the support of distributed multimedia communications is available in an additional document [4]. The approach concentrate on an ATM network layer rather than the full range of multiservice networks. In the proposed QoS architecture an extract of traffic characteristics has been analysed w.r.t. their specific handling (e.g. scheduling, correction) in different layers (network, transport, orchestration, application platform). Layer specific QoS considerations have been applied to the video channel example.

The current work in SC21 project 57 is of high importance since OSI QoS terminology has to be used for any contribution on QoS verification methodology to ISO.

3 LOTOS Extensions

3.1 Identification of limitations and new QoS concepts

Considering the QoS categories introduced in [1], it is obvious that current LOTOS [6] does not meet all requirements needed. Existing approaches for the integration of QoS characteristics and real facilities in LOTOS (called LOTOS*) [8, 9] are proposing at least three new concepts:

- concept of time consumption,
- concept of resource allocation, and
- concept of action probabilities (quantified behaviour alternatives).

These concepts focus on the specification of QoS categories like capacity, robustness, accuracy, speed and temporality rather than others like protection, manageability or accountability.

The concept of time consumption will be realized by the introduction of time parameters of actions. It is the basic idea of the proposal that LOTOS* actions are not necessarily atomic events starting and ending at the same moment (as it is in standard LOTOS), since in real time environments some time may pass during the lifetime of an action. In this sense the time consuming action could be understood as a process. The time consumption of the specified action represents a time constraint for the termination of the action.

The concept of resource allocation will be introduced by resource parameters of actions. Actions may need an amount of resources to be available for their occurrence. The resources of an action are exclusively allocated to the action during its whole processing. Since they form an action prerequisite they could be understood as additional constraints to the action.

A third additional parameter to actions will provide the concept of action probabilities. In opposition to the concept of time consumption and resource allocation the introduction of probabilities in combination with behaviour alternatives allows the specification of statistical properties, e.g. error rates or availability requirements.

In the following we first present an execution model for LOTOS* (the so-called *gate machine*), and we introduce the specification concepts of LOTOS* (its syntactical form of parameterization and its semantics). Secondly we present a practical approach to generate QoS verification scenarios for QoS verification experiments. Since this work is under development, the current state of the art is provided.

3.2 Execution Model for standard and extended LOTOS

Since in real environments the synchronization of actions is a time consuming process a certain consideration this is needed. Specially a decision process in case of alternative actions (to be synchronized) has to be taken into account. Our underlying model for real time synchronization is based on an *offer-and-commit* protocol between the communicating processes and the decision process, the so-called *gate machine*. Actions to be synchronized at gates will be first offered to its related (even different) gate machines. A gate machine is collecting all action offers from the different processes (which participate at the gate) and replies with *ready* as soon as action offers might be matched. In a second phase the participating processes have to commit the selected action ready for synchronization. Finally the gate machine indicates the termination of the time consuming action (see figure 1).

It has to be stated that standard LOTOS provides no means at least for the concepts of time intervals and probabilities. A simple approach for the integration of new semantics properties in LOTOS is the interpretation of particular (added) action parameters like e.g. time consumption, resources requirements, and action probability. Each of these properties could be supervised by its appropriate semantics process connected to the gate machine. The domain of these processes is determined by the scope of the synchronizing processes participating at the same gate machine (see figure 2).

Following this approach a *time* process is needed which provides with a continuous stream of *tick* events whereby each time interval between two successive ticks is regarded as a predefined time unit. A *resource management* process represents the constraint of limited resource allocation. The action probability concept is devoted to a *probability* process.

According to the present proposals for new concepts for QoS specification within LOTOS a distinction could be made between deterministic QoS properties (e.g. time intervals and resource allocation) and statistical QoS properties (e.g. probabilities). Other properties like protection, manageability and accountability are left for future contributions.

3.3 Specification of QoS deterministic properties

QoS parameters related to the categories temporality or speed only (as timelines, transmit delay or time windows) have to be described in a time consuming model. They could be expressed by appropriate parameterized actions in LOTOS*. QoS parameters related to the category of resource allocatability (as buffer memory, processing capacity or bandwidth) could to be expressed as additional constraints on the action synchronization. Summarizing the constraints on actions, this implies an evolution of the actions in three dimensions (see figure 3) in opposite to a single dimension of action traces in standard LOTOS:

- progress in the behaviour tree (as in standard LOTOS),
- progress on the time scale, and
- variations of the resource pool.

Taking into account different specification requirements on QoS (e.g. timeout, resource allocation) a time consuming action needs not only an attribute for its interaction time but also: (a) a fixed time interval (wait state or delay) before the action will be offered after the previous action, and (b) a timeout value for the preparation phase (beginning with the action offer, including the allocation of resources, terminating with the interaction start).

Considering new applications in high-speed broadband environments QoS specifications are frequently using capacity parameters like bandwidth or buffer memory. Individual capacity parameters could be simply specified with standard data types (e.g. set or tuple) parameterized and restricted to particular intervals or selected numbers. Thus, the specification problem is not the definition of the related data type parameters,

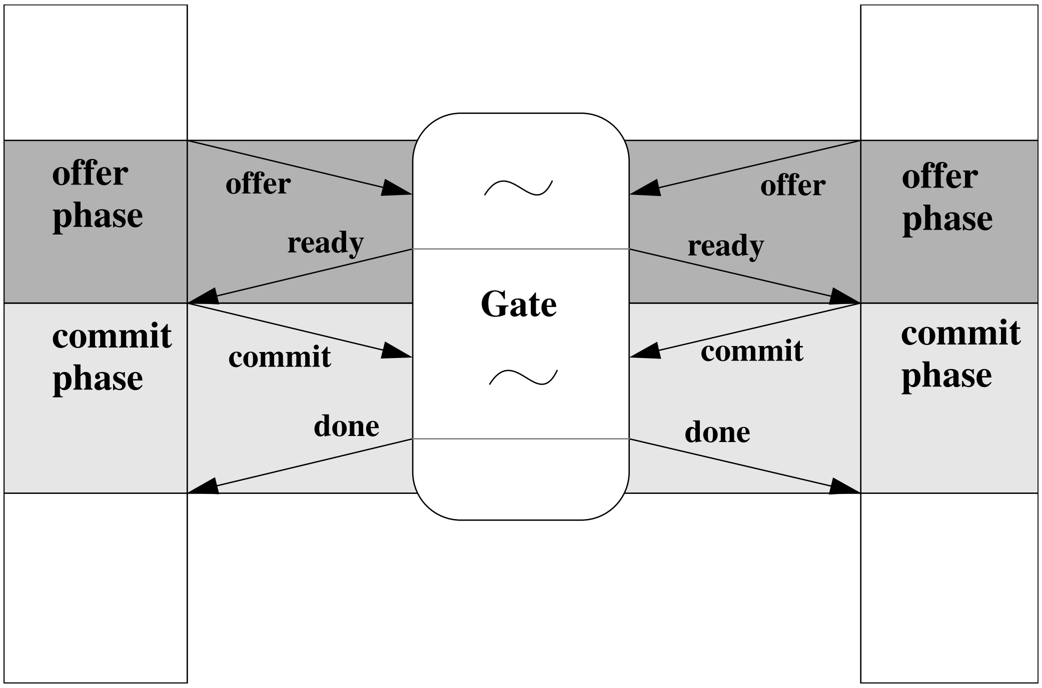

Figure 1: Offer-and-commit protocol

but the relationship between different - possibly concurrent - actions using same resources. Similar to the global time concept a global resource allocation concept is useful which permits the process of actions only if all resources needed are available.

Taking the requirements above into account, the following notation is proposed for a time and resource consuming action a in LOTOS* (see also figure 4):

$$a\{(t_w, t_p, t_i), r\},$$

where t_w means waiting time before action offering, t_p means maximum preparation time (offer-commit phases, resource request phase), and t_i means interaction time. The r attribute represent any set of requested resources.

Consider the example:

$$a\{(0,3,4), r_1\}; B_1[]error\{(3,x,y), r_2\}; B_2$$

The resulting semantics is the following: If the interaction of a does not start after 3 time units (whereby 0 means to start the preparation immediately) the interaction of the *error* alternative is applied (which is delayed by 3 time units combined with any preparation and interaction time). During the interaction time, either the resources r_1 or r_2 will be removed from the environment. Like in standard LOTOS, we claim for the behavioural semantics that the interaction phase cannot be interrupted, but only the waiting and preparation phase. Furthermore, we propose that the provided waiting and interaction time are consumed completely, but the preparation time represents a maximum time value, at least one time unit.

This kind of time consumption specification for actions implies two synchronization constraints in the following sense:

1. the interaction could synchronize only if both interaction times are equal or, if time intervals are used then they must have an overlapping; and

2. the preparation phases of the actions must overlap in realtime (execution).

For example:
$a\{(w_1, p_1, \mathbf{2}), r_1\}; B_1||a\{(w_2, p_2, \mathbf{2}), r_2\}; B_2$
$a\{(w_1, p_1, \mathbf{2}), r_1\}; B_1||a\{(w_2, p_2, \mathbf{i}), r_2\}[\mathbf{i} < \mathbf{3}]; B_2$

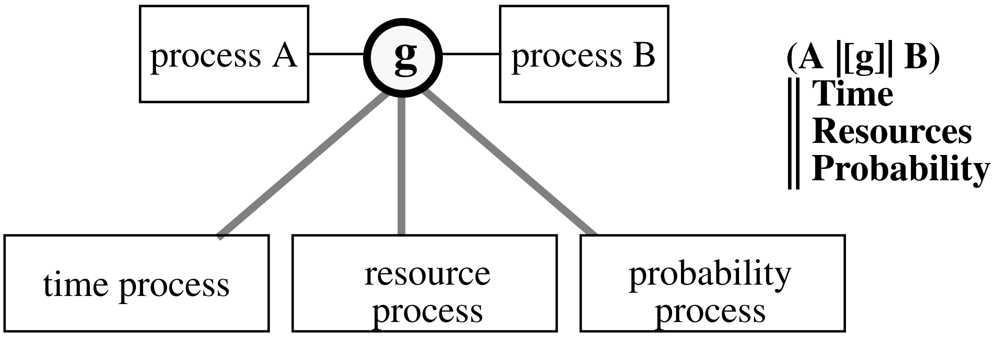

Figure 2: semantic processes

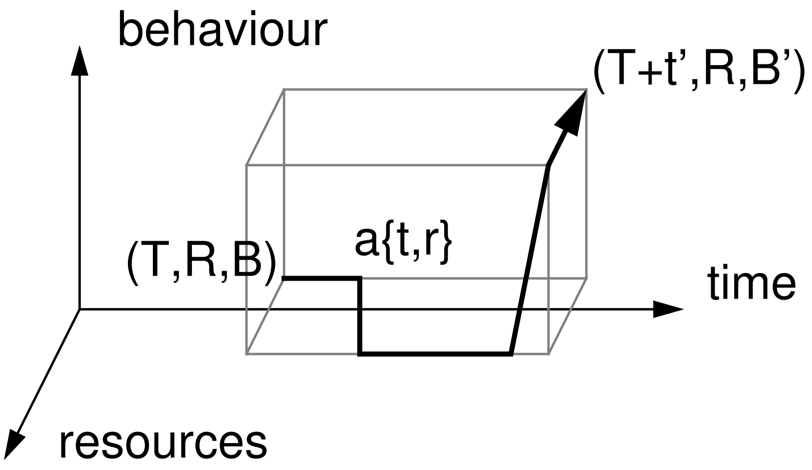

Figure 3: Three Dimensions of Actions

do match (if the resources r_1 and r_2 are available, **and** if p_1 and p_2 are overlapping according to the time state of the system), but
$a\{(w_1, p_1, \mathbf{2}), r_1\}; B_1 || a\{(w_2, p_2, \mathbf{1}), r_2\}; B_2$
$a\{(w_1, p_1, \mathbf{4}), r_1\}; B_1 || a\{(w_2, p_2, \mathbf{i}), r_2\}[\mathbf{i} < \mathbf{3}]; B_2$

do not match.

Formal Semantics

The related formal semantics are given in the following axioms and inference rules. Like standard LOTOS the extended formalism to be used for LOTOS* is also the labeled transition system. But we intend to break the atomic term of the action into smaller pieces of time units. In LOTOS*, the time unit will be the atomic basis for the formal semantics.

The axioms and inference rules provided by the LOTOS standard are describing the reduction of behaviour expression, e.g.

$$B - \mathbf{i} \rightarrow B'$$

represents an axiom (here it represents the semantics of the *internal event*).

A rule looks like:

$$\frac{B_1 - a \rightarrow B_2}{B - a \rightarrow B_2}$$

where B, B_1, B_2 denote the behaviour expressions, and where a denotes the action (including the gate identifier and the data constraints). The whole expression means that, if the axiom above the line holds, then the transition (and with that the axiom) below the line is valid.

Since we intend to describe the three dimensions (behaviour, time, resources), the new axioms and inference rules contain a triple due to the describtion of the time and resource requirements. Now, the action term contains also the time and resource parameter. For simplification, we do not consider the data type part, since we do not change this semantics of data matching. The following inference rules presume that the data part of an action allows this synchronization.

The basic form of the new inference rules is:

$$\frac{(T, R, a\{t, r\}; B) \text{ with } r \leq R}{(T, R, a\{t, r\}; B) - a\{t, r\} \rightarrow (T + t, R, B)}$$

where T (≥ 0) denotes the global time, R (≥ 0) denotes the set of resources (which will be returned after the action has been happened), B the behaviour expression, and $a\{t, r\}$ the action (whereby a is seen as the gate a provided with some data). All time values used in the following are non negative integers. Remember that t represents the time triple (t_w, t_p, t_i), and r represents the required resources.

Due to the time and resource consumption of actions we have to distinguish several different stages of the action: delay, preparation (with offer and commit, and allocation of resources), and communication (interaction). The transitions between the stages could be formalized by the following set of axioms:

a) action delay: At first, the delay time will be reduced without using any resources. The synchronization with parallel actions is not intended but synchronization with the global time:

$$\begin{array}{c}(T, R, a\{(t_w, t_p, t_i), r\}; B)\\ -a\{(1, t_p, t_i), r\} \rightarrow\\ (T + 1, R, a\{(t_w - 1, t_p, t_i), r\}; B)\end{array}$$

with $t_w > 0$

b) action start preparation: The delay time has been consumed, and now the resources will be allocated if available. Here, the synchronization with the resources (if available or not) is expressed in two axioms:

$$\begin{array}{c}(T, R, a\{(0, t_p, t_i), r\}; B)\\ -a\{(0, 1, t_i), r\} \rightarrow\\ (T + 1, R - r, a\{(0, t_p - 1, t_i), -r\}; B)\end{array}$$

with $t_p > 0$ and $0 \leq r \leq R$
and:

$$\begin{array}{c}(T, R, a\{(0, t_p, t_i), r\}; B)\\ -a\{(0, 1, t_i), r\} \rightarrow\\ (T + 1, R, a\{(0, t_p - 1, t_i), r\}; B)\end{array}$$

with $t_p > 0$ and $r > R$. The resources are available in the first case only. The negative value of r means the resources have been already allocated.

c) action preparation: After the resources are reserved the action will be prepared for synchronization:

$$\begin{array}{c}(T, R, a\{(0, t_p, t_i), r\}; B)\\ -a\{(0, 1, t_i), r\} \rightarrow\\ (T + 1, R, a\{(0, t_p - 1, t_i), r\}; B)\end{array}$$

with $r \leq 0$.

d) action interaction: Now the interaction runs (non interruptible):

$$\begin{array}{c}(T, R, a\{(0, 0, t_i), r\}; B)\\ -a\{(0, 0, t_i), r\} \rightarrow\\ (T + t_i, R, a\{(0, 0, 0), r\}; B)\end{array}$$

with $r \leq 0$be

e) action termination: After the interaction the resource will be released without time consumption:

$$\begin{array}{c}(T, R, a\{(0, 0, 0), r\}; B)\\ -a\{(0, 0, 0), r\} \rightarrow\\ (T, R - r, B)\end{array}$$

with $r \leq 0$

If the resources cannot be allocated during t_p time units, this action cannot be synchronized any longer (analogous to unmatching data in standard LOTOS). The semantics of actions which does not require any time and resources is equal to standard LOTOS.

The semantics of the parallel behaviour expression is given in the following inference rules. Before the actions can be synchronized, the delay time has to be consumed (which is indicated by $t_w = 0$). After that, both preparation times must be greater than zero if any resources are requested. The synchronization concept between

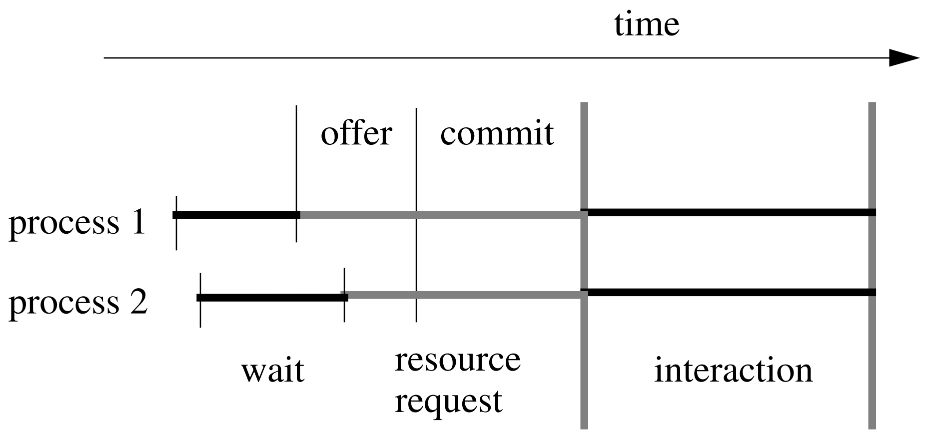

Figure 4: Action synchronization time phases

two processes is visualized in figure 4. For the following rules it holds that $a \in G$.

a) The delay time has to be consumed.
If

$$(T, R, B) = (T, R, B_1|[G]|B_2)$$

with

$a_1 = a\{\mathbf{1}, t_{1p}, t_{1i}), r_1\}$ and $a_2 = a\{(\mathbf{1}, t_{2p}, t_{2i}), r_2\}$

then

$$\frac{(\mathbf{T}, R, B_1) - a_1 \rightarrow (\mathbf{T+1}, R, B_1'),\quad (\mathbf{T}, R, B_2) - a_2 \rightarrow (\mathbf{T+1}, R, B_2')}{(\mathbf{T}, R, B) - a_1, a_2 \rightarrow (\mathbf{T+1}, R, B_1'|[G]|B_2')}$$

b) Since the equation $B_1|[G]|B_2 = B_2|[G]|B_1$ should be also valid for extended LOTOS, the following inference rule considers only the case, where the delay time of one (the left hand side) action is already consumed and the preparation phase has started. This could lead to a modified (reduced) resource pool (R').
If

$$(T, R, B) = (T, R, B_1|[G]|B_2)$$

with

$a_1 = a\{(\mathbf{0}, \mathbf{1}, t_1), r_1\}$ and $a_2 = a\{(\mathbf{1}, t_{2p}, t_{2i}), r_2\}$

then

$$\frac{(T, \mathbf{R}, B_1) - a_1 \rightarrow (T+1, \mathbf{R}', B_1'),\quad (T, \mathbf{R}, B_2) - a_2 \rightarrow (T+1, \mathbf{R}, B_2')}{(T, \mathbf{R}, B) - a_1, a_2 \rightarrow (T+1, \mathbf{R}', B_1'|[G]|B_2')}$$

c) This inference rule defines the behaviour after the delay time has been consumed and the resources have to be allocated. Four cases will be considered depending on the allocated and available resources. The case in which both behaviour expressions already have allocated their resources will be considered in d).
If

$$(T, R, B) = (T, R, B_1|[G]|B_2)$$

with

$a_1 = a\{(0, \mathbf{1}, t_{1i}), r_1\}$ and $a_2 = a\{(0, \mathbf{1}, t_{2i}), r_2\}$

and $r_1 > 0$ or $r_2 > 0$
then

$$\frac{(T, \mathbf{R}, B_1) - a_1 \rightarrow (T+1, \mathbf{R_1'}, B_1'),\quad (T, \mathbf{R}, B_2) - a_2 \rightarrow (T+1, \mathbf{R_2'}, B_2')}{(T, \mathbf{R}, B) - a_1, a_2 \rightarrow (T+1, \mathbf{R}', B_1'|[G]|B_2')}$$

with
$R' = R_1'$ if $r_2 \leq 0$ or
$R' = R_2'$ if $r_1 \leq 0$ or
$R' = R$ if $(r_1 + r_2) > R$ and $r_1, r_2 > 0$, (i.e. not enough resourres available),

$R' = R - (r_1 + r_2)$ if $r_1, r_2 > 0$ and $(r_1 + r_2) \leq R$ (all required resources are allocated).

d) After the resources have been allocated the synchronization starts immediately (indicated by $r_1, r_2 \leq 0$ and that T will not be incremented which leads to the reduction of the axioms). In other words: the unused preparation time is deleted.
If

$$(T, R, B) = (T, R, B_1|[G]|B_2)$$

with

$$a_1 = a\{(0, 1, t_{1i}), r_1\} \text{ and } a_2 = a\{(0, 1, t_{2i}), r_2\}$$

and $r_1, r_2 \leq 0$
then

$$\frac{\begin{array}{c}(T, R, B_1) - a_1 \rightarrow (T+1, R, B_1'),\\ (T, R, B_2) - a_2 \rightarrow (T+1, R, B_2')\end{array}}{(\mathbf{T}, R, B) - a_1, a_2 \rightarrow (\mathbf{T}, R, B_1'|[G]|B_2')}$$

and, in the case that the preparation time of a_1 has been elapsed, the processing of B_1 is postponed until the preparation time of a_2 is zero, too (remember also that $B_1[G]|B_2 = B_2|[G]|B_1$).
If

$$(T, R, B) = (T, R, B_1|[G]|B_2)$$

with

$$a_1 = a\{(0, \mathbf{0}, t_{1i}), r_1\} \text{ and } a_2 = a\{(0, \mathbf{1}, t_{2i}), r_2\}$$

and $r_1, r_2 \leq 0$
then

$$\frac{\begin{array}{c}(T, R, \mathbf{B_1}) - a_1 \rightarrow (T+1, R, \mathbf{B_1'}),\\ (T, R, B_2) - a_2 \rightarrow (T+1, R, B_2')\end{array}}{(\mathbf{T}, R, B) - a_1, a_2 \rightarrow (\mathbf{T}, R, \mathbf{B_1}|[G]|B_2')}$$

e)The interaction itself is not interruptable, which is indicated by using t_i instead of a single time unit step. Please notice that t_i is identical in a_1 and a_2.
If

$$(T, R, B) = (T, R, B_1|[G]|B_2)$$

with

$$a_1 = a\{(0, 0, \mathbf{t_i}), r_1\} \text{ and } a_2 = a\{(0, 0, \mathbf{t_i}), r_2\}$$

and $t_i > 0$, $r_1, r_2 \leq 0$
then

$$\frac{\begin{array}{c}(\mathbf{T}, R, B_1) - a_1 \rightarrow (\mathbf{T} + \mathbf{t_i}, R, B_1'),\\ (\mathbf{T}, R, B_2) - a_2 \rightarrow (\mathbf{T} + \mathbf{t_i}, R, B_2')\end{array}}{(\mathbf{T}, R, B) - a_1, a_2 \rightarrow (\mathbf{T} + \mathbf{t_i}, R, B_1'|[G]|B_2')}$$

After the synchronization ($t_i = 0$) the resources are reallocated (in this current model without time consumption in this current model):
If

$$(T, R, B) = (T, R, B_1|[G]|B_2)$$

with

$$a_1 = a\{(0, 0, 0), \mathbf{r_1}\} \text{ and } a_2 = a\{(0, 0, 0), \mathbf{r_2}\}$$

and $r_1, r_2 \leq 0$
then

$$\frac{\begin{array}{c}(T, R, B_1) - a_1 \rightarrow (T, R_1', B_1'),\\ (T, R, B_2) - a_2 \rightarrow (T, R_2', B_2')\end{array}}{\begin{array}{c}(\mathbf{T}, \mathbf{R}, B)\\ -a_1, a_2 \rightarrow\\ (\mathbf{T}, \mathbf{R} + (-\mathbf{r_1}) + (-\mathbf{r_2}), B_1'|[G]|B_2')\end{array}}$$

The other operators like disabling and choice, and the parallelism in case of that $a \not\in G$, are for further study, but the inference rules may be developed similar to that of the parallel operator. Specially, the choice operator will handle a third attribute (probability) as presented in the following.

3.4 Specification of QoS statistical properties

Most QoS specifications make usage of statistical statements on accuracy and robustness qualities, e.g. error rates, availability or mean times (e.g. MTBF). The common characteristic of these parameters is a quantification of alternative behaviour in opposite to the concept of non-determinism in LOTOS. Since different behaviour paths in LOTOS processes should occur according to particular frequencies only they have to be marked with probability attributes. For the sake of e.g. maximum error rates they have to be constraint by maximum limits. Although in different proposals the syntax for the definition of probabilistic choices are different the intended semantics is common according to the following[9]:

```
    <p1>a1;B1
[]  <p2>a2;B2        -<pi>ai->  Bi
...
[]  <pn>an;Bn
```

The arising question is concerned with the specification of these semantics properties in standard LOTOS. First, it should be clear that a decision on the behaviour path to be selected depends on the (local) probabilities of the behaviour alternatives but also on the probabilities of the corresponding action to be synchronized. Therefore any selection of probabilistic behaviour could be done only by considering all relevant probabilistic rates. Consider two communication processes A and B both offering two behaviour alternatives "a" and "b" marked with some probabilities: see figure 5.

It is the basic idea for the specification of the probability semantics to use LOTOS abstract data types. The equal stochastic distribution is represented by the non- deterministic choice of an element from a set. If "P" is the mathematical probability function and "e IsIn S" denotes the LOTOS data type operator meaning "e is element of set S" then:

for all elements of S: *P("e IsIn S") = constant.*

The integration of the probability attribute and the time and resource attributes defined in the previous subsection is left for further study.

4 Derivation and Specification of QoS verification scenarios

Since Network Performance (NP) influence QoS it is clear that QoS statements from service suppliers are depending on specific NP. Thus QoS verification also has to be undertaken under well defined NP conditions (constraints). In the following the pair "QoS parameters - NP parameters" is called basic 'QoS verification statement' (QoS VS) and means: all QoS parameter values (or parameter ranges, mean values) specified are to be verified after the network have been tuned according to the NP parameters specified (see figure 6).

Initial basic QoS verification statements could be derived by transferring general NP/QoS mapping rules (provided NP, provided QoS by the SUT) to the specific service/network situation to be verified. For instance, consider an ATM network used by the service to be verified:

The mapping rule (see also [7])

InformationLossQoS
= f3 (InformationErrorNP, InformationLossNP, PrematureTerminationNP)

could be derived to the QoS verification statement:

```
DataIntegrity (>= 0.99)
~
(CRER <= 0.01, CLR <= 0.001, RTD <= x),
```

using the NP parameters Cell Residual Error Ratio (CRER), Cell Loss Ratio (CLR) and Round Trip Delay (RTD). It is clear that only those QoS and NP parameters will remain in a final QoS verification statement which are measurable w.r.t. the availability of measurement tools.

A generic specification of this small QoS verification statement using the FDT LOTOS and a constraint oriented specification style is given here:

```
process verification-statement :=
      user1 [s] (0.99)
|[s]| service [s,n]
|[n]| network [n] (0.01, 0.001, x).
```

The "user" process is parameterized with the QoS expectations at the SAP and the "network" process is defined by network conditions to be tuned. The "service" process might be parameterized by service specific options (e.g. special service modes) to be established during e.g. service installation. During the verification experiment QoS and NP parameters of the "user" and "network" processes will be considered by measurement and tuning tools, i.e. there exist a correspondence between these parameters and the tools (see figure 7).

It is obvious that additional time constraints (e.g. experiment duration 1 h, or 100 reiterations) are necessary and additional service users or network background loads (e.g. well defined workload processes) are desirable. A complete set of QoS expectations and constraints supposed for an QoS verification experiment is called "QoS verification statement". A QoS verification statement could be described with LOTOS in the following way:

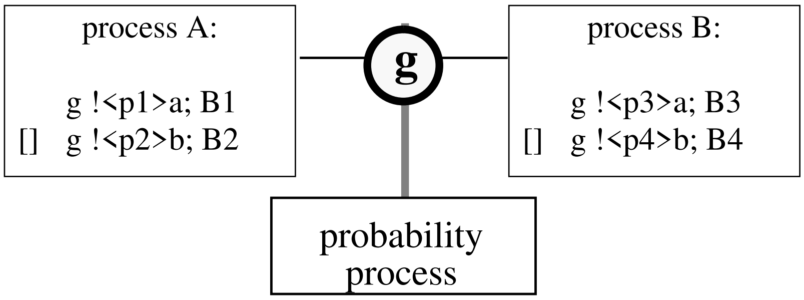

Figure 5: probability process

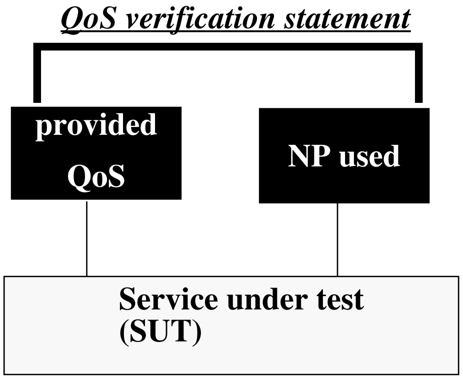

Figure 6: QoS verification statement

```
   start-timer [t] (1 h)
>> (statement1 [s,n] [> timeout [t])

where
process statement1 [s,n] : noexit :=
      (user1[s](..) ||| user2[s](..))
|[s]| (service[s,n] ||| networkload1[n])
|[n]| network [n] (..)
endproc
```

In comparison with protocol conformance test suites QoS verification statements (QoS VS) are similar to PCT test bodies, i.e. the addition of some preamble and postamble sequences together with the QoS VS represents a complete test scenario for QoS test tools. Preambles could be the establishment of a certain service status, or it could be also the initiation of network load. These are representing the conditions of the QoS measurement.

Any coordination between the verification equipment components (e.g. for temporal disconnection of a workload process, simulation of any "rush hour") belongs to the realm of a FDT and should be specified by using the FDT. The realization of this approach requires an FDT based "synchronization machine" for the control of the individual measurement and tuning tools.

5 Concluding remarks

QoS verification statements could be described using any informal or formal languages. How-

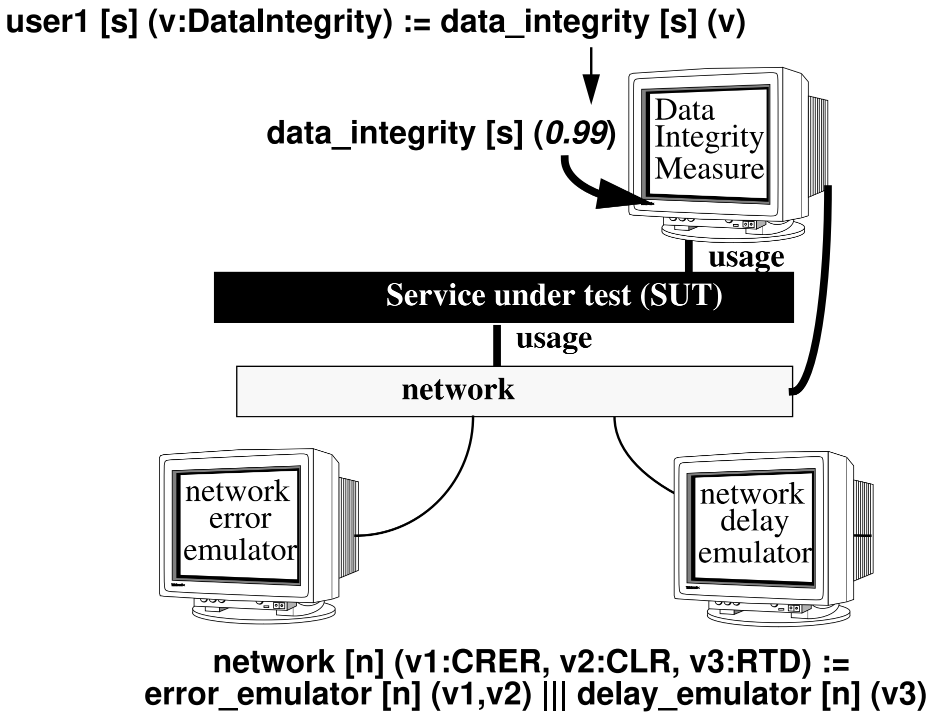

Figure 7: QoS verification statement

ever, the major advantages in using an FDT (e.g. LOTOS) are the:

1. consideration of all subjects of a QoS verification statement using a formal and standardized syntax,
2. independence of the formal QoS verification statement from implementation details (presupposition for all standardized test languages),
3. formal specification of the logical configuration of the QoS verification experiment by representing each distributed IUT, measurement and tuning component,
4. formal specification of the internal coordination between QoS verification experiment equipment components; to be used in combination with an FDT based synchronization service which realizes the specified coordination of distributed components (based on the formal description),
5. common approach for a description language for the IUT specification, protocol conformance test suites and QoS verification statements; it allows common usage of general experiment tools (e.g. synchronization service) and provides a base for a (semi-) automatic derivation of QoS verification statements from the formal IUT specification (cp. PCT derivation from formal protocol specification),
6. first step towards a detailed formal specification of the IUT and its environment for simulation purposes.

Following the formal representation of QoS verification statements (QoS VS), further work is needed on the definition of processes to be used in QoS VS. They have to be defined w.r.t. QoS or NP parameters measurable by real experimental tools. E.g. the "data-integrity" process introduced above is intended to represent a threshold control process, i.e. it compares actual measurements on the data integrity with the specified threshold values. In (standard) LOTOS this semantics could be specified by:

```
process data-integrity[s]
          (th:DataIntegrity) : exit :=
s !experiment-tool1 ?act:DataIntegrity;
(   [act < th]-> ...; data-integrity[s](th)
 [] [act >= th]-> ...; exit
)
endproc
```

During the QoS verification experiment a permanent control and possibly some tuning of the NP is necessary to keep the network constraints of the QoS verification statement.

About the authors

The authors are researchers of the System-Engineering and Methods (SEM) group at the Research Center for Open Communication Systems (FOKUS) in Berlin, Germany.

Within FOKUS the SEM group concentrates on the application of formal methods in advanced communication systems and Open Distributed Processing (ODP).

Their address is: GMD-FOKUS, Hardenbergplatz 2, D-10623 Berlin, FR Germany. Phone: +49 30 25499-239, fax: +49 30 25499-202, email: jdm@fokus.berlin.gmd.d400.de.

References

[1] ISO SC21 N7521: Quality of Service Framework - Working Draft 1, ANSI, New York 1992.

[2] ISO SC21/WG1 N1145: Quality of Service Framework - Outline, BSI, London 1992.

[3] ISO SC21/WG1 N1146: User Requirements for Quality of Service, BSI, London 1992.

[4] ISO SC21/WG1 N1201: A suggested QoS architecture for multimedia communications, BSI, London 1992.

[5] ISO TR 12178: Identifying user requirements for systems surporting time critical systems, ISO, Geneva 1992.

[6] ISO IS 8807, Information processing systems – Open Systems Interconnection – LOTOS – A formal description technique based on the temporal ordering of observational behaviour, 1989.

[7] R1082/KT/LM/DS/B/013/b1, QOSMIC Deliverable 1.3C, QoS and Performance Relationship, Juni 1992.

[8] Ina Schieferdecker, Adam Wolisz: Operational Semantics of Timed Interacting Systems: A Performance Oriented Approach to the Algebraic Specification of Communication Protocols, Technical Report, Technical University of Berlin, 1992.

[9] Nathalie Rico, G.v.Bochmann: Performance description and analysis for distributed systems using a variant of LOTOS. in: Protocol Specification, Testing, and Verification XII, Stockholm 1991.