INTERNATIONAL STANDARD

ISO/IEC 15437

First edition
2001-08-15

Information technology — Enhancements to LOTOS (E-LOTOS)

Technologies de l'information — Mises en valeur de LOTOS (E-LOTOS)

Reference number
ISO/IEC 15437:2001(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

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.

Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 15437 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software engineering*.

Annexes A and B of this International Standard are for information only.

Introduction

This International Standard contains the definition of the revised version of the LOTOS standard (ISO8807) known as E-LOTOS. The definition of E-LOTOS was based on the initial LOTOS goals enriched with feedback coming from the application of LOTOS to system design in industrial environments, as well as with the goal of supporting the ODP framework (Open Distributed Processing).

The semantics of the language is formed of a behavioural process algebra part which generalizes various LOTOS operators, and of a functional data definition part which is executable and more user friendly. The number of enhancements is high and all of them are very intertwined. A list is given now which tries to highlight from the user point of view the new features from which an E-LOTOS user should benefit when using the language. The most notable differences are:

- Modularity: E-LOTOS includes the following modularization facilities: module and interface definitions; export, import and visibility control; and generic modules. The modules can contain definitions of types, functions and/or processes.

- Data typing. E-LOTOS includes the following facilities for data definitions: predefined types, union types, recursive types, records, extensible records and record subtyping. The predefined data types and schemes are pervasive, i.e. they do not have to be declared before use in a module. As a consequence, they are directly part of the semantics model. At the semantic level, functions have been defined as a particular kind of processes which are deterministic and do not perform any visible event except termination and exception signaling.

  A two level approach has been devised for backward compatibility with LOTOS. The first level is declarative. The second level provides executable definitions of the data types which should satisfy the declarative definition of level one. With this scheme, backwards compatibility with LOTOS is achieved by having declarative specifications at level one, either in ACT ONE or in an enhanced version of ACT ONE. Functional level two data definitions should satisfy level one specifications.

- Time: E-LOTOS models events as atomic and instantaneous occurrences. The specifier can define the time at which actions or behaviours may occur. For example, time restrictions can be added to action denotations restricting the occurrence to a given set of time instants or wait statements can be used to delay the occurrence of a behaviour in time.

- Sequential composition: a new operator " ; " is introduced which substitutes the operators existing in LOTOS for sequential composition, action prefix and exit/enabling (>>) pair. For example $a;B$ or $B1;B2$. In LOTOS the >> operator always generates an internal action when the enabling is performed. In E-LOTOS, internal actions are generated only when necessary. This produces some minor differences with the semantics in LOTOS of some operators.

- Unification of processes and functions at the semantic level. A function is a process which performs only a termination action upon termination.

- Introduction of write-many variables. Write many variables are included in E-LOTOS, with a safe use assured by static semantic means. For example: assignment (write) must be performed before usage (read); no dangerous use of shared variables is made by parallel behaviours; ....

- Introduction in E-LOTOS of output variables in processes and functions as the means to pass values in sequential composition. This substitutes exit statements with value passing coupled with accept statements. The existence of input and output variables in processes and functions provides a unified approach for value communication among sequentially composed behaviours which is much more readable and concise. It is in line with notations used in ODP, like IDL (Interface Definition Language). To

illustrate the new approach to how sequential composition and value passing are performed let us see an example which includes variable assignment, processes and functions.

```
?x := 2;
phase1 [...] (x, ?result) ; (* a process *)
compute (result, ?res1, ?res2) ; (* a function *)
(lphase2 [...] (x, res1) ||| rphase2 [...] (x, res2))
```

- A more general parallel operator has been introduced in E-LOTOS which has the actual LOTOS parallel operator as a particular case. This operator is n-ary and supports the synchronization of J processes among K composed processes where $J \leq K$. Therefore synchronization patterns of 2 among N can be modeled in E-LOTOS. The new operator is also more readable because it clearly identifies the synchronizing gates for each behaviour composed.
- The Suspend/Resume operator which generalizes disabling. The new operator generalizes the LOTOS disabling operator by allowing a disabled behaviour to be resumed by a specific action of the disabling behaviour.
- Introduction of exceptions and exception handling in the behavioural and data parts with a uniform approach. Exception mechanisms permit new ways of structuring and are demanded by ODP.
- Explicit renaming operator for observable actions or exceptions. The renaming operator allows not only to change the name of the events occurring but to add and remove fields from the structure of events, or to merge and to split gates.
- Typed gates and partial synchronization. Gates must be explicitly typed in E-LOTOS. The use of the record subtyping relation permits partial synchronization of gates as well as provides backward compatibility with standard LOTOS untyped gates.

The introduction of enhancements has been performed trying to minimize the growth in the complexity of the language. Rather than introducing new operators, the enhancements have been introduced by generalizing existing operators. Only new statements and/or operators have been introduced when absolutely necessary. E-LOTOS is, as LOTOS, a language which permits a rich variety of specification styles which may be used to model different aspects of the design process. The styles existing in LOTOS, constraint oriented, resource oriented, state oriented, EFSM oriented, monolithic, ... can be also used in E-LOTOS. In addition the existence of exceptions, partial synchronization, event renaming, and other new constructs open new ways of specifying.

Information technology — Enhancements to LOTOS (E-LOTOS)

1 Scope

This International Standard defines the syntax and semantics of the enhanced LOTOS language (ISO 8807), named E-LOTOS. E-LOTOS is used for the formal description of the behavioural aspects of distributed and concurrent systems in general and in the area of open distributed processing in particular.

2 Conformance

A formal specification written in E-LOTOS conforms to the requirements of this International Standard if and only if it is derivable according to the syntactic rules defined in clause 5 and the semantics is unambiguously derivable from the semantic definition in clause 7.

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 8807, *Information processing systems — Open Systems Interconnection — LOTOS — A formal description technique based on the temporal ordering of observational behaviour*

4 Terms, definitions and notation

This clause describes the concrete syntax for E-LOTOS. Here we use a notation similar to Extended Backus-Naur format which is summarized in the following table:

Symbol	*meaning*
|	aternative definition
$D, K, V, S, RT \ldots$	terminal and non-terminal symbols
text	E-LOTOS keywords
::=	definition
*	repetition of the preceding syntactic unit (zero or more times)
()	grouping syntactic units
[]	optional (may or may not occur)

It is worth noting that the metalinguistic '()' and '[]' must not be confused with the E-LOTOS symbols '[', ']', '(', ')', and '|'. In the syntax, these E-LOTOS symbols appear between single quotes. Other symbols (: , -> , etc.) will appear without quotes to keep the International Standard readable.

Syntax for predefined types are described in clause 10.

5 E-LOTOS grammar

5.1 Lexical Structure

This clause describes the lexical units (tokens) of E-LOTOS.

5.1.1 Character set

Characters are divided into several classes denoted by the nonterminals below:

```
<character> ::= <letter> | <digit> | <special-character> | <blank-character>

<letter> ::= "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i"
           | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r"
           | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"

<digit> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

<normal-character> ::= <letter> | <digit>

<special-character> ::=
          "#" | "%" | "&" | "*" | "+" | "-" | "." | "/" | "<" | "="
        | ">" | "@" | "\" | "^" | "~" | "{" | "}"

<blank-character> ::= SP | HT | VT | FF | LF | NL | CR

<character> ::= <letter> | <digit> | <special-character> | <blank-character>
```

Formally, no distinction is made between different versions of the same character, such as capitalized, bold, italic, etc.

5.1.2 Comments and separators

Let <any-string-of-text> represent any string of characters not containing the substring "*)". Then, comments are defined as:

```
<comment> ::= "(*" <any-string-of-text> "*)"
```

Comments are no part of E-LOTOS description. Comments may be inserted anywhere between two other lexical units or left out, except when they play the role of separators. Basically, a comment may be substituted by a SP.

A separator is defined as:

```
<separator> ::= <blank-character> | <comment>
```

Zero or more separators may occur between any two consecutive tokens, before the first token, or after the last token of the E-LOTOS text.

There shall be at least one token separator between any pair of consecutive tokens if the concatenation of their texts changes their meaning.

5.1.3 Identifiers

All E-LOTOS objects are designated by *identifiers*. An identifier begins with a letter, possibly followed by any number of digits, letters and underscore "_" symbol, and ends with a digit or a letter.

```
<identifier> ::= <letter> ( [ "_" ] <normal-character> )*
```

All characters in an identifier are significant, and there is no limit to their length.

Special identifiers are defined in order to allow an intuitive notation for mathematical operators, such as "+", "/", etc. They are built from special characters, normal characters and digits.

```
<special-identifier> ::=
            <special-character> <special-character>*
          | <digit> ( [ "_" ] <normal-character> )*
```

5.1.4 Reserved words

The reserved words for the language are:

