Minutes of the ISO/IEC Editing Meeting on E-LOTOS Helsinki (Finland), 14-18 July 1997

1. List of participants

Hubert Garavel (INRIA, France)
Pekka Kahkipuro (University of Helsinki, Finland)
Matti Luukkainen (University of Helsinki, Finland)
Luigi Logrippo (University of Ottawa, Canada)
Giovanny Lucero (University of Pernambuco, Brasil)
Juan Quemada (DIT-UPM, Spain)
Fumiaki Sato (Shizuoka University, Japan)

2. Rapporteur's report

The Committee Draft (CD) send out for balloting received the positive vote of 11 countries therefore the document has reached officially the CD status. Three negative votes with attached comments were also issued.

With the new ISO/IEC rules we have two choices at this point:

- Submit a Final CD (FCD) for balloting as an output of this meeting. It the FCD is approved the document would reach DIS status.
- Submit a second CD for balloting as an output of this meeting.

A SC21 Plenary meeting is scheduled in Berlin, on January 29.

The possibility of changing the editorship is raised by the editor.

3. List of input documents

[HEL1-1] Belgian comments

[HEL1-2] Canadian comments

[HEL1-3] French comments

[HEL1-4] Japanese comments

[HEL1-5] Romanian comments

[HEL2] A Proposal for Coroutines in E-LOTOS (source : French and Romanian experts)

[HEL3] Japan's Additional Comment on ISO/IEC CD 15437 (source : Japan experts)

[HEL4] Belgian Comments on Japanese, French and Romanian comments accompanying their negative votes on the CD « Enhancements to LOTOS » (source : Belgian experts)

France expresses strong concerns about the fact that the French comments accompanying the vote were sent to NB in an unreadable form, although AFNOR sent them in both PostScript and PDF formats.

4. Appointment of a Secretary

Hubert Garavel volunteers.

5. Resolution of Comments Received

The Canadian Comments [HEL1-2] are presented. The main concerns are :

- Against write-many variables. They are not in the present document.
- Reverse to let instead of write-once variables. Resolved together with French comments.
- Separation of the E-LOTOS tutorial from the standard (the tutorial should be kept as an informative annex). Accepted.
- Inclusion of substantial explanations in the E-LOTOS definition. Accepted
- It is not clear from the tutorial whether it is possible to have generic stacks of lists ; this should be checked.

The Japanese Comments [HEL1-4] are presented :

- Comment JP1 expresses the need for guidelines explaining how LOTOS descriptions should be upgraded to E-LOTOS. These guidelines should be an Appendix of the E-LOTOS. The guidelines are especially mandatory if E-LOTOS is meant to supersede the LOTOS Standard. Accepted.
- Comment JP2 is agreed by everybody (it is also a Canadian concern). It is agreed that the next version of the CD should contain appropriate explanations for semantic rules.
- Comment JP3 resolved together with AFNOR-13.
- Comment JP4 resolved together with Comment RSI-4-21.
- Comment JP5 An example illustrating JP5 would be helpful. A restriction should exist in order to avoid non-determinism in patterns (e.g. as in page 49 of the standard).
- Comment JP6 is discussed. It appears that the definition of non-terminal D (Section 3.3.1) should be moved at the module level (Section 3.4.3 page 53). Comment closed.
- Comment JP7 resolved together with AFNOR-2.
- Comment JP8 is discussed: infinite divergence blocking time is undesirable, but no appropriate solution has been found, since any solution would seriously limit the expressiveness of E-LOTOS. Comment closed.
- Comment JP9 is agreed. A better notation for type judgments should be found and one should avoid conflicts between the words used in type judgments notations (i.e , type, guarded...) and the E-LOTOS keywords (either by using a different font or by choosing different names).

The French Comments [HEL1-3] are presented :

- Comment AFNOR-1 The proposal for a « ... » notation (same as defined previously in the user-language) based on macro-expansion was approved in principle. France will provide a detailed technical proposal.
- Comment AFNOR-2 is agreed. France will provide a detailed technical proposal. An IDL-like syntax is found desirable for E-LOTOS : this should be taken as a concrete goal for producing the next revision of the CD.
- Comment AFNOR-3 is agreed in principle, subject to receiving a concrete proposal.
- Comment AFNOR-4. It is acknowledged that there is a contradiction between clauses (D3) p. 48 and (D3-D6) p. 58. Alignment should be achieved.

- Comment AFNOR-5 resolved together with RSI-1.
- Comment AFNOR-6 is accepted
- Comment AFNOR-12. It is agreed that the exit operators should be removed from E-LOTOS, although they can be kept as semantical notations for defining the dynamic semantics for E-LOTOS. There is agreement to remove process definitions with exit statements (clause (D3) page 48) and to merge (D5) with (D6) page 48 : a function can have in and/or out argument and an optional result (if there is no result, the « void » keyword should be used). Page 58 should be aligned with these changes. There will be a neutral element for sequential composition, noted « null » (« null » is identical to « exit » without parameters)
- Comment AFNOR-7 is accepted in principle subject to receive a proposal.
- Comment AFNOR-13 is accepted. It was decided not to rely upon priorities for behaviour operators (this was considered as potentially unsafe) excepted for sequential composition (« ; »), which has the highest priority, and to force bracketing wherever there is an ambiguity. Of course, behaviour expressions of the form « B1 [] B2 [] B3 » do not create ambiguities (because « [] » is associative) and should not require bracketing. The choice of bracketing syntax was the discussed ; three choices are considered, parentheses (as in LOTOS), curly braces (as in C), or « dis/enddis, sel/endsel », as in Brinksma's PhD thesis. It was decided to go for the 3rd solution, using « select/endsel, disable/enddis, suspend/endsusp », but to investigate the use of the 2nd solution on concrete examples.
- The Committee agreed that there are two kinds of uses for E-LOTOS: a more mathematical use (which makes use of infinite types) and a more implementation-oriented use. It was agreed that there should be two levels in language : a specification-oriented one and an implementation-oriented one (« implementable E-LOTOS »). In « implementable E-LOTOS », the « choice » operator should only deal with finite types. The standard should clearly identify both levels, such that tools are only obliged to implement the "implementable level".
- Comment AFNOR-15 is accepted. Regarding the 3rd item of AFNOR-15, the « choice X :any » construct should be removed from implementable E-LOTOS or limited to finite types. A related question is left open : should E-LOTOS allow infinite parallel composition, in the case where these processes have functionality « noexit » and where the parallel operator considered is interleaving.
- Comment AFNOR-16. Rationale 1 of AFNOR-16 is accepted. Rationale 2 is recognized as a potential problem and deferred until the base library is defined. Rationale 3 is discussed : it is acknowledged that the need to send different values on a gate is a common situation, which should not be addressed only by « any » or « (etc) ». The need for union types is recognized to be important. Rationale 4 is discussed : the introduction of extensible unions in the type language is considered for inclusion in the output document, at least at the static semantics level using textual substitution. The proposed syntax is :

```
type T is
        C1 (...)
        | C2 (...)
        ...
        | Cn (...)
        | etc
endtype
type T' is T ... endtype
```

The « etc » keyword in unions will have only a static semantics (using textual substitution), not a dynamic semantics. Additionally, the lists of arguments of constructors should be flat lists. The Committee considers type definitions of the form

```
type PACKET is
        C1 (int, bool)
        | C2 (A => int, B => bool)
endtype
```

Whether both unamed and named fields should be allowed simultaneously is discussed with the following result. In the case of C2, it should be possible to invoke constructor C2 with either positional

or named arguments. Both invocations should be equivalent modulo a translation to a canonical form. This can be provided with a translation from positional to named following the order of the definition. Rationale 5 was already agreed and it was decided not to have extensible parameters lists for processes and functions (see Comment AFNOR-4).

- Comment AFNOR-10 The « forever » keyword will be removed. The loop construct is recognized to be a particular case of « write-many » variables, because some variables must be initialized at the beginning of the loop and modified within the loop. It is agreed that the problems detected in AFNOR-10 will have to be solved.
- Comment AFNOR-11. It is agreed that the « functional » form of loop (i.e., loop returning a value as in clauses (B30) and (E11)) should be replaced by a classical form of loop that assigns variables, which will be used after the loop. The « break » clause for exiting from a loop should be available. A "while" loop with a condition should be introduced and the "for" loop of the C language should be introduced as syntax sugar"
- Comment AFNOR-9 is agreed. The « var » and « init » clauses are removed. The « local » construct is kept without « var » and « init ». Then, the syntax of the « local » construct should be changed into

  var X1 :T1 [:= E1], ..., Xn :=T2 [:= E2] in B endvar

  For instance, the Committee agrees that the following example should be accepted :

```
var X :S := 0 in
case Y in
   ... -> X := 1
    otherwise null
endcase
endvar
```

- Comment AFNOR-14. It is acknowledged that the current write-once scheme has problems as shown by these pathological examples. These pathological examples should be prohibited in the next version of the CD. There was agreement that the present variable scheme is unsufficient and that a write-many scheme that preserves process algebra properties should be considered for the next version of the CD.
- Comment AFNOR-8. Expressions in favour of unifying gates and signals, if this does not introduce major side side-effects, were expressed. The question is left open for a next meeting.
- Comment AFNOR-5 is reviewed and should be discussed when the predefined type library will be available.

The Belgian General Comments (GC).

- Belgian GC-1 expresses Belgian support
- Belgian GC-2 is a point of view.
- Belgian GC-3 is resolved : the will be an annex explaining backward compatibility with LOTOS and E-LOTOS.
- Belgian GC-4 is agreed.
- Belgian GC-5. Renaming and dot notation are two solutions for solving name clashes. All participants except France would like to have dot notation. Several participants express concerns about redundancy between these two constructs. Concrete contributions on this topic are expected.

The Belgian Technical Comments (TC) are reviewed.

- Belgium TC1 and TC2 are related to AFNOR-14 and is solved as decided for AFNOR-14.
- Belgian TC3 and TC4. It is noticed that a simpler solution for the example of TC3 is provided at the bottom of page 41 of the CD. However, the Committee would like to have a simpler syntactical form for renaming/instantiation, such as :

```
module ListNat is
GenericList (E := Nat, List := ListNat)
endmod
```

  if this does not create semantical problems.

- Belgian TC5. The problem is acknowledged, but the proposed change (generating an internal action for every non-deterministic constructs) creates backward compatibility with standard LOTOS. For instance, the LOTOS expression

  choice X :S [] A ; B !X ; stop

  could not be described in E-LOTOS if « X := any S » generates an internal action (unless one only considers testing equivalence). Therefore, the proposed changes are left for further study in the next meeting.

The Belgian editorial comments will be taken into account.

The Romanian comments are reviewed.

- RSI-1 is agreed. It is decided to remove anonymous tuples because the only two places where the need was clearly established are (1) functions returning several results : this is already covered by « out » parameters ; (2) « case » statements which check several patterns at the same time. For (2), it will be necessary to introduce « case (x, y, z) » statements, where « (x, y, z) » is a flat list of arguments.
- RSI-2. Subtyping should be kept in its present form and extended with extensible unions. RSI-2 seems to be partially covered with the decision of introducing extensible unions. In order to ensure static type-checking, the use of the « any » type should be restricted to communications only, subject to further checks (in particular, the semantics of « G ?X :any » should be checked to ensure that the corresponding LTS model is countable).
- RSI-3. It is decided to introduce the minimal amount of overloading needed for the base library and easy instantiations of generic types.
- RSI-4. Regarding Comment 28, it is decided to add « else if » clauses in the « if-then-else » statement (it will be decided later whether to introduce curly braces). Comment 47 should be solved by putting declarations at the module level. Comment 36 is discussed : the default « exit » (or « null ») behaviour is kept when there is no « else » clause : this comment will go into the « backward compatibility » appendix. The errors dectected in the RSI document will be taken into account in the edition of the document.

General decisions

- It was decided to have only one « exit » (without arguments) and rename it to « null ».
- Regarding patterns: there should be simplified. Rule (P4) should be removed. In rule (P2), the keyword « any » should be typed (this keyword is misleading). Anonymous records will be removed from patterns too (subject to check of undesirable effects). Patterns should be extended to allow constants (which can be defined as terms made of constructors).
- The base library. The base library should contain enumerated types (as a special case of unions with additional operations =, <, succ, pred, and subtranges of enumerated types), integers, booleans, real numbers, lists, strings, arrays, FIFO queues, sets, bags, etc. There should be infix notations and priorities between operators. It should not be allowed to declare simultaneously an infix operator and prefix operator with the same name.
- It was agreed to have natural numbers inductively defined by constructors 0 and Succ. Also, numeric constants such as 2, 3, 4, ... should be available. These constants will be defined in terms of constructors, so that they can be used in « case » patterns.
- Signed integer numbers should be available as well.
- Subrange types should be allowed for natural, integer and enumerated types (e.g. 10..100). Subrange type should be different from their parent type. Conversion of a subrange type to its parent type should be done explicitly using conversion functions, which can raise exceptions in case of domain error.
- The E-LOTOS language should not be bound to a particular character set. Several character sets should be available (including Iso-Latin-1 and Unicode). The choice of the character set should not be built-in, but it should be definable by the user.
- The base library should have a bit type and an octet type.
- The base library should also support bit strings, byte strings and character strings and provide convenient notations for these strings (i.e., encoding formats for data types).
- The Java notations for constants should be taken as a basis for E-LOTOS notations of constants.

As far as backward compatibility with LOTOS is concerned, there will be an appendix in the output document giving a mapping table for translating LOTOS descriptions in E-LOTOS ones. For the behaviour part, guidelines for translation statement by statement by statement will be provided. For the data part, there should be indications of how to translate abstract data types into interfaces containing types declarations , functions declarations and equations.

6. Presentation of Input Documents

- [HEL2] is discussed. It is an interesting proposal, which provides a clear and general coroutine mechanism. The decision of introducing this mechanism in the output document is deferred to the next email meeting.
- [HEL3] is discussed. Further contributions on this topic are expected. The impact of establishing priorities between inference rules should be carefully studied.
- [HEL4] is discussed. These comments have been covered in the previous discussions (see the related points).

7. Relation of E-LOTOS with the LOTOS Standard

The Committee agrees that the E-LOTOS standard will not supersede the International Standard IS8807 « LOTOS » when it will be approved. The reason is that LOTOS was defined with the goal of serving as a specification language for ISO-OSI standards. E-LOTOS has been defined with the goal of supporting the specification of ODP standards. In order to align with the ODP framework the E-LOTOS standard is not simply an enhancement of LOTOS but a complete FDT aligned with ODP. There will exist guidelines for translation of LOTOS specification into E-LOTOS, but nevertheless some countries have expressed concerns about the E-LOTOS standard superseding the LOTOS Standard. Therefore the validity of the LOTOS standard should be bounded to its need in the ISO-OSI framework only.

8. Planning of work

A liaison statement should be sent to other Working Groups in order to show the use of E-LOTOS for describing the ODP Trader.

The resolution of comments after the Helsinki meeting should be done before the end of August, 31, 1997.

It is decided that the next document will be submitted for balloting as a Final CD, although France express views in favour of balloting the document as a second CD.

The planned target is :

- End of August 1997 : Final CD should be ready, followed by a Final Ballot (which will take 4 months)
- January 1998 (editing meeting option 1): If ballot results are available an editing and comment resolution meeting will take place during the SC21 plenary in Berlin.
- January 1998 : A joint meeting with other WIs will be help at the SC21 plenary in Berlin to analyze the use of E-LOTOS in other Wis. An E-LOTOS tutorial will be given in this meeting.
- February/March 1998 (editing meeting option 2): If ballot results are not available for the SC21 plenary an editing and comment resolution meeting will take place after ballot resolution.
- February-April 1998 : the output of the editing meeting should be ready.

Juan Quemada keeps the editorship responsibility for producing the next output document.

An e-mail meeting for producing the output document will start on July, 21st, 1997and end on August 31st, 1997. This meeting will be conducted according to the ISO rules for e-mail editing meetings.

The output document should have the following structure :

1. Introduction (which includes description of the structure of the document and rationale)
2. E-LOTOS complete BNF grammar (which should merge the base language and module language grammars). The grammar should proceed in a top-down presentation order, starting with the module section (currently Section 3.4, followed with Section 3.3). Value expressions (Section 3.3.9) should be presented before behaviour expressions (Section 3.3.8). A richer explanation of the notation used, including examples, should be given.
3. E-LOTOS modules
4. E-LOTOS base language. Suggestion to add further subdivisions in this section (grouping types, functions and expressions, processes and declarations) if this improves readability.
5. E-LOTOS predefined library

Appendix A : tutorial (including ODP trader)
Appendix B : guidelines for LOTOS to E-LOTOS translation

The next SC21 Plenary Meeting will be held after ballot resolution (see options 1 and 2).

The next plenary meeting is planned in Brisbane (Australia) in summer 1998.