

**Title:** Position Statement regarding E-LOTOS Progression of Work

**Source:** French-Romanian experts

**Date:** December 1996

**Action:** For consideration during the Grenoble meeting

## General comments

1. Since the SC21 meeting in Kansas City, work on E-LOTOS has progressed noticeably.
2. However, several decisions taken by the Committee have not been properly enforced. In particular, it is clear that structure of the *Revised Working Draft on Enhancements to LOTOS (V4)* does not match the decisions taken during the Kansas City meeting regarding the contents of the Kansas City output document. This results in a lack of integration between the user language and the core language, that is likely to delay in the production of the Committee Draft.
3. The definition of E-LOTOS core language included in the definition of the Revised Working Draft (V4) contains many new features, which have been introduced after the Kansas City meeting.  
As these features have not been approved (nor even discussed) by the Committee, it is unclear to which extent this document can be considered as an output of the Kansas City meeting.
4. Also, there are still technical issues which are not addressed by the current Revised Working Draft on Enhancements to LOTOS (V4). Especially, two important open issues are: *modules* and *suspend-resume* operators (possibly in their generalized form: *coroutines*). The Grenoble meeting should focus, in priority, on these open issues.
5. In spite of these difficulties, the production of a Committee Draft after the Grenoble meeting still seems to be a reachable goal.

## Decisions taken in Kansas City

During the Kansas City meeting, the Committee agreed to organize the output document in order to avoid useless duplication of work between the user language and the core language. The decision taken were twofold:

1. A precise structure of the output document was defined:
  - Chapter 1: INTRODUCTION (Spain)
  - Chapter 2: SYNTAX OF USER LEVEL LANGUAGE (France & Romania)
  - Chapter 3: STATIC SEMANTICS (France & Romania)
  - Chapter 4: SYNTAX OF CORE LEVEL LANGUAGE (United Kingdom & Belgium)
  - Chapter 5: TRANSLATION (United Kingdom & Belgium & France & Romania)
  - Chapter 6: CORE DYNAMIC SEMANTICS (United Kingdom & Belgium)
  - Annex A: TUTORIAL WITH RATIONALE FOR THE LANGUAGE (Spain & France & Romania & Belgium & Canada)
  - Annex B: UPWARD COMPATIBILITY WITH LOTOS (France & Romania)
2. The Committee also stated that: *“The Committee agrees not to have two different static semantics. Static Semantics should be made by the translation function. The core will provide only requirements for static semantics and a dynamic semantics.”*

## Report of facts

France & Romania respected carefully these editing instructions:

- On July 2, 1996, they made available a contribution containing chapters 2, 3, and a proposal for chapter 5 (based upon the decisions taken for the core language during the Kansas City meeting). This document is still available from <ftp://ftp.imag.fr/pub/SPECTRE/LOTOS/ELOTOS/kansas-out-0.ps.Z>.
- On July 10, 1996, France & Romania made available a revised contribution, still aligned on the decisions taken in Kansas City. This document is still available from <ftp://ftp.imag.fr/pub/SPECTRE/LOTOS/ELOTOS/kansas-out-0.1.gz>.

But France & Romania had to face a patent lack of co-operation from UK & Belgium:

1. Instead of producing the requested chapters 4, 5, and 6 as separate parts to be integrated in the co-operative work, UK & Belgium decided to produce a single, standalone document in which syntax and semantics of the core language are intricated together.

Faced to this questionable attitude, France & Romania had to modify entirely their contribution, in order to save the uniformity of the output document.

As a consequence, the Kansas City document is the juxtaposition of two different languages rather than a progress towards a unified language.

This unfortunate status of things must be solved before producing a Committee Draft.

2. Instead of concentrating on the core semantics, UK & Belgium also included in their document a complete static semantics for the core language, in a patent violation of the decisions taken in Kansas City.

This static semantics is duplicated with chapter 3 and it is not actually necessary, since static semantic controls are already performed at the user language level and by the translation function.

3. UK & Belgium even increased the complexity of the core language by introducing lot of syntactic sugar borrowed from the user language, such as “**in**” and “**out**” parameters, “**if-then-else**” and “**loop**” statements, etc.

As the base justification given for the core language was the wish for a small semantic calculus suitable for publication in scientific papers, the need for such syntactic sugar is not obvious. Again, this leads to a useless duplication of work with chapters 2, 3 and 5.

4. At the same time, UK & Belgium did not extend the core language to support useful features of the user language (such as write-many variables, which are a standard in any imperative language). Clearly, the amount of work spent by UK & Belgium on syntax sugar and static semantics would have been better spent on semantic issues.

Technically, some design choices made by UK and Belgium during summer 1996, and included in the Kansas City output document, are questionable:

- The new core language allows non-deterministic evaluation of expressions, which implies that the evaluation of an expression can generate a LTS whose transitions are labelled by exceptions names.

For efficiency in implementations (especially for compilers that will translate E-LOTOS to imperative languages), the evaluation of expressions should be deterministic.

We recommend to revert to the previous proposals for semantics of the core language and to ensure a deterministic evaluation of expressions by evaluating the arguments of functions and processes from left to right, as it is the case in SML.

- The need for having expressions in patterns is not clear. Many languages, including SML, do not support this feature. The two examples given page 11 (zero test and palindrome) could be done in a simpler and more classical way using an “**if-then-else**”.
- Multiply-assigned variables are definitely a “must” for the industrial acceptance of E-LOTOS and should be supported either in the dynamic semantics of the core language or by the translation function from the user to the core language.
- The discussion with ODP experts during the Kansas City meeting made clear that the concept of subtyping included in the core language has nothing to do with ODP subtyping (which is related to preorder relations between behaviour expressions).

On the other hand, this notion of subtyping requires that a significant amount of type-checking (including implicit type conversions) be performed at runtime. For instance, the synchronization “**G ?x:int || G ?y:real**” will succeed, as the integer **x** will implicitly be promoted to a real number. The need for run-time type-checking is even greater with the special type “**any**”.

We recommend to avoid implicit type conversion and to handle polymorphism using the generic features provided by the module system, so that type-checking can be done at compile-time.