EC-Canada Exploratory Collaborative Activity EC-CA 001 : 76099

EUCALYPTUS

A European/Canadian LOTOS Protocol Tool Set

Periodic Progress Report — 1993

Contents

Chapter 1

Scientific and technical progress

1.1 Participants

The first year of the EUCALYPTUS project involved the following participants:

For Grenoble: In the VERIMAG laboratory, five researchers have contributed to the EUCALYPTUS project:

- Pr. Joseph Sifakis: project management
- Dr. Hubert Garavel: project management, improvement of the CÆSAR and CÆSAR.ADT tools
- Dr. Laurent Mounier: improvement of the ALDÉBARAN tool
- Alain Kerbrat: improvement of the ALDÉBARAN tool and development of the graphical interface
- Radu Mateescu: improvement of the CÆSAR.ADT tool

For Liège: At University of Liège, six researchers have contributed to the EUCALYPTUS project:

- Pr. André Danthine: project management
- Dr. Guy Leduc: project management, supervision of assessments of tools, convergence of tools
- Charles Pecheur: design of APERO, management of the EUCALYPTUS server, convergence of tools
- France Bierbaum: assessments of CÆSAR, CÆSAR.ADT, ALDÉBARAN and ISLA

- Michel Jankowski: assessments of CÆSAR, CÆSAR.ADTand ALDÉBARAN
- Luc Léonard: assessment of SMILE

For Ottawa: The Ottawa participation in EUCALYPTUS is supported by the Telecommunications Research Institute of Ontario (TRIO). The following researchers have contributed to EUCALYPTUS:

- Pr. Luigi Logrippo: project management
- Jacques Sincennes: improvement of the tools

For Montréal: The Montréal participation in EUCALYPTUS is supported by the IDACOM-NSERC-CWARC Industrial Research Chair on Communications Protocols at the University of Montreal of which Gregor v. Bochmann is the chairholder. The support has been approved by the steering committee of the chair in March 1993. The following researchers have contributed to EUCALYPTUS:

- Pr. Gregor v. Bochmann: project management
- Pr. Rachida Dssouli: collaboration
- Daniel Ouimet: supervision
- Omar Bellal: improvement of the TETRA tool

1.2 Task reports

This section reflects the decomposition in tasks provided by the workplan attached to the Contract (technical annex I, section 2.1). These tasks are summarized below:

Task number	*Task name*	*participants*
0	Management and coordination	Grenoble
1	Tool assessment	Liège
2	Tool improvement	Grenoble, Ottawa, Montréal
3	Tool convergence	Grenoble, Liège, Ottawa, Montréal
4	Tool integration	Grenoble, Ottawa, Montréal

The following sub-sections report the activities carried out in the performance of Tasks 0, 1, 2, and 3.

Task 4 will not be mentioned here since it is planned to start only in 1994.

1.2.1 Activities performed in Task 0 (Management and coordination)

Management and administration are carried out by Grenoble, with a deliberate attempt to avoid excessive administrative overhead, that would slow down the research and development activity.

As a deliberate choice, most of the communication between the EUCALYPTUS partners is done using electronic mail.

Being responsible for the tool assessments, Liège has set up and manages an FTP server in its premises, whose purpose is to have a common repository for storing the last releases of the EUCALYPTUS tools and reports.

Since the beginning of the EUCALYPTUS project, three meetings were held:

- a two-day kick-off meeting, held on March the 3rd and 4th 1993 in Liège
- a one-day meeting, held on September the 10th 1993 in Montréal
- a one-day meeting, held on January the 11th 1994, in Madrid

In the context of Task 0, Grenoble was in charge of writing the minutes of these meetings [EUCA/GR/01] [EUCA/GR/03] and [EUCA/GR/04] and editing the periodic progress reports.

Reports

[EUCA/GR/01] H. Garavel, *Minutes of the 1st* EUCALYPTUS *meeting – Liège, March the 3rd and 4th 1993*. 13 pages. In French.

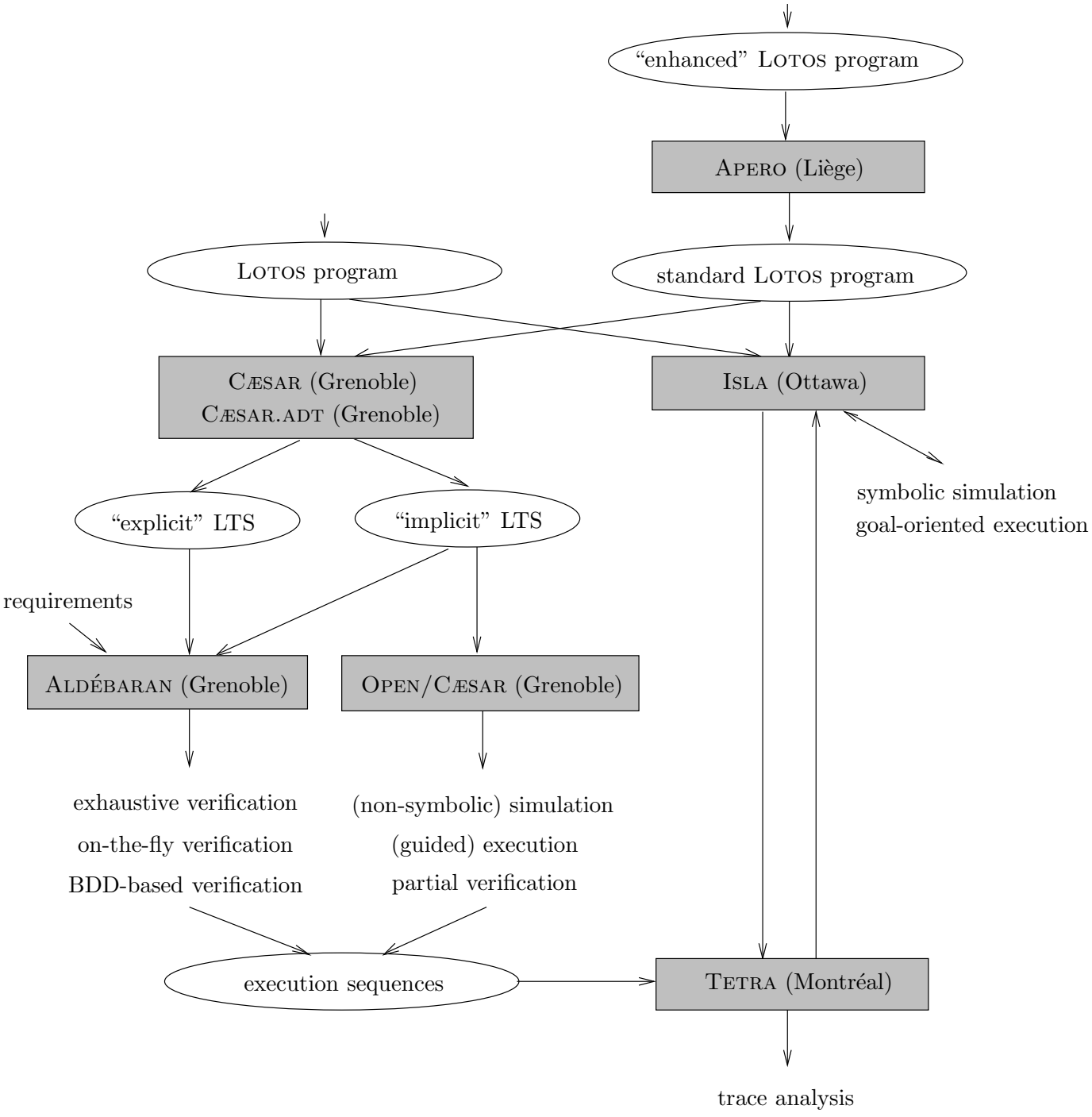
[EUCA/ULg/01] F. Bierbaum, *Liège's comments on document EUCA/GR/01 – Current Situation*. 3 pages. In French.

[EUCA/GR/03] H. Garavel, *Minutes of the 2nd* EUCALYPTUS *meeting – Montréal, September the 10th 1993*. 7 pages. In French.

[EUCA/GR/04] H. Garavel, *Minutes of the 3rd* EUCALYPTUS *meeting – Madrid, January the 11th 1994*. H. Garavel. 8 pages. In French.

1.2.2 Overall architecture of the EUCALYPTUS toolset

The simplified architecture of the EUCALYPTUS toolset is represented below.



The toolset contains the following tools:

APERO (Liège): The LOTOS language features two clearly separated parts, for the specification of data structures and dynamic behaviors respectively. The data part is issued from the language ACTONE, which is based on a highly abstract model (multi-sorted algebras). This yields considerable expressive power to the language but turns out to be too fundamental with respect to the usual needs in dynamic system specifications. Moreover this model lacks some theoretical computability properties (decidability), so that existing software tools (simulators, compilers, verifiers, etc.) have to put some restrictions on this model, which the specifications have to fulfill.

Furthermore, when taking a look at existing LOTOS specifications, it appears that the description of the needed data types is often very large. This lack of concision had already been identified by G. Scollo in 1986 who proposed to extend the language definition with shorthand notations. However, most of the extensions proposed by G. Scollo were not included in the standardized definition of the language. Thus, the problem remained unsolved as system descriptions using the extended language were neither internationally accepted by the scientific community nor tractable by LOTOS related tools.

To tackle this problem Liège proposed in 1991 a first tool called DAFY(DatA FacilitY compiler). The main function of DAFY was the translation of the aforementioned shorthand notations (or ACTONE extensions) into standard LOTOS.

APERO (a loose acronym for Act one PrE-pROcessor) is the new version of DAFY. It seeks to offer a technical solution allowing simple and compact specification of common data structures in LOTOS while keeping compatibility with the standard language, and therefore with existing tools. APERO will not only improve the pre-processing functionality of DAFY, but also extend it with a concept of infinite virtual library. Refer to section on tool improvements for details.

CÆSAR (Grenoble): CÆSAR is a compiler which translates LOTOS descriptions into labelled transition systems (LTSs) i.e., transition machines the transitions of which are labelled by action names. CÆSAR interfaces a number of verification tools for LTSs and temporal logic evaluators, including ALDÉBARAN (see below).

CÆSAR translation algorithms proceed in several steps. First the LOTOS description is translated into a simplified process algebra called SUBLOTOS. Then an intermediate Petri Net model is generated, which provides a compact, structured and user-readable representation of both the control and data flow. Eventually the LTS is produced by performing reachability analysis on the Petri net.

CÆSAR does not handle all LOTOS specifications, but only those who have static control features. CÆSAR can not handle LOTOS features involving dynamic creation of processes and gates: process recursion is not allowed on the left and right hand part of `|[...]|`, nor on the left hand part of `>>` and `[>`. Despite

these restrictions, the subset of LOTOS handled by CÆSAR is large and usually sufficient for real-life needs.

The current version of CÆSAR allows the generation of large LTSs (some million states) within a reasonable lapse of time. Moreover, the efficient compiling algorithms of CÆSAR can also be exploited in the framework of the OPEN/CÆSAR environment (see below).

CÆSAR.ADT (Grenoble): CÆSAR.ADT is a compiler that translates the data part of LOTOS specifications into libraries of C types and functions.

Each LOTOS sort is translated into an equivalent C type and each LOTOS operation is translated into an equivalent C function (or macro-definition). CÆSAR.ADT also generates C functions for comparing and printing abstract data types values, as well as iterators for the sorts the domain of which is finite.

The user must indicate to CÆSAR.ADT which LOTOS operations are "constructors" and which are not. CÆSAR.ADT does not allow non-free constructors ("equations between constructors"). However, it is always possible to transform a LOTOS specification in order to remove equations between constructors.

CÆSAR.ADT is fast: translation of large programs (several hundreds of lines) is usually achieved in a few seconds. CÆSAR.ADT can be used in conjunction with CÆSAR, but it can also be used separately to compile and execute efficiently large abstract data types descriptions.

ALDEBARAN (Grenoble): ALDÉBARAN is a tool for verifying communicating systems, represented by LTS. For instance, one can check the LTS of a protocol against the LTS of the service implemented by the protocol. Both LTSs are generated using CÆSAR and compared using ALDÉBARAN. It is also possible to specify protocol properties using temporal logic formulas that can be evaluated on the protocol LTS.

It allows the reduction of LTSs modulo various equivalence relations (such as strong bisimulation, observational equivalence, delay bisimulation, $\tau^*.a$ bisimulation, branching bisimulation, and safety equivalence). It also allows to perform comparison according to strong bisimulation preorder, $\tau^*.a$ preorder, or safety preorder.

The verification algorithms used in ALDÉBARAN are based either on the Paige-Tarjan algorithm for computing the relational coarsest partition, or on the "on-the-fly" techniques proposed by Fernandez-Mounier, or on symbolic LTS representation using Binary Decision Diagrams (BDDs), or on compositional algorithms.

ALDÉBARAN has diagnosis capabilities that provide the user with explanations (execution sequences) when two LTSs are found to be not equivalent.

OPEN/CÆSAR (Grenoble): OPEN/CÆSAR is an extensible environment based upon CÆSAR and CÆSAR.ADT. It allows user-defined programs for simulation, execution, verification (partial, on-the-fly, etc.), and test generation to be developed in a simple and modular way. Users can easily extend the environment by adding their own modules to fit their specific needs.

Various modules have already been created in the OPEN/CÆSAR framework, including:

- two interactive simulators (with shell-like and X-window interfaces),
- a random execution tool,
- a deadlock detection tool based on G. Holzmann's technique,
- a reachability analysis tool (with $\tau^* a$ on-the-fly reduction),
- a sequence-searching tool,
- etc.

ELUDO (Ottawa): The LOTOS environment under development at the University of Ottawa is named ELUDO (an acronym for "Environnement LOTOS de l'Université d'Ottawa"). The main currently operational tools in ELUDO are ISLA, SELA, and GOAL. ELUDO includes common facilities available to all the tools, including:

1. A graphical interface for ASCII terminals, based uopn the "curses" library of UNIX;
2. A graphical interface based upon X-windows;
3. A LOTOS translator, which is a stand-alone program that performs initial verification and preprocessing on the LOTOS specification to be analyzed under ELUDO. It may be executed from inside ELUDO or as and independent LOTOS specifier's development tool. The main functions supported by the translator are the following:
   - Lexical, syntactical and static semantics analysis;
   - Translation to a Prolog internal representation suitable for the ELUDO tools, which are written in Prolog; the translation of the LOTOS specification into its equivalent Prolog form is done only once;
   - Generation of cross-references of processes, gates and types;
   - Creation of a user-defined type library, to replace the default standard library;
   - Output of a parse tree of the specification;
   - Pretty-printing of the LOTOS specification.

ISLA (Ottawa): ISLA provides a step-by-step execution mode which allows to simulate the sequence of possible actions that are permitted by a LOTOS specification. The execution of a LOTOS specification can be represented as a tree, where the root of the tree is the specification itself, the intermediate nodes are behavior expressions and the branches of the tree represent LOTOS actions.

The user may choose to simulate the whole specification at once, or only parts of it (certain processes). At each step, during simulation, the user is prompted with a menu of possible next actions. The user chooses the next action to be executed and, if the selected action requires data to be supplied by the environment (the user plays the role of the environment), then data must be entered for the simulation to continue.

A menu-driven facility prompts the user with appropriate choices for data. Also, at any point during simulation, the user may ask to see the current behavior of the system or the behavior that will result by executing one of the possible next actions.

Furthermore, ISLA displays the complete set of execution paths that have been exercised by the user during the current simulation session, in the form of a tree. This allows her to check, for example, where in the execution tree a guard was evaluated and where certain value identifiers were instantiated. So, if certain chosen values did not lead to the desired sequence, the user can back up to a point where a different value can be entered. Therefore, the user may return to a previous execution point, and redo execution from that point with different choices.

It is also possible to save the sequence of actions, executed up to some point in the tree, in the memory or in an external file, thereby gaining the possibility of continuing the simulation, at a later time, from where it was left off.

SELA (Ottawa): SELA is a symbolic expander for LOTOS. The step-by-step execution mode provided by ISLA is very useful, but it is also time consuming.

SELA allows one to compute the tree of all possible next actions from the current point, or any given point in the tree. This is known as the symbolic execution tree because expressions are computed in terms of (not necessarily) bounded value identifiers. In terms of LOTOS theory, the calculation of this tree is called 'expansion'. When generating a symbolic tree, guards and predicates, whose values depend on interactions with the environment, are assumed to be true. In addition, the user is required to set limits on the depths and widths of the symbolic tree to be generated.

Although calculation of the symbolic tree may not terminate, it can yield finite initial subtrees of an infinite monolithic specification equivalent to the original one.

GOAL (Ottawa): GOAL allows so-called "goal-oriented" execution for LOTOS. Both ISLA and SELA suffer from the well-known problem of state explosion: for most practical specifications, execution trees grow very quickly.

Goal-oriented execution attempts to relieve this problem. In this type of execution, the tool attempts to find execution sequence(s) leading to a certain action or sequence of actions (these are the "goals"). The specification is scanned statically to find where the action(s) can be found. Then the inference rules are applied, taking into consideration this information. The result are sets of execution sequences reaching the goals.

TETRA (Montréal): TETRA (TEst and TRace Analyzer) compares a given trace of interactions with a reference specification written in LOTOS, checking whether an execution history of the specification could produce the given trace. The tool allows for two modes of analysis:

- off-line trace analysis, where the reference specification is compiled together with the traces to be analyzed, which are written in the form of LOTOS processes;
- on-line trace analysis, which compiles the reference specification alone and analyses the interactions of a trace one after the other as they are received form another site executing/simulating the implementation under test. The result of a trace analysis is either "valid trace" or "invalid trace".

In the latter case, an optional error diagnostic facility provides indications about possible causes of the discrepancy between the trace and the specification according to various error hypotheses.

Besides trace analysis, TETRA also has an option to validate test cases and their verdicts with respect to a reference specification which defines the expected behavior of the tested system. It establishes in this case whether or not the branches of a test case conform to the reference specification. Diagnostic analysis for erroneous branches is possible as well.

1.2.3 Activities performed in Task 1 (Tool assessment)

Before assessing the EUCALYPTUS toolset components, Liège has evaluated another existing LOTOS toolset, denoted LITE 3.0, which has been developed within the ESPRIT LOTOSPHERE project and distributed by ITA (Information Technology Architecture BV). The evaluation effort focused on SMILE, which is the symbolic simulator of LITE and also its most consequent and useful tool. SMILE happened to be time and memory consuming. When faced with Liège's OSI95 LOTOS specification (2200 lines of data types + 2200 lines of processes), it took 4 hours and 270 Mb of swap-space to unfold two steps in the specification [EUCA/ULg/01].

The first tools from the future EUCALYPTUS Toolset that Liège evaluated is the CADP (CÆSAR.ADT + CÆSAR + ALDÉBARAN) toolkit from Grenoble (W version of January the 29th 1993) [EUCA/ULg/04]. Some conclusions are summarized hereafter:

- CÆSAR.ADT, the data type compiler, is able to support very large abstract data type specifications. Its only shortcomings are:
  - It does not support the formal data types definitions and the actualizations;
  - Values of complex abstract data types cannot be enumerated.
- CÆSAR, the model generator, works perfectly on medium sized LOTOS specifications. On large ones, and in particular on large constraint-oriented ones, such the Liège's OSI95 specifications, CÆSAR currently faces difficulties. The most important limitation is the explosion of the size of the intermediate model (Petri Net) used by CÆSAR. Some solutions have been suggested such as the interleaving of the optimization and generation phases of CÆSAR to prevent the Petri Net from getting too large.
- As regards ALDÉBARAN, the model verifier, no problem was detected except in the minimization of the Labelled Transition System model modulo the testing equivalence.

Liège also started the assessment of a second component of the EUCALYPTUS toolset: the ELUDO toolkit (ISLA + SELA) from Ottawa. This work is still under progress.

Reports

[EUCA/ULg/01] C. Pecheur, L. Leonard, *Evaluation of* LITE*, a Toolset for* LOTOS. March 1993, 11 pages.

[EUCA/ULg/04] F. Bierbaum, M. Jankowski, *Assessment of the* CÆSAR/ALDÉBARAN *toolset on the OSI95* LOTOS *specifications.* October 1993, 24 pages.

1.2.4 Activities performed in Task 2 (Tool improvement)

Activities in Grenoble

Grenoble spent considerable efforts in improving the CÆSAR/ALDÉBARAN toolset [EUCA/GR/02]. This effort was motivated by the results of the tool assessment evaluation by Liège, Grenoble's own experience in dealing with the LOTOS specifications provided by Liège, and the remarks of many other users.

A new version of the toolset was released in September 1993, which provided significant improvements on the previous version of January 1993:

1. The algorithm used by CÆSAR to generate the Petri Net was modified in order to avoid a combinatorial explosion that sometimes occurred with the "disable" operator of LOTOS. The problem was faced on the first LOTOS specification provided by Liège.

2. The same algorithm was also modified in order to avoid the generation of many useless transitions which are later destroyed during the optimization phase (optimization V4). The modified algorithm gives good results on various large LOTOS programs, including the *Plain Old telephony System* developed by SICS (Sweden) and the *Flight Warning Computer* of Airbus A330/340 specification developed by Aerospatiale (France).

3. The state vector of CÆSAR was reorganized in order to reduce its size, thus allowing more states to be stored. The average gain in size is of 21.5%. Additionally, the part of this state vector where it is possible to compute safely a hash-code was extended from 18% to 81%, therefore reducing the number of potential collisions and improving the speed of the simulation phase.

4. The C code generated by CÆSAR.ADT for structured types was improved. Record types are now implemented with a minimal number of bits. Practically, this optimization reduces of $\approx 40\%$ the number of bytes needed to implement structured types. In some cases, this number of bits is divided by 8!

5. The C code generated by CÆSAR.ADT for constant and macro-definitions was improved. Additional checkings were added to detect recursive constant and macro definitions, and to override various limitations of the standard C compiler available on SUN workstations.

6. The OPEN/CÆSAR environment was extended with a new module, implementing a "generic state table". Using this module, various tools have been developed, including a tool that searches for execution sequences matching a given pattern.

This tool will certainly be helpful to analyze the OSI95 specifications; it can be seen as a complement of the TETRA tool developed by Montréal.

7. The conventions for interfacing the C code automatically generated by CÆSAR and CÆSAR.ADT with the external C code provided by the user have been improved, so as to generate unique names by default. Therefore, it is no longer necessary to insert special comments in the LOTOS specifications to solve the conflicts created by operation overloading.

8. Missing libraries (like `OCTETSTRING`) were added and bugs were fixed in existing libraries.

9. An option was added to CÆSAR in order to specify where temporary files are to be stored. This option solves a problem faced by Liège.

10. Various bugs were fixed in the ALDÉBARAN tool.

Activities in Liège

In this task Liège is rebuilding its abstract data types (ADT) pre-processor DAFY tool. The new tool, called APERO, will not only improve the pre-processing functionality of DAFY, but also extend it with a concept of infinite virtual library. APERO is thus based on two complementary mechanisms:

- a pre-processor that catches and expands some language extensions into standard LOTOS,

- an extension of the standard library mechanism, giving access to non-finite collections of generic data types (records, enumerated types, etc.).

Both functions rely on a generic text transformation algorithm and externally specified transformation rules. On one hand, this facilitates the modification and/or extension of the provided extensions and library types. On the other hand, this allows several alternative transformation rules for the same set of facilities, where the translated specification is tuned for several environments (human reading, compiler, simulator, etc.).

The generic name for the data type processing facilities of the EUCALYPTUS toolset is APERO (which is a loose acronym for Act one PrE-pROcessor). The pre-processor for syntax extensions is APERO.SYN, the library generator is APERO.LIB. The APERO language is used to specify transformation rules for both tools.

The common text transformation algorithm is derived of "Macro By Example", a macro processing formalism used in the Scheme language. An elementary transformation rules consists of a pair <source pattern, target pattern>. The source pattern is matched against the source text and the corresponding target pattern is then expanded into target text. A variable binding mechanism allows parts of the source text to be transferred to the target text.

APERO has been developed using the language ML (NJ/SML 0.93). ML is a strongly typed functional language, which eases the complex manipulation of symbolic structures that have to be implemented.

A preliminary prototype, offering only the library extension mechanism, had been previously developed. This work has been published [Pec93].

The preliminary studies and the design phase for APERO have been carried during March, April and May 93. June to December have been fully devoted to the programming effort. A working first version of both tools was available at the beginning of January. A first version of the APERO package, with reasonably complete definition files and documentation, is to be distributed among the project partners at the beginning of February 1994.

Activities in Ottawa

All the components of the ELUDO environment have been improved in the framework of the EUCALYPTUS cooperation:

1. The bottom-up value expression evaluator of ISLA was enhanced, and a top-down value expression evaluator was developed.
2. Work was done to integrate the SELA and GOAL tools into the ELUDO environment.
3. A major restructuring occurred for the execution control of the SELA tool, in order to handle interrupts and to allow execution monitoring. A handler was developed, which facilitates the export of symbolic execution trees.
4. The development of a Static Derivation Path extractor and a goal-oriented inference rule engine for the GOAL tool is going on.

Additionally, Ottawa fixed various bugs: the syntax analyzer was repaired to make it process special identifiers properly and to be "case insensitive" as required by the LOTOS ISO standard.

Ottawa also worked on the documentation aspects: a reference guide and on-line manual pages have been written for ELUDO.

The improved version of ELUDO was installed on the EUCALYPTUS server in September 1993. Ottawa provided assistance to Liège in setting up and running the X-windows version of ELUDO.

Activities in Montréal

The effort of Montréal in EUCALYPTUS has been mainly devoted to the task of adapting TETRA to the new version of ELUDO.

TETRA uses the internal representation of LOTOS specifications produced by the LOTOS compiler of the ELUDO environment, as well as its inference engine and data evaluator. These components of the ELUDO environment were improved in the last versions of the toolset.

The adaptation of TETRA to the new version consisted of bringing the necessary modifications concerning the use of the internal representation and the inference rules.

Special needs of TETRA, such as the translation of external observed actions from LOTOS format to their corresponding internal format, were taken into consideration by Ottawa.

In addition, the observation mechanism, previously implemented on top of an old version of ISLA to suit the on-line trace analysis feature of TETRA, is now seen to be better integrated to the new ISLA system in the context of integration of the EUCALYPTUS toolset.

Most of the adaptation task is by now completed, but remains to be tested.

Publications

[GT93] H. Garavel, Ph. Turlier, CÆSAR.ADT: un compilateur pour les types abstraits algébriques du langage LOTOS. In R. Dssouli, G. v. Bochmann, eds, Actes du Colloque Francophone pour l'Ingénierie des Protocoles CFIP'93 (Montréal, Canada), September 1993.

[Pec93] C. Pecheur, *VLib: infinite virtual libraries for LOTOS*, in A. Danthine, G. Leduc, P. Wolper, eds, Protocol Specification, Testing and Verification, XIII, Elsevier Science Publishers (North Holland), Amsterdam, 1993, 29-44.

Reports

[EUCA/GR/02] H. Garavel, *Contribution of Grenoble to the progress of the* EUCALYPTUS *project*. September 1993, 21 pages. In French.

[EUCA/UO/01] J. Sincennes, *Subset of* LOTOS *accepted by* ISLA *and related restrictions*, July 1993. In French.

1.2.5 Activities performed in Task 3 (Tool convergence)

This task is composed of several sub-tasks:

Evolution of the ELUDO environment Ottawa has made significant changes to the ELUDO environment in order to allow the integration of foreign tools, and especially TETRA. The most important improvements are the following:

- structural reorganisation and modularisation;
- creation of "sub-tools", including a filename tool for navigation into directories and file selection;
- improvement of the value expression editor;
- creation of a manipulation tool for constants;
- addition of various widget management functionalities, including a library for drawing vertical trees.

Ottawa provided assistance to Montréal for integrating TETRA in the ELUDO environment and for upgrading TETRA to the new ISLA kernel.

Definition of a common format for execution sequences
Verification tools such as ALDÉBARAN and OPEN/CÆSAR can discover "faulty" execution sequences, i.e., execution sequences that do not correspond to a permitted behavior of the service or the protocol under consideration.

A desirable feature of the toolset would be the possibility to replay and examine these faulty execution sequences using the ISLA and TETRA tools: these tools operate at closer level from the source LOTOS program than ALDÉBARAN and OPEN/CÆSAR, and can therefore provide better error diagnosis.

In order to allow cooperation between these tools, the definition of a common format for execution sequences was undertaken and is now almost complete.

Ottawa adapted its tools in order to accept "external" execution sequences to be analyzed by TETRA and replayed by ISLA. In this context, Ottawa worked on the parsing and reconstruction of LOTOS events.

Definition and implementation of a common graphical user interface The integration of all existing tools in a coherent toolset requires the development of a common graphical user interface.

The X-windows system (version X11R5) was selected by the EUCALYPTUS consortium, because the graphical interface of ISLA is based on X11R5.

Grenoble has experienced various environments for developing graphical user interfaces, namely MOTIF and associated libraries, INTERVIEWS, GRIF and XTPANEL.

As a result of this evaluation, the XTPANEL tool developed at Stanford University was selected and adopted by all the partners of the EUCALYPTUS project.

Grenoble has defined the requirements for the graphical interface of the CÆSAR, CÆSAR.ADT, ALDÉBARAN, and OPEN/CÆSAR tools [EUCA/GR/02].

Grenoble developed a prototype user interface (based on XTPANEL) for the ALDÉBARAN tool. This prototype was sent to all the partners to illustrate the capabilities of XTPANEL.

Montreal developed a similar user interface for the TETRA tool, using XTPANEL.

Liège has defined the requirements for the graphical interface of the APERO tool [EUCA/ULg/02].

On this basis, the EUCALYPTUS meeting in Madrid (January 1994) determined the general appearance of the common user interface and fixed the schedule for its realization.

Reports

[EUCA/ULg/02] C. Pecheur, *Interface Specification for ADT Pre-processing Tools*. June 1993, 5 pages.

[EUCA/GR/02] H. Garavel, *Contribution of Grenoble to the progress of the* EUCALYPTUS *project*. September 1993, 21 pages. In French.

1.3 Dissemination of results

The EUCALYPTUS project has a good international visibility. It was presented in several international workshops and conferences, including the following:

- *13th IFIP Symposium on Protocol Specification, Testing and Verification (Liège)* organized by André Danthine, Guy Leduc and Pierre Wolper :
  - Joseph Sifakis, Hubert Garavel and Guy Leduc gave a one-day tutorial on formal verification techniques.
  - Charles Pecheur presented a paper describing the principles of the APERO tool.
- *Colloque Francophone pour l'Ingénierie des Protocoles CFIP'93 (Montréal)* organized by Rachida Dssouli and Gregor v. Bochmann.

  Hubert Garavel presented a paper describing the improvements of the CÆSAR.ADT tool.

The improved version of the CÆSAR/ALDÉBARAN toolset (version X, released in September 1993) has been installed in more than 20 sites.

Also, the EUCALYPTUS participants are actively contributing to the ongoing standardization effort intended to enhance the LOTOS language with new features (ISO/IEC JTC1/SC21/WG1 New Work Item on "Extended LOTOS").

1.4 Conclusions and objectives

Formal description techniques like LOTOS were defined to allow a precise and unambiguous description of complex reactive systems. LOTOS features (and especially the "constraint-oriented" style) are especially useful in describing large protocols and services.

On the other hand, it is highly desirable to check formal descriptions (using both semi-automatic and automatic analysis tools) before they are standardized.

A number of software engineering tools for LOTOS have been developed for this purpose during the past decade. They provide interesting functionalities which attempt to cover the full life-cycle of the elaboration of LOTOS descriptions.

However, it appears that all these tools have problems in dealing with large LOTOS specifications, such as the *Enhanced Transport Service* description elaborated by Liège for high-speed networks. The problems faced are mostly due to the size of this specification, the hundreds of processes running in parallel, and the systematic use of the constraint-oriented style.

As a first result, the EUCALYPTUS project revealed the fact that most existing tools for LOTOS could not handle properly the large constraint-oriented descriptions developed by Liège. This is true for both Grenoble and Ottawa tools, and also for other tools developed in the ESPRIT LOTOSPHERE project.

However, it should be pointed out that it was possible to use these tools to analyse and verify formally various subsets of the *Enhanced Transport Service* specifications, as reported in the deliverables.

During the first year of the EUCALYPTUS project, the aforementioned problems led to significant tool improvements. But it is fair to say that these improvements do not solve all the problems reported up to date.

In the second year of the EUCALYPTUS project, it will be a real challenge for software developers to improve their tools in order to overcome the difficulties experienced during the first year of the project. Work has already started and will continue in this direction.

Another positive point of the EUCALYPTUS collaboration is the development of a common graphical interface for the integrated toolset. Designing a user-friendly interface for complex applications such as formal verification, test generation, etc. is not obvious nor easy. However, it is considered as a necessary feature by most users and the EUCALYPTUS project will provide a real opportunity to meet their expectations.

Although not explicitly planned in the Technical Annex of the EUCALYPTUS contract, the participants are actively preparing a review with demonstrations, which could take

place during the second half of April 1994 in Liège. A similar review could be organized in Ottawa to present the half-term results of the project to Canadian industrials.

In spite of the geographical distance between Canada and Europe, the cooperation is really effective. During the first year, a common meeting was held every three months (more frequently than initially foreseen in the Technical Annex).

We believe that EUCALYPTUS has a good international visibility, since several research groups have expressed the wish to join the project or to participate to a possible continuation, if any.

In 1994, we intend to continue the dissemination of results. Presentations are already planned, for instance in the CONCUR'94 conference (Kista, Sweden) where Joseph Sifakis will give a tutorial on verification techniques.

Also, we expect that the final EUCALYPTUS workshop — to be held in Grenoble in December 1994 — will constitute an important event for the protocol and distributed systems community.

Chapter 2

Financial situation

TO BE PROVIDED BY GRENOBLE AND LIEGE