INTERNATIONAL STANDARD

ISO/IEC 14756

First edition
1999-11-15

Information technology — Measurement and rating of performance of computer-based software systems

Technologies de l'information — Mesurage et gradation de la performance des systèmes de logiciels d'ordinateurs

Reference number
ISO/IEC 14756:1999(E)

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 14756 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software engineering*.

Annexes A to D form an integral part of this International Standard. Annexes E and F are for information only.

Introduction

In both the planning and using of data processing systems, the speed of execution is a significant property. This property is influenced greatly by the efficiency of the software used in the system. Measuring the speed of the system as well as the influence of the efficiency of the software is of elementary interest.

In order to measure the influence of software on the time behaviour of a data processing system it is necessary to measure the time behaviour of the whole system. Based on the metrics of the measurement procedure proposed in this standard it is possible to define and to compute the values of the time efficiency of the software.

It is important that time behaviour characteristics are estimated in a reproducible way. Therefore it is not possible to use human users in the experiment. One reason is that human users cannot reproduce longer phases of computer usage several times without deviations in characteristics of usage. Another reason is that it would be too expensive to carry out such experiments with human users if the job or task stream comes from many users. Therefore an emulator is used which emulates all users by use of a second data processing system.

This means that measurement and rating of performance according to this International Standard needs a tool. This tool is the emulator which shall work according to the specifications of this standard. It has to be proven that the emulator used actually fulfils these specifications.

All relevant details of this experiment are recorded in a logfile by the user emulator. From this logfile the values which describe the time behaviour (for instance response times and throughput values) can be computed. From these performance values the software efficiency rating values will be computed.

Not all of these values are always necessary to carry out a measurement and rating procedure. For instance if a simple workload having only a few interactive task types or only a simple sequence of batch jobs is used, then only a small subset of all terms and values which are defined is required. This method also allows the measuring and rating of a large and complex computer-based software system (CBSS) processing a complex job or task stream which is generated by a large set of many different users. As far as it is necessary the definitions include mathematical terms. This is in order to obtain an exact mathematical basis for the computations of performance and rating values and for checking the correctness of the measurement run and rating steps as well as for the (statistical) significance of the performance values and rating results.

The result of a measurement consists of the calculated performance values. These are throughput values and execution time values. The final result of performance assessment of a CBSS consists of the rating values. They are gained by comparing the calculated performance values with the user's requirements. In addition it is possible - if desired - to rate the performance values of the CBSS under test by comparing them with those of a reference CBSS (for instance having the same hardware configuration but another version of the application program with the same functionality).

The result of the rating procedure is a set of values, each being greater than, less than or equal to 1. The rating values have the meaning of "better than", "worse than" or "equal to" the defined requirements (or the properties of a second system under test used as a reference). The final set of rating values assesses each task type which are defined separately in the workload.

Annexes E and F contain software as well as special data that are not printable. Therefore they are delivered on the CD-ROM which constitutes this International Standard. A short overview is provided in both annexes.

Information technology – Measurement and rating of performance of computer-based software systems

Section 1: General

1 Scope

This International Standard defines how user oriented performance of computer-based software systems (CBSS) may be measured and rated. A CBSS is a data processing system as it is seen by its users, e.g. by users at various terminals, or as it is seen by operational users and business users at the data processing center.

A CBSS includes hardware and all its software (system software and application software) which is needed to realize the data processing functions required by the users or what may influence to the CBSS's time behaviour.

This International Standard is applicable for tests of all time constrained systems or system parts. Also a network may be part of a system or may be the main subject of a test. The method defined in this International Standard is not limited to special cases like classic batch or terminal-host systems, e.g. also included are client server systems or, with a broader comprehension of the definition of 'task', real time systems. But the practicability of tests may be limited by the expenditure required to test large environments.

This International Standard specifies the key figures of user oriented performance terms and specifies a method of measuring and rating these performance values. The specified performance values are those which describe the execution speed of user orders (tasks), namely the triple of:

- execution time,
- throughput,
- timeliness.

The user orders, subsequently called tasks, may be of simple or complex internal structure. A task may be a job, transaction, process or a more complex structure, but with a defined start and end depending on the needs of the evaluator. When evaluating the performance it is possible to use this International Standard for measuring the time behaviour with reference to business transaction completion times in addition to other individual response times.

The rating is done with respect to users requirements or by comparing two or more measured systems (types or versions).

Intentionally no proposals for measuring internal values, such as:

- utilisation values,
- mean instruction rates,
- path lengths,
- cache hit rates,
- queuing times,
- service times,

are given, because the definition of internal values depends on the architecture of the hardware and the software of the system under test. Contrary to this the user oriented performance values which are defined in this International Standard are independent of architecture. The definition of internal performance values can be done independently from the definition of user oriented performance values. They may be used and can be measured in addition to the user oriented performance values. Also the definition of terms for the efficiency with which the user oriented values are produced can be done freely. In addition this International Standard gives guidance on how to establish at a data processing system a stable and reproducible state of operation. This reproducible state may be used to measure other performance values such as the above mentioned internal values.

This International Standard focuses on:

- application software;
- system software;
- turn-key systems (i.e. systems consisting of an application software, the system software and the hardware for which it was designed);
- general data processing systems.

This International Standard specifies the requirements for an emulation (by a technical system - the so-called remote terminal emulator (RTE) - of user interactions with a data processing system. It is the guideline for precisely measuring and rating the user oriented performance values. It provides the guideline for estimating these values with the required accuracy and repeatability of CBSSs with deterministic as well as random behaviour of users. It is also a guidance for implementing a RTE or proving whether it works according to this International Standard.

This International Standard provides the guideline to measure and rate the performance of CBSS with random user behaviour when the accuracy and repeatability is required. It specifies in detail how to prepare and carry out the measurement procedure. Along with a description of the analysis of the measured values, the formulas for computing the performance value and the rating value, are provided.

This International Standard also gives guidance on:

- how to design a user oriented benchmark test using a:
  - transaction oriented workload,
  - batch oriented workload,
  - or transaction and batch mixed workload.

It specifies:

- how to describe such a workload,
- how to perform the measurement procedure,
- how to rate the measured results.

This International Standard is of interest to:

- evaluators,
- developers,
- buyers (including users of a data processing system),
- system integrators

of CBSSs.

NOTE 1 The field of application of this International Standard may be extended to include the following aspects. Workloads fulfilling the specifications of this standard and having a sufficiently general structure may be used as standard workloads. They may be used to measure and rate performance of data processing systems used in specific fields. E.g. a standard workload for word-processing may be used to compare the time efficiency of different software products or different versions of the same product running on the same hardware system. Such a standard workload may also be used if always applying the same application software version and the same hardware to compare the efficiency of the system software. When applying the same application software and workload to different systems, consisting of hardware and system software, as normally sold by system vendors, the efficiency of the data processing systems may be compared with respect to the application and workload used.

2 Conformance

Rating a software system without comparing to another can be done following the rules of this International Standard by rating against user requirements. In case of comparing performance values developed through the use of this International Standard, the comparisons depends upon equivalent functions in the compared systems. The values are most useful when comparing different releases or platform versions of the same software system, or comparing software systems which are known to have equivalent functions, or comparing hardware by using software with equivalent functions. The values are not useful for comparing software systems which do not have known equivalency.

To conform to this International Standard the requirements in

- subclauses 6.1 and 10.1 for descriptions of the configuration including the system under test,
- subclauses 6.2 and 10.2 and annexes A and C for user emulation,
- subclauses 6.3, 11, 12 and annex D for measurement procedures,
- clauses 7, 14 and annex B for calculating performance values,
- clauses 8, 9 and 15 for rating procedures

shall be fulfilled.

For results of a measurement in addition all requirements in this International Standard shall be fulfilled and the tests in 6.4 and - in more detail - in clause 13 shall be carried out successfully without any ensuing failures. It is the responsibility of the tester to submit proof of the results of the measurements done in accordance with this International Standard. Therefore the tester should supply additional documents of their own choice in addition to the documents requested in this International Standard, which are suitable to repeat the measurement by a third party to attain the same results.

3 Normative reference

The following normative document contains provisions which, through reference in this text, constitute provisions of this International Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent edition of the normative document indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 14598-1:1999, *Information technology — Software product evaluation — Part 1: General overview.*

4 Definitions

For the purposes of this International Standard, the following definitions apply.

4.1 activity: An order submitted to the system under test (SUT) by a user or an emulated user demanding the execution of a data processing operation according to a defined algorithm to produce specific output data from specific input data and (if requested) stored data.

4.2 activity type: A classification of activities defined by the execution of the same algorithm.

4.3 chain: One or more tasks submitted to the SUT in a defined sequence.

4.4 chain type: A classification of chains which is defined by the sequence of tasks types.

NOTES

2 The emulated users submit only chains of specified chain types to the SUT.

3 The mathematical symbol for the current number of the chain type is *l*. The mathematical symbol for the total number of chain types is *u*.

4.5 computer-based software system (CBSS): A software system running on a computer.

NOTE 4 A CBSS may be a data processing system as seen by human users at their terminals or at equivalent machine-user-interfaces. It includes hardware and all software (system software and application software) which is necessary for realizing data processing functions required by its users.

4.6 emulated user: The imitation of a user, with regard to the tasks he submits and his time behaviour, realized by a technical system.

4.7 execution time: The time which elapses between task submission and completion.

NOTES

5 In case of a task is representing a batch job the execution time is the elapsed time for the completion of the job. In case of an interactive task the execution time is the response time (submit until complete response) of the task.

6 The reader should notify that the definition (and therefore the measurement and rating) of the begin and end of the execution time does not depend on the task mode. In case of "NO WAIT" task mode for the following task it is possible and solved by the method described in this Intentional Standard, that the execution time of the following task may overlap one or more preparation and execution times of subsequent tasks of the same emulated user.

4.8 mean execution time: The mean value of all execution times of tasks of the j-th task type which were submitted within the rating interval.

NOTE 7 The mathematical symbol is $T_{ME\ (j)}$ corresponding to the j-th task type.

4.9 mean execution time rating value: The quotient (corresponding to the j-th task type) of the mean execution time reference value and the measured mean execution time.

NOTE 8 The mathematical symbol is $R_{ME\ (j)}$ corresponding to the j-th task type.

4.10 mean execution time reference value: The mean execution time maximally accepted by the emulated user. It shall be computed from the workload parameter set (see 9.1.1, 15.2.1 and clause B.1).

NOTES

9 The mathematical symbol is $T_{Ref\ (j)}$ corresponding to the j-th task type.

4.11 observation period: The time interval, where the measurement procedure is observed for collecting (logging) measurement results for rating or validation, consisting of the rating interval and the supplementary run.

4.12 preparation time: The time which elapses before the task submission. The event of starting the preparation time depends on the definition of the task mode of the following task. The preparation time value is the random chosen representation of an distributed variable with a defined mean and a standard deviation. They depend on the task type of the following task and the type of the emulated user generating the task.

NOTES

10 The preparation time starts with the preceding task completion of the same emulated user if task mode of the following task equals 1, it starts with the preceding task submission if task mode equals 0, (see definition of task mode), regardless whether the preceding task belongs to the same chain or to the preceding chain.

11 The mathematical symbol of the mean preparation time is *h(i,j)* with its standard deviation *s(i,j)* corresponding to the i-th user type the j-th task type.

4.13 rating interval: The time interval of the measurement procedure from the time the SUT reaches a stable state of operation to the time the measurement results are fulfilling the required statistical significance.

NOTE 12 The mathematical symbol of the duration is T_R.

4.14 relative chain frequency: The relative frequency of using the l-th chain type by an emulated user of the i-th type.

NOTE 13 The mathematical symbol is *q(i,l)* corresponding to the i-th user type and the l-th chain type.

4.15 remote terminal emulator (RTE): A data processing system realizing a set of emulated users.

4.16 stabilization phase: The time interval of the measurement procedure when the RTE starts submitting tasks until the SUT reaches a stable state of operation.

4.17 supplementary run: The time interval of the measurement procedure from the time the measurement results fulfil the required statistical significance to the time when all tasks, which were submitted during the rating interval, are completed.

4.18 system under test (SUT): The parts of the CBSS to be tested. All components which may influence the SUT's time behaviour shall be part of the SUT and if the influence depends on some workload, this workload shall be represented by the RTE too. Except if an influence to the time behaviour is impossible or not evident the emulation of the parts beside may be omitted.

NOTE 14 The SUT may consist of hardware, system software, data communication features or application software or a combination of them. Testing a part of a system, means that all parts of the system, which may influence the time behaviour of the part to be tested, are an integral part of the SUT, e.g. testing the time behaviour of one host application beside others on the same host, the workload of all applications have to be defined and emulated by the RTE with a representative workload parameter set.

4.19 task: The combination of:

- a specific activity;
- a demanded execution time, defined by a specific timeliness function;
- a specific task mode.

4.20 task completion: Timely event when for a specific task the total output string or, in case of a set of output strings, all parts are completely received by to the emulated user or another instance. If the task does not submit an output to the user (i.e.: during the measurement: to the RTE) a functionality, producing an „artificial output", indicating the task completion, may be added to the task for usage during the measurement.

NOTES

15 The time of task completion defines the end time of the preceding preparation time and the begin time of the execution time of the following task.

16 The correctness of the received output is validated by checking the computational result of the task (see 6.4.3, and in more detail 10.3.1 and 13.1).

4.21 task mode: Indication of whether the user's preparation time begins immediately with the task submission of the preceding task (*value = 0*, i.e. "NO WAIT") or begins when the preceding task has been completed (task completion) (*value = 1*, i.e. "WAIT").

NOTES

17 This is not standard usage mode of “Dialog” or “Batch” in UNIX based systems.

18 The mathematical symbol of the task mode is *M(j)* corresponding to the j-th task type.

4.22 task submission: Timely event when the input string is completely submitted from the emulated user to the SUT and the execution of the task may start, regardless if the SUT starts the execution immediately or not. The task submission is also indicated by the time when the action of the emulated user for this tasks ends. Following this International Standard, the task submission shall not be defined (and measured) by the event, when the SUT has received or interpreted the input string or when a receipt string, which may be send by the SUT after receiving and interpreting the input string, is received by the emulated user.

NOTE 19 Normally the task submission is defined internally by the submission of a special character (e.g. Carriage Return) or a character sequence at the end of the input string or at the end of several parts of the input string. Also it often happens that the task submission event is defined by the submission of the last character of an specified number characters in a string. For a classic batch task the task submission may be defined by the submission of the last character of the last string of the batch command sequence.

4.23 task type: A classification of tasks which is defined by the combination of:

- the activity type, or
  a set of activity types which are all belonging to an identical timeliness function and task mode;
- the timeliness function;
- the task mode.

NOTES

20 Emulated users submit only these types of tasks to the SUT.

21 The mathematical symbol for the current number of the task type is *j*. The total number of task types is *m*.

4.24 throughput: The rate (i.e. the average number per time unit with respect to the rating interval) of all tasks of a task type submitted to the SUT.

NOTES

22 The mathematical symbol is *B(j)* corresponding to the j-th task type.

23 Usually throughput is defined by the rate of terminated tasks during a period of time. In order to specify a supplementary run and not a “heads run” in this standard, throughput is defined in this standard by the rate of tasks submitted. Nevertheless the correct task completion is validated by checking the computational result of the task.

4.25 throughput rating value: The quotient (corresponding to the j-th task type) of the (actual) throughput and the throughput reference value.

NOTE 24 The mathematical symbol is $R_{TH}\ (j)$ corresponding to the j-th task type.

4.26 throughput reference value: The minimum throughput required by the set of emulated users.

NOTE 25 The mathematical symbol is $B_{Ref}\ (j)$ corresponding to the j-th task type.

4.27 time class: A time limit, combined with a relative frequency corresponding to the ratio of the number of tasks (of a specific task type) - with an execution time less than or equal to the corresponding time limit - to the total amount of tasks (of that particular task type), used for comparison with the execution time of a task (of that particular task type).

NOTE 26 The mathematical symbol for the total number of time classes of the j-th task type is *z(j)* ; the time limit is $g_T\ (j,c)$ where *j* is the current number of the task type and *c* is the current number of the time class, running from 1 to *z(j)*; the relative frequency is $r_T\ (j,c)$ corresponding to the time limit $g_T\ (j,c)$.

4.28 timeliness rating value: The quotient (corresponding to the j-th task type) of the timely throughput and the total throughput.

NOTE 27 The mathematical symbol is R_{TI} *(j)* corresponding to the j-th task type.

4.29 timeliness function: A description of the user requirements with respect to the execution times of tasks of a specific task type. It consists of one or more time classes. The relative frequency of the time class with the highest time limit shall be 1.0 (i.e. = 100%).

NOTES

28 The timeliness function corresponding to the j-th task type consists of *z(j)* time classes, each time class consisting of the pairs of numbers g_T *(j,c)* and r_T *(j,c)*.

29 An example of a timeliness function with 2 classes (*z(j) = 2*) is:

The completion of on-line transactions of a specific type have to be done within:

- 1,5 seconds by at least 90% of the transactions;
- 4,0 seconds by 100% of the transactions.

Up to 10% of the response times may have more than 1,5 seconds but not more than 4,0 seconds; response times greater than 4.0 seconds are not accepted. Assuming the timeliness function is defined for a task type with the number *j = 4* this would result in the following tabled timeliness function:

task type	time class	time class limit	relative class frequency
j = 4	*c = 1*	g_T *(4,1) = 1,5 sec*	r_T *(4,1) = 0,9*
j = 4	*c = 2*	g_T *(4,2) = 4,0 sec*	r_T *(4,2) = 1,0*

4.30 timely throughput: Throughput of all of those tasks whose execution times are accepted with respect to the timeliness function.

NOTE 30 The mathematical symbol is *E(j)* corresponding to the j-th task type.

4.31 user: A person (or instance) who uses the functions of a CBSS via a terminal (or an equivalent machine-user-interface) by submitting tasks and receiving the computed results.

4.32 user type: A classification of emulated users which is defined by the combination of:

- the relative frequencies of the use of chain types;
- the preparation times (mean values and their standard deviations).

5 Abbreviations and symbols

5.1 Abbreviations

For the purposes of this International Standard, the following abbreviations apply.

CBSS	computer-based software system
LAN	local area network
RTE	remote terminal emulator
SUT	system under test
WAN	wide area network

5.2 Symbols

For the purposes of this International Standard, the following symbols apply.

ALPHA	Confidence coefficient of mean execution time
B(j)	Throughput corresponding to the j-th task type
B_{Ref} *(j)*	Reference value of throughput corresponding to the j-th task type
c	Current number of a time class in a timeliness function
d(j)	Half width of the confidence interval of mean execution time corresponding to the j-th task type
$DELTA_h$	Required RTE precision indicator of mean preparation time. It is the maximally accepted relative difference of the actual mean preparation time to *h(i,j)*
$DELTA_q$	Required RTE precision indicator of chain frequency. It is the maximally accepted relative difference of the actual chain frequency to *q(i,l)*
$DELTA_s$	Required RTE precision indicator of the standard deviations of the preparation times. It is the maximally accepted relative difference of the actual standard deviations of the preparation time to *s(i,j)*
$DIFF_h$	RTE precision indicator of mean preparation time. It is the relative difference of the actual mean preparation time to *h(i,j)* that has occurred during the observation period
$DIFF_q$	RTE precision indicator of chain frequency. It is the relative difference of the actual chain frequency to *q(i,l)* that has occurred during the observation period
$DIFF_s$	RTE precision indicator of the standard deviations of the preparation times. It is the relative difference of the actual standard deviation of the preparation time to *s(i,j)* that have occurred during the observation period
E(j)	Timely throughput corresponding to the j-th task type
f(l,k)	Task type of the k-th task in a chain of the l-th chain type
g_T *(j,c)*	Time limit of the c-th time class of the timeliness function corresponding to the j-th task type
h(i,j)	Mean preparation time of the j-th task type and corresponding to emulated users of the i-th user type
i	Current number of the user type
j	Current number of the task type
k	Serial number of a task in a sequence of tasks ('the k-th task of a particular type in the observation period' or 'the k-th task in a chain')
K(j)	Total amount of tasks of the j-th task type which were submitted within the rating interval
l	Current number of the chain type
L_{chain} *(l)*	Total number of tasks in the chain of the l-th chain type (i.e. length of the chain type)
m	Total number of different task types
M(j)	Task mode corresponding to the j-th task type
n	Total number of different user types
N_{tot}	Total amount of emulated users to be emulated by the RTE

$N_{user}\ (i)$	Total amount of emulated users of the i-th user type
p	Total number of different timeliness functions
$q(i,l)$	Relative chain frequency; the relative frequency of using the l-th chain type by an emulated user of the i-th user type.
$r_T\ (j,c)$	Maximum accepted relative frequency of tasks of the j-th task type whose execution times are less than or equal to the time limit $g_T\ (j,c)$
$R_{ME}\ (j)$	Mean execution time rating value corresponding to the j-th task type
$R_{TH}\ (j)$	Throughput rating value corresponding to the j-th task type
$R_{TI}\ (j)$	Timeliness rating value corresponding to the j-th task type
$s(i,j)$	Standard deviation of the preparation time of emulated users of the i-th user type and of tasks corresponding to the j-th task type
t_0	Start time of the RTE, begin time of the stabilization phase
t_1	Begin time of the rating interval
t_{1ind}	Begin time of the individual rating interval of a specific emulated user
t_2	End time of the rating interval, begin time of the supplementary run
t_{2ind}	End time of the individual rating interval, begin time of the individual supplementary run of a specific emulated user
t_3	End time of the supplementary run
t_{3ind}	End time of the individual supplementary run of a specific emulated user
t_4	End time of the latest ending individual supplementary run
$t_{ET}\ (j,k)$	Execution time of the k-th task (of the j-th task type) which was submitted within the rating interval
$T_{ME}\ (j)$	Mean execution time corresponding to the j-th task type
T_{MR}	Mean duration of the individual rating intervals of all emulated users, i.e. the mean of all T_{Rind} values
T_R	Duration of the rating interval from t_1 to t_2 .
T_{Rind}	Duration of the individual rating interval of a specific emulated user from t_{1ind} to t_{2ind} .
$T_{Ref}\ (j)$	Reference value of mean execution time corresponding to the j-th task type
u	Total number of different chain types
w	Total number of different activity types
$z(j)$	Number of time classes of the timeliness function corresponding to the j-th task type

Section 2: Principles of measurement and rating

6 The measurement

6.1 Configuration requirements

A CBSS to be measured shall consist of a specified configuration of its hardware, its system software and application software. All the hardware components (including interconnections of the hardware components and the used WANs and LANs) and all software components shall be specified in detail.

All the stored data required for correct work of the SUT (i.e. data files, content of a used data base system, etc.) shall be described in detail.

None of the components shall have any changes or special modifications with respect to getting better results in the measurement. Because of the responsibility of the tester to submit the results of the measurement to proof the tester shall supply additional documents of the configuration (hardware, software, internal data) of their own choice in addition to the documents requested in this International Standard, which are suitable to repeat the measurement by a third party to attain the same results.

6.2 User emulation

6.2.1 Random user behaviour

In reality, human users do not work strictly deterministically but have random behaviour (i.e. randomly changing preparation times, randomly changing sequences of tasks, etc.). This influences the SUT and results in internal queuing problems. Therefore, users' behaviour shall not be described by fixed sequences of jobs, tasks, transactions or process starts (which have a fixed preparation time before task initialisation and similar properties) but shall be described by statistically significant properties. This yields the following descriptive values e.g.:

- mean preparation times and its standard deviations;
- relative frequencies of the task types.

6.2.2 Remote terminal emulator

The real users, submitting tasks and receiving computational results, are connected to a CBSS (see figure 1).

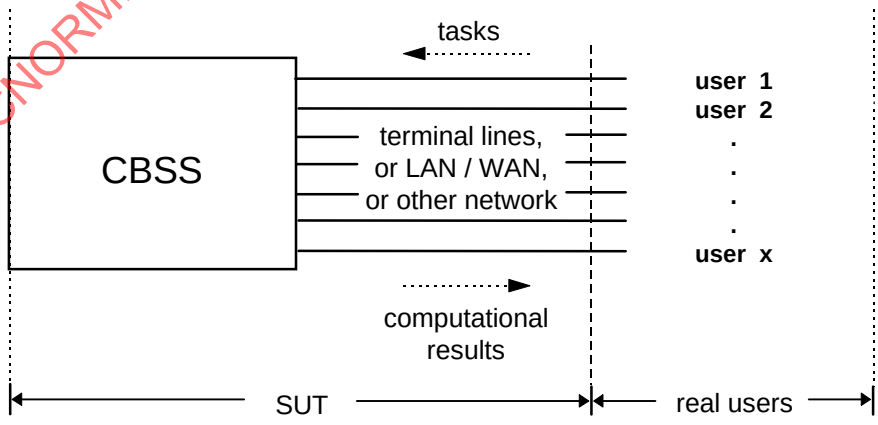

Figure 1 — CBSS in real operation

The group of real users of the SUT shall be emulated by use of a remote terminal emulator (RTE). This RTE shall be connected to the SUT via the real LAN or WAN and the real terminal connection lines. The RTE is driven by the workload parameter set (see 6.2.3). The RTE emulates each user separately and in a real time mode (see figure 2).

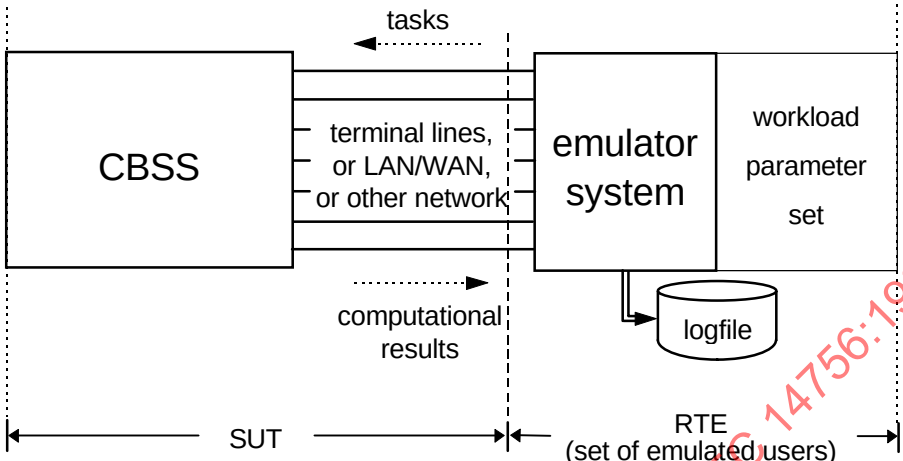

Figure 2 — CBSS in measurement

The chain sequences are random, reflecting the relative frequencies of the task types as defined in the workload parameter set. For special cases this task stream may contain fixed sequences of chains, depending on the necessities of the function of the application software. The preparation times shall be varied randomly . The mean values and standard deviations of the preparation times, as defined in the workload parameter set, shall also be realized. Details of the RTE functions are described in annex A. The RTE shall be implemented by use of a separate data processing system. It shall not be implemented by a program running on the SUT itself.

6.2.3 Workload parameter set

The set of emulated users and their work in relation to the SUT shall be classified and described by a set of parameters. The set of parameters is:

- number of user types and number of emulated users of each type;
- task types, each including:
  - activity type,
  - timeliness function,
  - task mode,

  submitted by the emulated user;
- chain types arranging the defined sequence of the task types;
- relative frequencies of submission of the chain types by each user type;
- mean preparation time values and their standard deviations of all task types for each user type.

The detailed description of this parameter set is given in 10.2 and annex C.

6.2.4 Parameter set for proving the accuracy of the user emulation

The user emulation shall be proved by a set of described parameters which are:

- the required accuracy of the RTE's time behaviour (see 13.2);
- the required statistical significance of the measurement results (see 13.3).

6.3 The measurement procedure

6.3.1 The time phases of the measurement procedure

The measurement procedure consists of three basic time phases:

- stabilization phase,
- rating interval,
- supplementary run,

as shown in figure 3 and the

- observation period

which consists of the rating interval and the supplementary run.

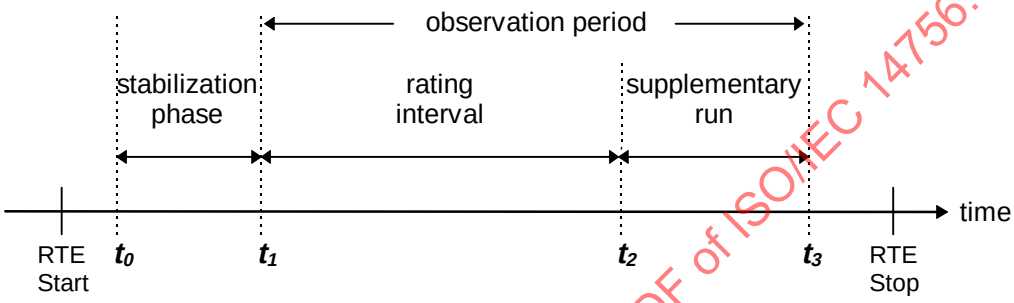

Figure 3 — Phases of the measurement procedure

The measurement begins with a stabilization phase. At the beginning of this phase the login procedures of the emulated users may be performed. The stabilization phase is also needed to bring the SUT in a stable state of operation (e.g. because of the effects of caches). During this stabilization phase the tasks (except the login procedures if any) should be submitted according to the workload parameter set (see 6.2.3).

When the system has reached the stable state of operation the rating interval starts. The task submission shall be performed according to the workload parameter set. Each task submitted during this phase is taken into account for rating. Its duration has to be chosen appropriately to the application which is represented by the software under test.

At the end of the rating interval the RTE shall not be stopped yet. It shall continue submitting tasks as specified by the workload parameter set. It shall only be stopped when the supplementary run is finished, which means that the RTE shall continue submitting tasks until all tasks, which had been submitted within the rating interval, are completed. This is necessary for achieving the same (statistically) influence over the performance of all rated tasks. The influence on the performance shall be the same on those tasks which are running during the rating interval as on those which are still running during the supplementary run but which were submitted during the rating interval.

The phase from the beginning of the rating interval to the end of the supplementary run is the observation period. For this period the measurement procedure is validated by checking all tasks submitted during this period (see 6.4). The measurement shall run under validated conditions during the whole observation period and not only during the rating interval.

6.3.2 Writing a measurement logfile

The required history data of all tasks which are submitted during the observation period shall be recorded in the logfile. The main information is:

- the identification of the emulated user who submitted the task;
- the task type;
- the task submission time stamp;
- the task completion time stamp

Further details are given in 12.1 and annex D.

6.3.3 Writing a computation result file

Results produced by the tasks running on the SUT shall be stored during the observation period in the RTE. The results, delivered by the SUT, may be stored shortened or compressed by the RTE. The computation result file shall include the total computed results or information that allow the proof of the correct computation of the SUT without doubt.

6.4 Proof of validity of the measurement

6.4.1 Proof of the CBSS's computational correctness

The output of all tasks which were initiated and terminated within the observation period shall be proved for having been satisfactorily completed, and having produced correct computational results. If the computational results of the CBSS are not equal to the defined correct results, the proof of the CBSS's computational correctness fails. The measurement is invalid.

Further details are given in 13.1.

6.4.2 Proof of the remote terminal emulator's accuracy

The actual values of mean preparation times, relative chain type frequencies and preparation time standard deviations shall be computed from the logfile. These actual values shall be compared to the values defined in the workload parameter set. The differences between the actual values and the defined values shall be computed and shall not exceed the defined relative limits *DELTA* (e.g. *DELTA = 1%*). Otherwise the measurement is invalid due to imprecision of the RTE, in which case the measurement shall be repeated using a more precisely working RTE.

Further details are given in 13.2

6.4.3 Proof of the measurement result's statistical significance

The statistical significance of the measurement results shall be tested. With the sequential test (so-called from the theory of mathematical statistics) the statistical error of the measured mean execution time shall be proven. It shall not exceed a given level, defined by ALPHA and d(j) (see 10.3.4).

Further details are given in 13.3.

NOTES

31 The method of this standard makes no effort to get measurement runs which are reproducible in minute detail because this does not correspond to the situation in the reality of data processing. But with respect to the macroscopic (i.e. the statistical) properties the method is precise. This is ensured by use of statistical tests and the proof of the CBSS's correct work.

32 Whenever a data processing system is working the details of time behaviour are random. This is due to (among others):

- technical reasons like changing disc spindle resolutions;
- small differences of the clock frequency;
- random command and data collisions on buses, channels and networks;
- random occupation of cache stores.

Therefore the mean values which are calculated from measured response times of task (or jobs) differ randomly to a certain extent if the measurement is repeated. This means that the mean values delivered by a measurement deliver some uncertainties in themselves. Every mean value differs by a random amount from the unknown real value.

7 Calculation of the performance values of the SUT

When calculating the performance values listed in this clause, the values shall be computed from the data recorded in the logfile (see 6.3.2). To calculate these performance values, only the tasks submitted within the rating interval shall be taken into account (see 6.3.1).

7.1 Mean execution time

For each task type the mean execution time ("average") shall be computed with respect to the rating interval.

7.2 Throughput

For each task type the average number of submitted tasks per time unit shall be computed with respect to the rating interval.

7.3 Timely throughput

The timely throughput is the average number of all jobs per time unit which have been completed within time with respect to the defined timeliness function. It shall be computed for each task type with respect to the defined timeliness function of this task type and to the rating interval.

8 Basic data for rating

8.1 User requirements

The user requirements are defined by specifying timeliness functions for all task types. The timeliness function describes the requirements of time behaviour for the execution time of a specific task type. They are used for rating the mean execution time, the throughput, as well as the timeliness.

NOTE 33 No application of a computer is useful in practice if it works too slow in view of user requirements. In order to rate a CBSS with respect to its time behaviour it is necessary to describe the user requirements. This is done by using the timeliness functions.

8.2 The reference environment for rating software efficiency

If in addition to the performance of a CBSS it is also planned to assess the efficiency of some parts of the software then a reference environment shall be defined.

8.2.1 Reference environment for assessing application software efficiency

A reference environment for assessing the efficiency of application software shall be defined. Essential parts of the basic information for rating the efficiency are the (reference) hardware, on which the software is intended to be used, and its (reference) system software, e.g. the operating system, compiler type and module library, network system, etc. All components shall be defined and listed in full. The software configuration and installation parameters shall be clearly specified and documented, including all system software components which may have influence on the SUT's time behaviour. This comprises the reference environment.

The calculated results of the application software efficiency always refer to the reference environment used.

NOTE 34 Software is generally designed with respect to a computer of defined type and size which is assumed to be available to the user. For example, when constructing a bookkeeping program the designer has to consider if it will be run on a small or large PC, a PC network including a data server, a workstation, a midrange multi-user system, a mainframe, or any other type of data processing architecture. The computer intended for use greatly influences the software design; an application software package is always designed for a particular hardware configuration and its system software.

8.2.2 Reference environment for assessing system software efficiency

A reference environment for assessing the efficiency of the system software shall be defined. Its components shall be defined and listed in full detail, analogously to 8.2.1.

The calculated results of the system software efficiency always refer to the reference environment used.

9 Rating the performance values

9.1 Computing the performance reference values

9.1.1 Mean execution time reference values

The reference values for the execution times result from the user's time requirements, defined to be the mean values of the timeliness functions. Mean execution time reference values shall be computed for all task types.

NOTE 35 Example: If 90% of the tasks are allowed to have an execution time of 1.5 seconds and the remaining 10% are allowed to have an execution time of 4.0 seconds, then the reference value for the mean execution time is:

$$0{,}9 \times 1{,}5\ \text{sec} + 0{,}1 \times 4{,}0\ \text{sec} = 1{,}75\ \text{sec}$$

9.1.2 Throughput reference values

The reference values for the throughput result from the hypothetical situation that the mean execution times of the SUT (for each task type) exactly equal the corresponding mean execution time reference values (see 9.1.1). Throughput reference values shall be computed for all task types.

9.2 Computing the performance rating values

9.2.1 The mean execution time rating values

The mean execution time reference value divided by the measured mean execution time value produces the mean execution time rating value for each task type.

9.2.2 Throughput rating values

The measured throughput divided by the reference throughput produces the throughput rating value for each task type.

9.2.3 The timeliness rating values

The value of timely throughput of a task type divided by the measured (total) throughput of this task type produces the timeliness rating value for each task type.

9.3 Rating the overall performance of the SUT

The rating values for the throughput, the mean execution time and the timeliness shall be calculated separately from a measurement and computed for each task type.

NOTE 36 The rating values for throughput, mean execution time and for timeliness are not or only slightly correlated. Therefore they cannot be computed one from another.

Whenever one of the throughput rating values is less than 1 then the throughput of the related task type is too weak.

Whenever one of the mean execution time rating values is less than 1 then the mean execution time of the related task type is too long .

Whenever one of the timeliness rating values is less than 1 there are more execution times which are too long than is acceptable with respect to the user's requirements which are defined by the timeliness functions.

NOTE 37 It could happen that the mean execution time rating value for the related task type is sufficiently high, the timelines rating value is less than 1. In this case, although the mean execution time is as required, the task type concerned would then carry out excessive "runaway" execution times.

9.4 Assessment of performance

9.4.1 The steps of assessment process

The steps for an assessment are shown in figure 4.

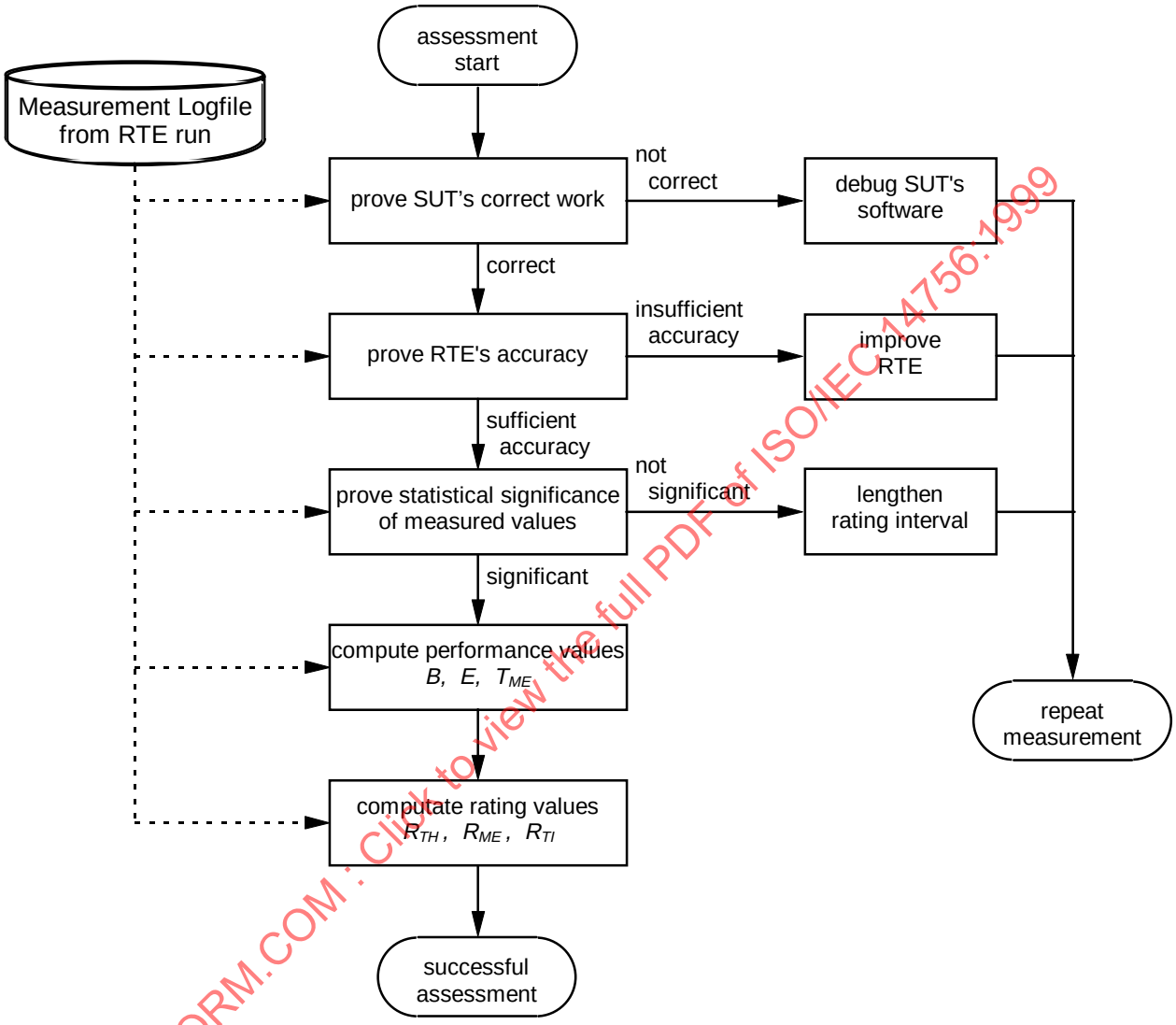

Figure 4 – Assessment of a CBSS with respect to its performance

9.4.2 Weak reference environment

If, one or more rating values of the reference environment (see 8.2) remain less than 1, the internal speed of the hardware of the chosen configuration (see 8.2) is too weak, regardless of whether several efforts had been made to improve the efficiency of the tested software. Therefore, to improve the situation, a faster computer may be used as a reference and a new "reference environment" may be established. In this case the measurement and rating values obtained using the previous reference environment shall be ignored.

Section 3: Detailed procedure for measurement and rating

10 Input requirements

10.1 The SUT description

10.1.1 Specification of the hardware architecture and configuration

The properties of the hardware shall be described comprehensively. The specification of hardware architecture should contain a description of:

- single point of processing system or the client server system, including the type of communication network;
- the main processor unit(s), for instance mono processor system or multi processor system with type of multi processor architecture etc.;
- the front-end processor(s), if used;
- the types of end user communication interface(s), for instance alphanumeric/graphic terminal, voice input/output, etc. .

The specification of the hardware configuration should contain (for instance at conventional system architectures):

- manufacturer, distributor;
- CPU model, serial number, date of shipping;
- number of processors;
- size of main memory (Mbytes);
- type, size of instruction/data cache;
- configuration of I/O-channels;
- secondary storage configuration
  (i.e. number of disks, type, controllers, disk cache type and size, solid state disks - if used, etc.);
- mass storage (optical, magnetic, etc.);
- the interconnections of the hardware components;
- the network architecture, components and their configuration, as far as they are used or as they may have influence to the accuracy (repeatability) of the measurement.

10.1.2 Specification of the system software configuration

The set of the software components shall be described comprehensively. The specification of system software (e.g. a conventional system architecture) should contain manufacturer name, type, and release date of:

- the operating system;
- additional system components as far as used (i.e. job scheduler, software cache for disks, etc.);
- the compiler for each used programming language; additionally, if used, the run time system, shared mode, used optimizer options and data compression features shall be listed;
- the data base system (if used);
- the data configuration, as far as it may influence the accuracy (repeatability) of the measurement, shall be listed;
- the network software as far as used.

10.1.3 The application programs

All programs used by the emulated users shall be presented on a digital storage medium. These programs shall be ready for use on the SUT (either as an executable program or the complete source code).

10.1.4 Additional software required for the measurement run

A list of all additional software components or standard system software modules (for instance system programs, libraries, compilers, data base systems, etc.) which are needed to run, shall contain their exact name, version, release, distributor or manufacturer.

10.1.5 The stored data

All data, which are needed by the programs for their correct working or which have any influence on the performance of the SUT so long as they are not contained in the descriptions of the task type input, shall be presented in their entirety on digital storage medium. They shall be formatted ready for immediate use and storage on the SUT without any further modification.

NOTE 38 Examples of such data can be:

- data files, required for a correct computation;
- output data files used by the programs, which are not empty when starting the test;
- the basic data of a data base system.

10.1.6 Additional information for proof

It is the responsibility of the tester to submit the results of the measurement to proof. Therefore the tester shall supply additional documents of his own choice in addition to the documents requested in this International Standard, which are suitable to repeat the measurement by an external person/group to attain the same results.

10.2 The workload parameter set

The values for the set of parameters described in 6.2.3 shall be defined as follows:

- The total number of user types *n*;
- The total number of emulated users $N_{user}\,(i)$ of each user type $i = 1, \ldots, n$;
- The total number of activity types *w;*
- The total number of timeliness functions *p*;
  this value may differ from the total number of task types, if different task types are defined with the same timeliness function;
- The total number of task types *m*;
- The mean value of preparation time and their standard deviation *h(i,j)* and *s(i,j)* of each user type $i = 1, \ldots, n$ and each task type $j = 1, \ldots, m$;
- The total number of chain types *u*.

Further shall be described the activity types (see 10.2.1), the activity input variations (see 10.2.2), the task types with timeliness function and task mode (see 10.2.3), the chain types and their relative frequencies (see 10.2.4); (for the preparation times' mean values and their standard deviations see 10.2.5).

10.2.1 The activity types

Each activity type (i.e. activities with a similar algorithm of execution) and its input shall be described uniquely.

The definition of the begin and the end point of a specific activity type may be chosen depending on the execution time the tester are interested in.

The activity type may consist of several sequential steps, each with a small preparation time and each with a small amount of execution time. In this case the input of all characters and the processing of the whole sequence shall be defined as one activity with internal steps. If defined in this way, the sum of all partial preparation times shall be defined as the preparation time in the sense of this International Standard and the sum of all partial execution times shall be defined to be the execution time in the sense of this International Standard. (Example: Each single keyed character of a field is processed separately with some leading "keying time", the contents of the field is processed after keying in the last character.)

Normally parallel tasks are seen as to be submitted by different, parallel acting emulated users. In some special cases the system performance of a CBSS which uses one or more sets of parallel steps submitted by one emulated user is to be measured. This means, that at a defined point, steps are forked from a single path of execution to parallel paths and later on are synchronised again to the earlier path. Also such CBSSs can be measured by use of an RTE as defined in this international Standard because the synchronisation is done by an executable step of the SUT itself. In this case not the single steps but the whole structure (the "network of steps") shall be defined as one activity type. The reference values and the rating values (of the throughput of the SUT, the preparation time and the execution time values) in the sense of this International Standard shall be defined as the values referring to such activity types. There is no disadvantage that the RTE specifications defined in this International Standard bear no functionality for doing the synchronisation. If interested in measuring and rating the mean execution time or the timeliness of a specific step inside the activity structure, this may be done additionally and analogously to the measurement and rating of specific task types described in this International Standard.

10.2.2 Activity input variation

The input to a particular activity type may be varied for different tasks of the same activity type, e.g. running numbers in case of opening accounts, access to different data elements, etc. If there is any use of activity type input variation, then all rules of input variation shall be defined comprehensively and uniquely.

10.2.3 The task types with timeliness function and task mode

For each task type the following list of information or values shall be defined:

- The activity type including the input information which have to be submitted to the SUT ("input string"). If variations of the input are required, see 10.2.2;
- The timeliness function consisting of:
  - The number of time classes $z(j)$ of each timeliness function for the task types $j = 1, \ldots, m$,
  - The time class limit $g_T(j,c)$ of each task type $j = 1, \ldots, m$ and of each the time class $c = 1, \ldots, z(m)$,
  - The relative frequency $r_T(j,c)$ corresponding to each task type $j = 1, \ldots, m$ and to each the time class $c = 1, \ldots, z(m)$;
- The value of the task mode $M(j)$ of each task type $j = 1, \ldots, m$. If synchronisation to different preceding tasks (steps) is needed, see 10.2.1
- The correct computational results of the tasks (see 13.1) of each task type.

Any variation of the combination activity type, timeliness function and task mode shall result in a different task type. The variation of input of an activity type (see 10.2.2) should not result in different task types.

10.2.4 The chain types and their frequencies

For each chain type the following values shall be defined where *l* is the current number of the chain type:

- The length of the chain $L_{chain}\ (l)$, i.e. the number of tasks which are contained in the sequence;

  NOTE 39 For independent tasks a chain type of length 1, containing only one task type, is to be defined.

- The sequence in which the tasks of the chain shall be processed, represented by the numbers of the task types:

  $f(l,1), f(l,2), \ldots, f(l,k), \ldots, f(l, L_{chain}\ (l))$

  where $f(l,k)$ is the task type of the k-th task in a chain of the l-th chain type.

For each chain type the relative frequency shall be defined. This shall be done separately for each user type. Therefore it is necessary to define the following *n* rows each having *u* values where *n* represents the number of user types and *u* represents the number of chain types:

for user type 1:
$q(1,1)$ to $q(1,u)$

for user type 2:
$q(2,1)$ to $q(2,u)$
.
.
.
for user type *n*:
$q(n,1)$ to $q(n,u)$

where $q(i,l)$ is the relative frequency by which the l-th chain type is used by an emulated user of the i-th user type.

NOTES

40 If a chain of a certain type (the l-th chain type) does not occur in the task stream submitted by an emulated user of a certain type (the i-th user type) then $q(i,l)$ has the value 0.

41 The sum of the relative frequencies $q(i,l)$ of all chain types $l = 1, \ldots, u$, submitted by a particular user type, is always 100%. This is valid for all user types $i = 1, \ldots, n$:

$$\sum_{l=1}^{u} q(i,l) = 1$$

This may be used as a control for the *q*-values.

10.2.5 Preparation times' mean values and their standard deviations

For each user type the following list of preparation time mean values $h(i,j)$ and corresponding standard deviations $s(i,j)$ have to be defined. *i* is the current number of the user type and *j* is the current number of the task type:

For user type 1:
$h(1,1)$ to $h(1,m)$;
$s(1,1)$ to $s(1,m)$.

For user type 2:
$h(2,1)$ to $h(2,m)$;
$s(2,1)$ to $s(2,m)$.
.
.
.
For user type *n*:
$h(n,1)$ to $h(n,m)$;
$s(n,1)$ to $s(n,m)$.

10.3 Input for measurement validation

10.3.1 Correct computation results

For all task types the correct results, including changes on the stored date (see 10.1.5) shall be specified and listed or stored on a digital storage medium in order to be able to prove the SUT's correct working.

10.3.2 Variation of input data and its resulting output

If there is any use of activity type input variation (see 10.2.2), then all modified results and or variations of task output shall be listed, corresponding to the rules of activity type input variation comprehensively and uniquely.

10.3.3 Criteria for precision of working of the RTE

The following values shall be stated as the criteria for a sufficiently precise working of the RTE (see 13.2):

- $DELTA_q$
  (this is the maximum accepted relative difference of the actual relative chain frequencies and the defined values)
- $DELTA_h$
  (this is the maximum accepted relative difference of the preparation time's actual mean values and the defined values *h(i,j)*);
- $DELTA_s$
  (this is the maximum accepted relative difference of the preparation time's actual standard deviations and the defined values *s(i,j)*).

10.3.4 Criteria for statistical validity of results

The following values shall be stated as the criteria for statistical validity of the measurement results (see 13.3):

- the confidence coefficient *ALPHA* of mean execution time;
- the *m* confidence intervals *2d(j)* of mean execution times of all task types. It is recommended to choose the same relative value for the confidence interval for all task types.

11 The measurement

Fundamentals of the measurement are specified in clause 6

11.1 The measurement procedure

The measurement procedure is carried out in the following 12 steps.

Step 1: Install the SUT (including terminal network and other communication features). Its hardware, system software and application software shall be checked to ensure that they correspond to the descriptions specified in 10.1 (10.1.2 to 10.1.4).

Step 2: Install a RTE which is constructed according to the RTE specifications of this International Standard (see also annex A) and which is proven to fulfil these specifications.

Step 3: Connect the SUT to the RTE using the data communication features used in step 1.

Step 4: Load the workload parameter set (see 10.2), the set of all input strings, and if necessary, also the activity type input variation rules (see 10.2.1), into the RTE.

Step 5: Load the stored data (see 10.1.5) into the SUT.

Step 6: Estimate the duration T_R of the rating interval.

Step 7: Start running the RTE (this is time t_0) and wait for the end of the "stabilization phase".

Step 8: Start recording the logfile and the computation result file. For each task:

- the log record shall be written to the measurement logfile (see 12.1 and annex D);
- the tasks' complete output shall be written to the computation result file (see 12.2, for use when controlling the correct computation of the CBSS, see 13.1).

Step 9: Start the rating interval, if the SUT has reached a stable state of operation (there is no requirement for the duration of the 'stabilization phase' and different systems may have different duration). This is time t_1.

If using 'individual rating intervals', see 11.2.

Step 10: Record the end of the rating interval at the time $t_2 = t_1 + T_R$.

Although the rating interval is finished, the RTE shall continue in full operation for the supplementary run (submitting tasks in the defined distribution for all emulated users according to the workload parameter set) until step 12.

If using 'individual rating intervals', see 11.2.

Step 11: The end of the supplementary run is reached, when all those tasks which were submitted before or at t_2 are completed. The time is t_3.

If using "individual rating intervals", see 11.2.

NOTE 42 Those tasks, which are submitted within the rating interval (i.e.: submission after t_1 and before t_2), are taken into account for computing performance values regardless of whether they are completed within the rating interval or in the supplementary run. Not included in the rating process are those tasks, the submission of which fall into the supplementary run (i.e.: submission after t_2 and before t_3). Nevertheless all the tasks which were submitted within the rating interval or within the supplementary run (i.e.: within the observation period; submission after t_1 and before t_3) are required for validation of the measurement (see clause 13).

Step 12: The recording of the logfile may be terminated, the logfile shall be saved. If no immediate proof of the CBSS's computational correctness has been done, it shall be ensured, that the contents of the computational result file have been saved.

The RTE may be halted.

11.2 Individual rating interval

The use of so called individual rating intervals is a slight modification of the procedure described in 11.1. It is useful for shortening the duration of the rating interval.

NOTE 43 The background is, that - after finishing the stabilization phase - the emulated users typically do not start a new chain at a common moment t_1. Also - coming to the end of the rating interval - the users typically do not finish their chains at a common moment t_2. In consequence the rating interval contains incomplete chains. This is disadvantageously for achieving the stated values of the indicators $DIFF_h$, $DIFF_s$ and $DIFF_q$. Typically long rating intervals are needed.

Shortening the rating interval is possible as follows: If at the planed time t_1 (the end of the stabilization phase) the actual task of an particular emulated user is not completed, then a later 'individual rating interval begin' t_{1ind} for this particular emulated user may be used in order to complete the current chain and to start a new chain. If at time t_2 (the planed end of the rating interval) or t_3 (the end of the supplementary run) the actual task of an emulated user is not the last task of the current chain of this emulated user, then (for this particular emulated user) a later 'individual rating interval end' t_{2ind}, respectively later 'individual supplementary run' t_{3ind} may be used in order to complete the current chain.

The measurement procedure varies as follows:

Step 1 to 8: Similar to ‘step 1’ to ‘step 8’ in 11.1.

Step 9: Start an individual rating interval for each emulated user, if the SUT has reached a stable state of operation (there is no requirement for the duration of the ‘stabilization phase’ and different systems may have different durations). For each emulated user this is the individual time t_{1ind}.

Step 10: Record for each emulated user the end of the rating interval at the individual time $t_{2ind} = t_{1ind} + T_{Rind}$.

Although the individual rating interval is finished, the RTE shall continue in full operation for the supplementary run (submitting tasks in the defined distribution for all emulated users according to the workload parameter set) until step 12 (see below).

Step 11: The end of the individual supplementary run of each emulated user is reached, when all those tasks which were submitted by the particular emulated user before or at t_{2ind} are completed. The individual time is t_{3ind}.

NOTE 44 Those tasks, which are submitted within the rating interval (i.e.: submission after t_{1ind} and before t_{2ind}), are taken into account for computing performance values regardless of whether they are completed within the rating interval or in the supplementary run. Not included in the rating process are those tasks, the submission of which is in the supplementary run (i.e.: submission after t_{2ind} and before t_{3ind}). Nevertheless all the tasks which were submitted within the rating interval or within the supplementary run (i.e.: within the observation period; submission after t_{1ind} and before t_{3ind}) are required for validation of the measurement (see clause 13).

Step 12: When all emulated users have reached their individual time t_{3ind}, the common time is t_4. The recording of the logfile may be terminated, the logfile shall be saved.

If no immediate proof of the CBSS’s computational correctness has been done, it shall be ensured, that the contents of the computational result file have been saved.

The RTE may be halted.

If using individual rating intervals the following requirements shall be fulfilled:

- The latest individual *rating* interval begin time t_{1ind} of any emulated user shall not differ by more than 10% of the mean rating interval T_{MR} from the earliest time t_{1ind} of any emulated user.
- The maximum duration of an individual rating interval shall not exceed 120% of the mean rating interval T_{MR}.
- The maximum duration of an individual supplementary run shall not exceed 150% of the shortest duration of any individual supplementary run.

The calculation of the performance values varies in case of using individual rating intervals:

- The mean execution time $T_{ME}(j)$ (see 14.1) shall be computed by dividing the sum of all execution times of that specific task type by the total number of occurrences of this particular task type, which were submitted in the set of individual rating intervals of all emulated users.
- The throughput (see 14.2) and the timely throughput (see 14.3) shall be computed individually for each emulated user with respect to the duration of its individual rating interval T_{Rind}. The throughput values $B(j)$ and the timely throughput values $E(j)$ respectively are the sum of the individual values of the emulated users.

12 Output from measurement procedure

12.1 Measurement logfile

This file contains all the logfile records according to step 8 in 11.1. Each logfile record contains the following information for the corresponding task:

- a serial number of the task within the observation period, arranged in chronological order from t_1, regardless of which emulated user has submitted the task;
- an individual identification of the emulated user which submitted the task (see note 45);
- the user type of the emulated user which submitted the task;
- the type of the task;
- the type of the chain to which the task belongs;
- the serial number of the task within the chain;
- the time stamp when the preparation time started for this task (see note 46). This time is identical to the task submission or task completion time stamp of the preceding task of the same emulated user, depending on the task mode;
- the task submission time stamp (see note 46);
- the task completion (see note 46).

NOTES

45 For instance the user-identification, as used by the operating system, can be used as the individual identification of the emulated user

46 It is recommended that the values for registered times have a typical precision of 1/100 second for on-line transactions and timesharing commands. If there are other classes of tasks (for instance information transport in a network or real-time functions) then proper values (which are typically shorter than 1/100 second) may be stated.

12.2 Computation result file

All computational results of the observation period (rating interval plus supplementary run) shall be stored in the computation result file. This file is used for controlling the correct working of the SUT. This means that:

- the computed output submitted by the SUT to the emulated users has to be recorded (and stored in the computation result file) for all the tasks submitted after t_1 or t_{1ind} respectively;
- changes on the "stored data" (see 10.1.5), have to be recorded (and stored in the computation result file) at t_3 or after the latest t_{3ind} respectively.

NOTE 47 If the SUT is a large CBSS the computed output, which is submitted by the SUT to the set of emulated users, can be a large amount of data and a problem may arise in storing this data. The alternate solution may be chosen, that the RTE verifies each task response immediately to ensure it is correct. Only those outputs which are not correct may be stored for feeding an error analysis. But, if there doesn't exist a correct output, the measurement is still not valid, i.e. the proof of the computational correctness has failed.

13 Validation of measurements

Only when the measurements are successfully completed (i.e. after executing the steps 1 to 12 in 11.1 or 11.2) the validation shall be carried out. It shall not be attempted before completion of the measurement.

13.1 Validation of the computational correctness of the SUT

The computation result file (see 12.2) shall be analysed. For all recorded tasks the results computed by the SUT shall be compared to the defined correct computation results (see 10.3.1). This may be done by a validation program. If all task's results were complete and proven to be correct then the measurement is acceptable with respect to the correctness of the SUT. This includes that a measurement shall always start with the same data configuration, except if the influence of the data configuration shall be tested. For instance the database content shall be the same at each start.

NOTE 48 See also note 47 in 12.2.

If the SUT apparently failed the evaluation shall neither be published nor rated. After correction of the failure a new evaluation shall only be carried out with a new successful validation based on the new measurement.

13.2 Validation of the accuracy of the RTE

13.2.1 Validity test by checking the relative chain frequencies

The logfile shall be analysed. The relative chain frequencies of all those chains started during the observation period shall be computed. This shall be done separately for all user types and all chain types used. Then the differences for all these relative chain frequencies, as they are computed from the logfile, to the values *q(i,l)*, as they are defined in the workload parameter set, shall be calculated. For the calculation use the formula of $DIFF_q$, which is given in the normative annex B.4.1.

If it holds that:

$$DIFF_q(i,l) \le DELTA_q(i,l)$$

(separately for all emulated user types i and for all chain types l),

then the accuracy of the RTE with respect to the chain generation is within the allowed tolerances.

If the evaluation provides results of minor RTE accuracy the evaluation shall neither be published nor rated. After improving the RTE an new evaluation shall only be carried out with a new successful validation based on the new measurement.

13.2.2 Validity test by checking the preparation times

Analogous to 13.2.1 the entries in the logfile have to be analysed for observation period with regard to the preparation times. The mean values and the standard deviations of the preparation times shall be calculated. This shall be done separately for all user types and for all task types used (for all of those tasks which were submitted and completed during the observation period). Then for all these mean values and standard deviations the relative differences to the values *h(i,j)* and *s(i,j)*, as they are defined in the workload parameter set, shall be calculated. For the calculation use the formulas of $DIFF_h$ and $DIFF_s$, which are given in the normative Annexes B.4.2 and B.4.3.

If there holds:

$$DIFF_h(i,j) \le DELTA_h(i,j) \quad and \quad DIFF_s(i,j) \le DELTA_s(i,j)$$

(separately for all emulated user types i and for all task types j),

then the accuracy of the RTE with respect to the preparation time generation is within the allowed tolerances.

Otherwise the RTE does not work precisely enough and the measurement results are not acceptable. Therefore, a new measurement using an improved RTE shall be carried out.

13.3 Validation of the statistical significance of the measured mean execution time

For each task type a test concerning the statistical significance of the measured mean execution time values shall be performed.

NOTES

49 As shown in the note 32 in 6.4.3 the computed mean values of a measurement are not the real mean values. The theory of mathematical statistics presents a tool to determine the amount of the possible involved error. By use of a mathematical test the so-called confidence interval can be calculated. It is - roughly said - the maximum deviation of the mean value (which was computed from the measured values) to the unknown real mean value. The confidence interval always refers to a so-called confidence coefficient *ALPHA*. *ALPHA*, if multiplied by 100, is the percentage of cases in which the error of the determined mean value possibly is greater than the confidence interval.

50 Example of confidence interval and confidence coefficient:

From the measurement for the j-th task type $T_{ME}\,(j) = 10{,}0$ *sec* is determined and the confidence coefficient is specified to be *ALPHA = 0,05 .* The result of the test may produce the confidence interval *d(j) = 3,5 sec.*

The meaning is that, with a probability of 5%, the unknown real value is not within the range of 10,0 ±3,5 sec. Or, in other words, the unknown real value is with a probability of 95% within the cited range.

Confidence intervals for the execution time shall be used as follows:

The confidence coefficient *ALPHA* is specified in 10.3.4. Additionally, the value *d(j)* for each task type is specified. It is the maximum tolerated error of the mean execution time $T_{ME}\,(j)$. After or parallel to the measurement the sequential test as specified in annex B.6 shall be performed for each task type. The test examines - for the j-th task type - the values $t_{ET}\,(j,1)$, $t_{ET}\,(j,2)$, ... , $t_{ET}\,(j,K(j))$ which are also used for computing the mean execution time (see 14.1). This test shows - by a result of *"OK"* or *"NOT OK"* - for which of the task types the confidence interval (with respect to the specified confidence coefficient *ALPHA*) is not greater than the upper limit *d(j).*

The demanded performance values are only valid when the test signs are on "OK" for all task types. In any other case, the results are not sufficient significant.

NOTE 51 A typical reason for getting a "NOT OK" is that there are too less samples with respect to the chosen *d(j)* values. A new measurement run may be done using a longer measurement interval resulting in more samples and a greater chance of getting confidence intervals which are not greater than the specified limits *d(j)*.

14 Calculation of the performance values of the SUT

14.1 Mean execution time

The mean execution time $T_{ME}(j)$ is the average value of the execution time of all tasks of the j-th task type which were submitted to the SUT within the rating interval with the duration T_R:

$$T_{ME}(j) = \frac{t_{ET}(j,1) + t_{ET}(j,2) + \ \dots \ + t_{ET}(j,k) + \ \dots \ + t_{ET}(j,K(j))}{K(j)} ,$$

where $K(j)$ is the total amount of tasks of the j-th task type submitted to the SUT within the rating interval,

and $t_{ET}(j,k)$ is the duration of the execution time of the k-th task of the j-th task type submitted to the SUT within the rating interval.

$T_{ME}(j)$ shall be computed for all task types, i.e. for $j = 1, 2, \dots, m$.

If during the measurement procedure 'individual rating intervals' are used, see 11.2.

NOTE 52 In case of "NO WAIT" task mode (M = 0; asynchronous mode) the execution times of sequential submitted tasks may overlap. The execution times are added with their full duration in every case (synchronise or asynchronous) in the same way for getting the correct measurement results and rating.

14.2 Throughput

The throughput $B(j)$ is the average number of all tasks of the j-th task type submitted to the SUT within the rating interval per time unit:

$$B(j) = \frac{\textit{Number of tasks of the j - th type, submitted to the SUT within the rating interval}}{T_R}$$

$B(j)$ shall be computed for all task types, i.e. for $j = 1, 2, \dots, m$.

If during the measurement procedure 'individual rating intervals' are used, see 11.2.

14.3 Timely throughput

The timely throughput $E(j)$ is the rate of tasks of the j-th task type, submitted to the SUT within the rating interval and having measured execution times which are completed in time with respect to the corresponding timeliness function:

$$E(j) = \frac{\textit{Number of timely tasks ot the j - th task type submitted to the SUT within the rating interval}}{T_R}$$

$E(j)$ shall be computed for all task types, i.e. for $j = 1, 2, \dots, m$.

If during the measurement procedure 'individual rating intervals' are used, see 11.2.

NOTE 53 The procedure showing how to compute $E(j)$ is somewhat sophisticated. It is described in annex B.3.

15 Rating the measured performance values of the SUT

15.1 Specification of rating level

Rating shall be done by matching the results of the measurement with user requirements and a given rating level (see ISO/IEC 14598-1). In this International Standard the user requirements are represented by the timeliness functions, from which the performance reference values are computed (see 8.1 and clause 9). A rating level shall be specified in conjunction with the user requirements (e. g. by the timeliness functions).

15.2 Computing performance reference values

15.2.1 Mean execution time reference values

For each task type $j = 1, 2, \ldots, m$ there is a reference value $T_{Ref}(j)$. It is (according to the timeliness function of the j-th task type) the maximum accepted mean execution time for tasks of the j-th task type. $T_{Ref}(j)$ shall be computed from values which are defined in the workload parameter set. All m mean execution time reference values shall be computed. The general formula is given in annex B.1.

NOTE 54 An example is given in 9.1.1.

15.2.2 Throughput reference values

For each task type $j = 1, 2, \ldots, m$ there is a reference value $B_{Ref}(j)$. It is the throughput of the j-th task type, which is required, as the minimum value, by the set of emulated users. $B_{Ref}(j)$ shall be computed from values which are defined in the workload parameter set. All m throughput reference values shall be computed. The general formula is given in annex B.2.

NOTE 55 The formula showing how to compute $B_{Ref}(j)$ is somewhat sophisticated. It is described and illustrated in annex B.2.

15.3 Computing rating values

15.3.1 Computing mean execution time rating values

For all task types $j = 1, 2, \ldots, m$ the mean execution time rating values:

$$R_{ME}(1),\ R_{ME}(2),\ \ldots\ R_{ME}(j),\ \ldots\ ,\ R_{ME}(m)\ ,$$

shall be computed. $R_{ME}(j)$ is (for the j-th task type) the quotient of the mean execution time reference value $T_{Ref}(j)$ and the measured mean execution time $T_{ME}(j)$:

$$R_{ME}(j) = \frac{T_{Ref}(j)}{T_{ME}(j)}$$

15.3.2 Computing throughput rating values

For each task type $j = 1, 2, \ldots, m$ the throughput rating values:

$$R_{TH}(1),\ R_{TH}(2),\ \ldots,R_{TH}(j),\ \ldots,R_{TH}(m)\ ,$$

shall be computed. $R_{TH}(j)$ is (for the j-th task type) the quotient of the actual throughput $B(j)$ and the throughput reference value $B_{Ref}(j)$:

$$R_{TH}(j) = \frac{B(j)}{B_{Ref}(j)}$$

15.3.3 Computing timeliness rating values

For all task types $j = 1, 2, \ldots, m$ the timeliness rating values:

$$R_{TI}(1),\ R_{TI}(2),\ \ldots,\ R_{TI}(j),\ \ldots,\ R_{TI}(m)\ ,$$

shall be computed. $R_{TI}(j)$ is (for the j-th task type) the quotient of the timely throughput $E(j)$ and the measured (total) throughput $B(j)$:

$$R_{TI}(j) = \frac{E(j)}{B(j)}$$

15.4 Rating

As shown in 15.3 the rating values $R_{ME}(j)$ and $R_{TH}(j)$ are normalized to the reference values and $R_{TI}(j)$ is normalized to the (actual) throughput B(j). This normalisation causes that a value of 1 for a rating term is the ideal case of a measurement. A value less than 1 means that the user requirement is not fulfilled. A value greater than 1 means the measured value is better than the user requirement.

The rating process focuses primarily on the aspect if the SUT fulfils a stated target. But it cannot be concluded that a SUT exceeding the target is rated in general as "good" as one just fitting the target. It might be more expensive (but it must not do so). It is in the responsibility of the user of this International Standard how to rate this situation. For further help see note 60.

15.4.1 Mean execution time rating

If the mean execution time rating value of the j-th task type $R_{ME}(j)$ is equal to or greater than 1 then the mean execution time of the SUT corresponding to this task type stated by the user requirements is fulfilled.

NOTE 56 The interpretation of a value of $R_{ME}(j) = 1{,}30$ is that the mean execution of the corresponding task type is 30% faster than specified by the user requirements.

If the mean execution time rating value of the j-th task type $R_{ME}(j)$ is less than 1 then the mean response time of the SUT corresponding to this task type is unsatisfactory.

This rating shall be carried out for all m task types.

15.4.2 Throughput rating

If the throughput rating value of the j-th task type $R_{TH}(j)$ is equal to or greater than 1 then the throughput of the SUT corresponding to this task type stated by the user requirements is fulfilled.

NOTE 57 The interpretation of a value of $R_{TH}(j) = 1{,}10$ is that the throughput of the corresponding task type is 10% higher than specified by the user requirement.

If the throughput rating value of the j-th task type $R_{TH}(j)$ is less than 1 then the throughput of the SUT corresponding to this task type is unsatisfactory.

This rating shall be carried out for all m task types.

15.4.3 Timeliness rating

If the timeliness rating value of the j-th task type $R_{TI}(j)$ equals 1 then the timeliness of the SUT corresponding to this task type stated by the user requirements is fulfilled.

If the timeliness rating value of the j-th task type $R_{TI}(j)$ is less than 1 then the timeliness of the SUT corresponding to this task type is unsatisfactory.

NOTE 58 The interpretation of a value of $R_{TI}(j) = 0.90$ is that 10 % of the tasks of the corresponding task type have not finished their execution in the time limits specified by the timeliness function, i.e. the user requirement.

This rating shall be carried out for all m task types.

NOTE 59 The timeliness rating values $R_{TI}(j)$ can not exceed the value 1. The reason is that in the best case all tasks will be completed in time, and dividing this number by all submitted tasks (the same number of tasks) equals to 1. But never could more tasks be completed in time than be submitted.

15.4.4 Overall rating

If at least one of the $3 \times m$ rating values (i.e. m values $R_{TH}(j)$ and m values $R_{ME}(j)$ and m values $R_{TI}(j)$) is less than 1 then the SUT fails to fulfil at least one criteria of the user requirements, specified within the workload parameter set. Therefore the SUT shall be rated with respect to a rating level (see 15.1) as being unsatisfactory or rated to a rating level equivalent to unsatisfactory. Otherwise, if it works sufficiently according to all the time behaviour requirements and no rating values is less than 1, the SUT shall be rated as being satisfactory or rated to a rating level rating level equivalent to satisfactory.

NOTE 60 Today's data processing systems are typically not able to realise exact conformity to the performance reference values of a defined workload description. This is due to insufficiencies of the schedulers. Therefore it is - in sight of the SUT's endusers - important to test the fulfilling of its expectations, i. e. to test the fulfilling of the performance reference values. But the user not always can rate an over-fulfilling. On the other hand over-fulfilling may (but must not) result in a higher price of the SUT. If so, the over-fulfilling would be a disadvantage from the SUT buyer's viewpoint. It is felt that - if including costs - one can not conclude in general that the specific SUT is "good"' or not. If including costs one can conclude only, that the specific SUT is conforming to the assigned profile or not. If not conforming the deviations from the profile are visible by the differences of the R-values to 1. The user of this International Standard is free to define personal bandwidths for the R-values (for instance by setting x-values as follows) and to accept only SUTs within such ranges:

$$x_{ME\text{-}lower} \leq R_{ME}(j) \leq x_{ME\text{-}upper}$$
$$x_{TH\text{-}lower} \leq R_{TH}(j) \leq x_{TH\text{-}upper}$$
$$x_{TI\text{-}lower} \leq R_{TI}(j) \leq 1$$

But he should be aware of the consequence that, if at least one of the values of $x_{ME\text{-}lower}$, $x_{TH\text{-}lower}$, $x_{TI\text{-}lower}$ will be defined to be below 1.0, the meaning is, that the tester may accept a SUT the time behaviour of which is clearly insufficient in sight of the SUT's endusers.

Annex A
(normative)

Specification of the RTE's basic functions

The RTE emulates N_{tot} users. This value is the sum of the number of emulated users of all user types $i = 1 \ldots n$:

$$N_{tot} = N_{user}(1) + N_{user}(2) + \ldots + N_{user}(i) + \ldots + N_{user}(n)$$

All emulated users shall be emulated independently. An emulated user works as follows:

The emulated user submits chains of tasks to the SUT. The sequence of the tasks corresponds to the chosen chain type (see 10.2.4). The emulated user randomly chooses a chain type, both at start of his work and again at the end of each chain, and submits the tasks in the sequence given by the specification of the chain type.

The chain types shall be chosen randomly by the i-th emulated user but according to the defined values of the relative chain frequencies *q(i,1), q(i,2), ... , q(i,u).*

NOTE 61 Assuming the emulated user of user type i has chosen a chain of the chain type l_1. The emulated user submits tasks according to the specification of task type sequence of the l_1-th chain type.

Having completed this sequence the emulated user randomly chooses a new chain type. The number of this chain type is l_2 with the relative chain frequency $q(i,l_2)$. The emulated user submits tasks according to the task type sequence in the definition of the chain type l_2. And so on. Naturally the emulated user never chooses a chain of the l-th chain type if $q(i,l)$ equals zero,

The submission of the tasks of a chain shall be as follows:

(1) The emulated user waits for the (randomly taken) preparation time and submits a task corresponding to the first task type in the chain definition (see 10.2.4). Assuming this first task type is j_1 and the next task type is j_2, the emulated user considers the task mode $M(j_2)$ of the j_2-th task type.

(2) If $M(j_2)$ equals 1 (i.e. emulating "WAIT" mode) then the emulated user waits first for the output of the proceeding task (i.e. the waiting time in this state is equal to the execution time of the task of type j_1); having received completely this output than the emulated user waits for the (randomly taken) preparation time of task type j_2, after which the task of the type j_2 is submitted.

(3) If $M(j_2)$ equals 0 (i.e. emulating "NO WAIT" mode) the emulated user does not wait for the output of the proceeding task. The (randomly taken) preparation time of the task type j_2 starts immediately when submitting the task of the type j_1. After expiration of this preparation time, the task of the type j_2 is submitted.

(4) Continue with steps (1) to (3) until the last task of the chain

(5) This procedure is also valid, when a chain is completed and a new one is chosen. As described above the emulated user acts according to the task type and task mode of the first task of the new chain. The task mode of this task defines whether the emulated user waits for the completion of the last task of the preceding chain before starting the preparation time of the first task of the actual chain or starting the preparation time immediate after the submission of the last task of the preceding chain.

The duration of preparation times of the emulated user of the user type i shall be taken randomly but according to the preparation time mean value $h(i,j)$ and standard deviation value $s(i,j)$ which follows the considered preparation time (see definition of $h(i,j)$ and $s(i,j)$ in 10.2.5 where j is the type of the submitted task.

Annex B
(normative)

Additional calculation formulas

B.1 Formula of mean execution time reference value $T_{Ref}(j)$

$$T_{Ref}(j) = \sum_{c=1}^{z(j)} g_T(j,c) \cdot \left[r_T(j,c) - r_T(j,c-1) \right] \quad , \quad \text{where by definition} \quad r(j,0) = 0 \; .$$

The mathematical terms are defined in 5.2.

NOTE 62 Illustration of the $T_{Ref}(j)$ formula:

The mean execution time reference value results from the hypothetical situation that the SUT executes the tasks just as fast as needed to fulfil the timeliness function. No task is executed faster and no task is executed slower.

This means that the relative frequencies of the execution time values correspond to the requirements of the timeliness function. The value $r_T(j,c)$ is the maximum accepted relative frequency of tasks of the j-th task type the execution times of which are less or equal to the time class boundary $g_T(j,c)$. The value $r_T(j,c-1)$ is the maximum accepted relative frequency of tasks of the j-th task type the execution times of which are less or equal to the time class boundary $g_T(j,c-1)$. Therefore the relative frequency of tasks of the j-th task type the execution time of which is in the time class

$$g_T(j,c-1) \;\; to \;\; g_T(j,c)$$

is

$$r_T(j,c) - r_T(j,c-1) \quad .$$

The situation that the SUT executes the tasks just as fast as needed implies that it responds only with execution time values which equal the upper time class boundaries. I. e. the execution time values of tasks of the j-th task type are

$$g_T(j,1) \;\; or \;\; g_T(j,2) \;\; or \;\; \ldots \;\; or \;\; g_T(j,z(j)) \quad .$$

The last value is the upper time class boundary of the highest time class of the j-th task type.

Now we have to determine the execution times and their relative frequencies:

- The relative frequency corresponding to the execution time $g_T(j,1)$ is $r_T(j,1)$.
- The relative frequency corresponding to the execution time $g_T(j,2)$ is $r_T(j,2) - r_T(j,1)$.
- The relative frequency corresponding to the execution time $g_T(j,3)$ is $r_T(j,3) - r_T(j,2)$
- and so on.

In general the mean value is the sum of the terms “execution time multiplied with its relative frequency“. Herewith and defining $r_T(j,0) = 0$ we find the right hand side of the $T_{Ref}(j)$ formula.

B.2 Formula of throughput reference value $B_{Ref}(j)$

$$B_{Ref}(j) = \sum_{i=1}^{n} \left[\frac{N_{user}(i) \cdot \left\{ \sum_{l=1}^{u} \left[q(i,l) \cdot a(j,l) \right] \right\}}{\sum_{l=1}^{u} \left\{ q(i,l) \cdot \left[\left(\sum_{k=1}^{L_{Chain}(l)} h(i, f(l,k)) \right) + \left(\sum_{k=1}^{L_{Chain}(l)-1} \left[T_{Ref}(f(l,k)) \cdot M(f(l,k+1)) \right] \right) + T_{Ref}\left(f\left(l, L_{chain}(l)\right)\right) \cdot M^{*}(i) \right] \right\}} \right]$$

The mathematical terms used in this formula are defined in 5.2.

Additionally is defined:

A) $a(j,l)$ the number of tasks of the j-th task type within a chain of the l-th chain type;

B) $M^{*}(i) = \sum_{l=1}^{u} \left[q(i,l) \cdot M\left(f(l,1)\right) \right]$.

NOTE 63 Illustration of the $B_{Ref}(j)$ Formula:

The reference values for the throughput result from the hypothetical situation that the mean execution times of the SUT (for each task type) exactly equal the corresponding mean execution time reference values $T_{Ref}(j)$.

We can compute the throughput reference value of the j-th task type according to the general definition of throughput: Throughput is number of tasks divided by the time in which these tasks were submitted.

For the time we take the mean duration of the chain. Consequently for the number of tasks we have to take the number of tasks of j-th type in this chain. The duration of a chain is the time from the beginning of the preparation time of the first task of the chain to the beginning of the preparation time of the first task of the next chain.

The number of tasks of the j-th task type in a chain of the l-th chain type may be named with $a(j,l)$. In general there are several chain types occurring with the relative chain frequencies $q(i,l)$. Therefore we have to follow the rule that the mean number of tasks of the j-th type is the sum of the terms "number of tasks of the j-th type in the chain" $a(j,l)$ multiplied with the "relative frequency of the regarded chain type" $q(i,l)$. The average number of submitted tasks of the j-th task type "during a chain" therefore is:

$$\sum_{l=1}^{u} \left[q(i,l) \cdot a(j,l) \right]$$

Because there are in general $N_{user}(i)$ which submit chains simultaneously we have to multiply the sum by $N_{user}(i)$. This term has to be divided by the mean duration of the chains generated by a user of the i-th type. This duration may be named $t_{chain}(i)$. The number of tasks of the j-th task type submitted per time unit from all users of the i-th type is:

$$\frac{N_{user}(i) \cdot \left\{ \sum_{l=1}^{u} \left[q(i,l) \cdot a(j,l) \right] \right\}}{t_{chain}(i)}$$

Because there are in general n groups of users (users of type 1, users of type 2, ... , users of type n) we have to sum the above written term for this n user types and we get:

$$\sum_{i=1}^{n} \left[\frac{N_{user}(i) \cdot \left\{ \sum_{l=1}^{u} \left[q(i,l) \cdot a(j,l) \right] \right\}}{t_{chain}(i)} \right]$$

This is the right hand term of the $B_{Ref}(j)$ formula. The illustration of the $B_{Ref}(j)$ formula now is complete except to demonstrate the computation of $t_{chain}(i)$. This will be done as follows:

The duration of a chain of the l-th type generated by a user of the i-th type may be named $t_{ch}(l,i)$. This user type submits chains with the relative frequencies $q(i,l)$. Therefore the mean duration of a chain submitted by a user of the i-th type is:

$$t_{chain}(i) = \sum_{l=1}^{u} \{\, q(i,l) \cdot t_{ch}(l,i) \,\}$$

This is the denominator in the $B_{Ref}(j)$ formula.

To illustrate the computation of $t_{ch}(l,i)$ the following may be done:

$t_{ch}(l,i)$ consists of two sums and an additional term:

$$t_{chain}(i) = \sum_{l=1}^{u} \{\, q(i,l) \cdot [\, t_{sum1}(l,i) + t_{sum2}(l,i) + t_{last}(l,i) \,] \,\}$$

The first sum $t_{sum1}(l,i)$ (compare the denominator of the $B_{Ref}(j)$ formula) is easy to understand. It is the sum of the mean preparation times included in the chain.

$$t_{sum1}(l,i) = \sum_{k=1}^{L_{chain}(l)} h(i, f((l,k)))$$

The second sum $t_{Sum2}(l,i)$ is also easy to understand. It is the sum of the mean execution times (which are the $T_{Ref}(j)$ values) of certain tasks. These are the tasks the computational result of which is needed by the user for the following task (i. e. the user is waiting for the task completion before he starts his preparation time of the following task).

$$t_{sum2}(l,i) = \sum_{k=1}^{L_{chain}(l)-1} [\, T_{Ref}(f(l,k)) \cdot M(f(l,k+1)) \,]$$

Not taken into account hereby is the last task of the chain. This will be done by the term $t_{last}(l,i)$ as follows:

Opposite to the tasks within the chain the successor task of the last task of a chain is unknown. It is the first task of the next chain. In case of this first task of the successor chain having a task mode $M = 1$ the execution time of the last task of the chain is part of the duration of the chain elsewhere it is not. The type of the next chain is randomly chosen and the relative frequency of taking a chain of the l-th type is $q(i,l)$. Therefore we have to compute the relative frequency of cases in which the chain starts with a task having the task mode $M = 1$. This is the term $M^*(i)$ (see formula "B)" below the $B_{Ref}(j)$ formula). The mean execution time of the last task of the chain is $T_{Ref}(f(l,L_{chain}(l)))$. The product of $T_{Ref}(f(l,L_{chain}(l)))$ and $M^*(i)$ is the mean value of the last part of the duration of the chain.

$$t_{last}(l,i) = T_{Ref}(f(l, L_{chain}(l))) \cdot M^*(i)$$

This completes the illustration of the above $B_{Ref}(j)$ formula.

B.3 Algorithm for computing the timely throughput E(j)

Let $b(j) = B(j) \cdot T_R(j)$ the amount of tasks of the j-th type submitted to the SUT within the rating interval. This amount may be understood as the committed work to the SUT. The timeliness function stipulates how many of these committed tasks are to be considered as completed in time, partitioned by time classes.

The number of tasks of the j-th type timely completed in the c-th time class may be named $e_X(j,c)$. For $c > 0$ (the smallest time class is given the value 1) $e_X(j,c)$ is defined by the following recursive formula:

$$e_x(j,c) = min\left\{\left[No\left(TR \mid t_{ET}(j,k') \le g_T(j,c)\right) - \sum_{y=0}^{c-1} e_x(j,y)\right] AND \left[\left(r_T(j,c) - r_T(j,c-1)\right)\cdot b(j)\right]\right\}$$

for $c > 0$, *and*

k' *running in the interval* $0 < k' < K(j)$, *and*

where by definition $e_x(j,0) = 0$.

In this formula:

$$min\{a_1 \ AND \ a_2\} ,$$

is a function which chooses the smallest value from the tuple a_1 , a_2 . The function:

$$No\left(TR \mid condition\right) ,$$

counts the number of tasks of the j-th type fulfilling the specified condition.

The number of timely tasks of the j-th task type, submitted within the rating interval is:

$$e(j) = \sum_{c=1}^{z(j)} e_x(j,c) \quad .$$

The timely throughput of the j-th task type, submitted within the duration T_R of the rating interval is (see 14.3):

$$E(j) = \frac{e(j)}{T_R} \quad .$$

NOTES

64 Explanation of the $e_x(j,c)$ formula:

The maximum number of completed tasks of the c-th time class, is defined by the product of the relative class frequency, $r_T(j,c) - r_T(j,c-1)$, and $b(j)$. This is the right-hand term in the min function.

The left-hand term in the min function takes the number of task requests whose execution time does not exceed the c-th time class limit. However, not all these execution times contribute to the number of completed tasks in this time class. On the contrary, the number of execution times, which have already been accounted for in the time classes *1, 2, 3, ... , (c-1),* have to be subtracted. The left-hand term may be smaller than, equal to, or greater than the right-hand term. The min function takes, if unequal, the smaller term. This is the number of completed tasks of the c-th time class.

65 Illustration of the e*(j)* formula:

e(j) is defined as the number of timely executed tasks of the j-th type in all time classes. It is the sum of the number of timely executed tasks $e_x(j,1)$ in the first execution time class, the number of timely executed tasks $e_x(j,2)$ in the second execution time class and so on. The execution time classes are defined by the timely function of the j-th task type. The c-th time class is the interval $[g_T(j,c-1)$ to $g_T(j,c)]$. By definition is $g_T(j,0) = 0$.

The values of $e_x(j,c)$ can be determined as follows:

The total number of tasks (of the j-th task type) the task starts of which are within the rating interval is $b(j)$. Classify all these tasks with regard to the execution time classes. This means that you have to count the number of tasks the execution times of which are in the 1st, 2nd, ... time class. The counted numbers may be named $b_{Class}(j,1)$, $b_{Class}(j,2)$, $b_{Class}(j,3)$, ... , $b_{Class}(j,z(j))$. $z(j)$ is the number of time classes of the timeliness function. Additionally count the number of tasks the execution time of which is greater than $g_T(j,z(j))$. This is the overflow class. The counted number in the overflow class may be named $b_{Class}(j,z(j+1))$.

1. Determination of the reference values $b_{RefClass}(j,c)$ of numbers of tasks in each class. c is the number of the class and j is the task type. $b_{RefClass}(j,c)$ is the product of the actual number of tasks and the required relative frequency. (For the required relative frequency see also the illustration of the $T_{Ref}(j)$ formula in B.1). I. e. it holds:

   $$b_{RefClass}(j,c) = b(j) \bullet (r_T(j,c) - r_T(j,c-1))$$

   By use of this formula the values $b_{RefClass}(j,1)$, $b_{RefClass}(j,2)$, ... can be computed.

2. Determination of the $e_x(j,c)$ values:

   a) Start with the first class:

   If holds

   $$b_{Class}(j,1) < b_{RefClass}(j,1) ,$$

   then

   $$e_x(j,1) = b_{Class}(j,1) .$$

   But if holds

   $$b_{Class}(j,1) \geq b_{RefClass}(j,1) ,$$

   then

   $e_x(j,1) = b_{RefClass}(j,1)$ and the value of $b_{Class}(j,2)$ has to be corrected. The new value of $b_{Class}(j,2)$ is the sum of the old value plus the value of $b_{Class}(j,1) - b_{RefClass}(j,1)$.

   The reason for the correction is that the SUT had executed more tasks with short execution time than required in the actual time class and they therefore are timely also with regard to the next class. In the next steps always use this new value of $b_{Class}(j,2)$.

   b) Examine the second class.

   If holds

   $$b_{Class}(j,2) < b_{RefClass}(j,2) ,$$

   then

   $$e_x(j,2) = b_{Class}(j,2) .$$

   But if holds

   $$b_{Class}(j,2) \geq b_{RefClass}(j,2) ,$$

   then

   $e_x(j,2) = b_{RefClass}(j,2)$ and the value of $b_{Class}(j,3)$ has to be corrected. The new value of $b_{Class}(j,3)$ is the sum of the old value plus the value of $b_{Class}(j,2) - b_{RefClass}(j,2)$.

   In the next steps always use this new of value of $b_{Class}(j,2)$.

   c) And so on still to the z(j)-th class. It is natural not to perform a correction in this last step (because a next higher class is not defined in the timeliness function).

Now the values $e_x(j,1), \ldots, e_x(j,z(j))$ are determined and the sum of them can be computed. It is the value $e(j)$.

The reader may not be irritated by the fact that the values of $b_{Class}(j,c)$, $b_{RefClass}(j,c)$, $e_x(j,1)$ and $e(j)$ are in general not integer values opposite to the value of $b(j)$ which is always an integer value. They are pure arithmetical values and do not represent real numbers of executed tasks.

66 The following example may explain how the algorithm acts:

In the situation of a lower time class being fulfilled by its frequency but a higher not, it should be observed, that a over-fulfilled time class can be used to balance an unsuccessful higher time class. But never a over-fulfilled time class can be used to balance an unsuccessful lower time class.

The timeliness function of a specified task type *j* may be:

time class	time class limit	relative class frequency
1	*≤ 3 sec*	*0,8*
2	*≤ 6 sec*	*0,9*
3	*≤ 20 sec*	*1,0*

The duration of the rating interval is assumed to be

$T_R = 30$ *min.*

The result of the measurement with

b(j) = 50 tasks

may be:

time class	time class limit	executed tasks
1	*≤ 3 sec*	*41 tasks*
2	*≤ 6 sec*	*43 tasks*
3	*≤ 20 sec*	*50 tasks*

The numbers of tasks from the time classes above arranged:

time class	executed tasks in this time class	executed tasks in next lower time class	difference of executed tasks for this time class	tasks over-fulfilling next lower time class	required tasks according to timeliness function	e_x *(j,c)*	tasks over-fulfilling this time class
1	*41 tasks*	*(0 tasks)*	*41 tasks*	*(0 tasks)*	*40 tasks*	*40 tasks*	*1 tasks*
2	*43 tasks*	*41 tasks*	*2 tasks*	*1 tasks*	*5 tasks*	*3 tasks*	*0 tasks*
3	*50 tasks*	*43 tasks*	*7 tasks*	*0 tasks*	*5 tasks*	*5 tasks*	*2 tasks*

In time class 1 are 41 tasks but only 40 are required. Time class 1 is over-fulfilled. It holds: fulfilling

$e_x(j,1) = 40$

and one task will be added to time class 2. Therefore the corrected number of tasks in time class 2 is 3. This new number has to be compared with the number of required tasks in this time class (which is 5). Time class 2 is not fulfilled and it holds:

$e_x(j,2) = 3$

In time class 3 are 7 tasks with no correction form time class 2. But only 5 tasks are required. Therefore it holds

$e_x(j,3) = 5$

Because a 4th time class doesn't exist the over-fulfilling of time class 3 can not be used for balancing. Therefore 2 tasks are "lost".

The sum of the e_x values is the number of timely tasks of the j-th task type:

$e(j) = 40 + 3 + 5 = 48$