LOTOS: State of the art and perspectives (Extended abstract)

Abdellatif Obaid
Universit du Quebec a Hull,
Departement d'informatique
B.P. 1250, Succ. B, Hull,
Quebec, J8X 3X7, Canada
E-mail: obaid@uqhull.bitnet

LOTOS is a formal language for the description of distributed systems. Its development began in the early 1980s. Based on the semantic model of CCS introduced by Milner [29], a first draft document was presented in 1985. Later on, a combination of CCS and Hoare's CSP [17] was suggested as a semantic background for the definition of the language. Since 1989 we have a stable version of LOTOS that was adopted as a standard formal description technique [19] in which the data part uses the abstract data type definition language ACT ONE. Since then many researchers throughout the world have subsequently used the language for the development of their systems.

The language has many aspects that make it attractive to people in the area of distributed systems:

- It is general enough to allow the description of many classes of systems (e.g. communication protocols, telephone systems, operating systems).
- It is abstract enough to allow systems to be described independently of any hardware or architecture.
- It has a good verification theory that allows for validation.

There have been many stages in the use of the language. First, its usefulness was demonstrated by numerous specification examples mainly in the area of communication protocols and services [20, 21, 18, 11, 7]. Then tools were developed in order to execute these specifications [25, 26, 27, 34]. Tools were also developed to support verification [2, 14]. Later on there were contributions towards the definition of design concepts and methodologies, notably the specification styles [35]. As a formal description method, LOTOS was also used for conformance testing. Many researchers have used the language to design test cases and testing processes [4, 36, 23].

Since a formal specification describes a system that is to be implemented, it is desirable to use LOTOS as an implementation language. There are systems that (partially) implement LOTOS specifications. In many cases, transformations to a target language or model are used [27, 1, 32].

LOTOS has been used in many applications for specification, design and verification. Although most of the specifications written in LOTOS are in the area of protocols, other types of applications are emerging: operating systems, distributed algorithms, local area networks, real-time systems, and so on [18, 7, 16, 31].

There is also a growing effort to come up with (interactive) design environments [30, 9]. This effort is linked to the existence of G-LOTOS, the graphical representation for LOTOS [22]. In spite of this, to our knowledge, there is no design environment completely based on LOTOS nor is there a methodology to support the whole life cycle of a software that was specified in the language.

In terms of future perspectives, we can expect developments in the following major directions:

- Use of LOTOS for real-size examples in different areas of distributed systems. This will show even more clearly the usefulness of the language.

- Use of LOTOS in real-time systems design and implementation. This requires the introduction of (upward) compatible versions of LOTOS which include timing. Inspite of the efforts that have been made in this area, timing has still not been adopted in any "official" version of the language [41, 24].

- Use of LOTOS in testing systems. There have been many contributions in the development of test suites based on extended finite state machine models. But only a few contributions has been made so far for LOTOS [6, 15]. However LOTOS formalism makes it possible to predict that future developments will occur in this area.

- Development of complete design discipline and environments. LOTOS allows for the specification of data communication and data flow between process. It also allows specification of distribution, system structuring and deveopment by refinements [30, 10, 9]. These features can and should be used to come up with complete and rigorous design methodologies. Specification styles such as resource, state and constraint oriented styles [35], are good formal basis in this area.

- Development of better tools and representations for the abstract data type part of the language. This will certainly ease its use.

- Definition and use of an object-oriented approach based on or for LOTOS. The popularity of this approach is based on considerations such as: cost optimization, reusability, modularity, structuring, etc. There have been some discussions on proposals for object-oriented versions of LOTOS as well as the use of the concept of modules [28, 8, 38]. We believe that this should be enforced by tools (compilers, run-time systems, etc.)

- Introduction of ways of expressing user requirements. This will permit formulation of properties such as safety and liveness at the specification level. Recent contributions have been made in this area [12, 13, 23].

- Finally, we believe that the use of graphical representations of LOTOS such as G-LOTOS can help in system design provided that they offer a concise and unambiguous mean to express specifications. In [3] we have some discussions of these aspects.

We can see that the existence of this language has triggered many activities mainly among researchers. Providing design tools and testing techniques can only help in introducing even more the language within the industrial environment where there are still hesitations in using formal description techniques.

References

[1] Bochmann G.V., Gao Q. and Wu C. Distributed Implementation of LOTOS. In [Vuong 90].

[2] Bolognesi T. and Caneve M. Equivalence Verification: Theory, Algorithms and a Tool. In [van Eijk 89] 303-326.

[3] Bolognesi T. A Graphical Composition Theorem for Networks of Lotos Processes", Proc. 10th International Conference on Distributed Computing Systems, 1990. 88-95.

[4] Brinksma E., A Theory for the Derivation of Tests. In [Aggarwal 89] 63-74.

[5] Brinksma E., Specification Modules in LOTOS. In [Vuong 90] 137-156.

[6] Brinksma E., A Framework for Test Selection, in [Pehrson 91].

[7] Cam R.C. and Vuong S.T., A Formal Specification, in LOTOS, of a Simplified Cellular Mobile Communication System. In [Vuong 90] 485-499.

[8] Cusack E., Rudkin E. and Smith C., An Object-oriented Interpretation of Lotos, in [Vuong 90].

[9] de Moura J. and Cunha P., Structured Development of LOTOS Specific ations Using G-LOTOS, Personal communication, 1991.

[10] de Souza L. W. and Pires L. F., Step-wise refinment Design Examp le Using LOTOS. In [Quemada 91]. 189-306

[11] Faci M., Logrippo L. and Stepien B., Formal Specification of Telephon e Systems in LOTOS: The Constraint-Oriented Approach. Computer Networks and ISDN Systems 21 (1991) 53-67.

[12] Gallouzi S. , Logrippo L. and Obaid A., A Hoare-style proof system for LOTOS. In [Quemada 91].

[13] Gallouzi S. , Logrippo L. and Obaid A., An Expressive Trace Theory for LOTOS. In [Pehrson 91].

[14] Garavel H. and Sifakis J., Compilation and Verification of LOTOS Specifications. In [Logrippo 90] 379-394.

[15] Gueraichi D. and Logrippo L., Derivation of Test Cases for LAP-B from a LOTOS Specification. In[Vuong 90] 361-374.

[16] Haj-Hussein M. and Logrippo L.. Specifying Distributed Algorithms in LOTOS. Proceedings of the Computer Networks Conference, Wroclaw, 1991.

[17] Hoare C.A.R., Communicating Sequential Processes. Prentice-Hall, 1985.

[18] Iglewski M. and Obaid A., Specification of the IEEE 802.5 Token Ring LAN in LOTOS. Rapport de Recherche, RR 91/01-2, Departement d'informatique, Universite du Quebeca Hull, 1991.

[19] IS 8807 LOTOS, A Formal Description Technique Based on the Temporal Ordering of Observational Behavior. 1989.

[20] ISO/IEC TR 9571. LOTOS Description of the Session Service, 1989.

[21] ISO/IEC TR 9572. LOTOS Description of the Session Protocol, 1989.

[22] ISO/IEC JTC1/SC21 N3253. G-LOTOS: A Graphical Syntax for LOTOS (1989).

[23] Leduc G., Relations d'impl mentation et transformations autoris es d 'unesp cification LOTOS, R seaux et informatique r partie, Vol. 1, n.1, 1991. 59-88.

[24] Leduc G., An Upward Compatible Timed Extension to LOTOS, Formal Description Techniques IV. North-Holland, 1991 (To appear).

[25] Logrippo L., Obaid, A., Fehri, M. and Briand, J.P., "LOTOS: An Executable Specification Language for Distributed Systems", Software Practice and Experience, Vol. 122, n.4. 1988. 365-385, 1988.

[26] Logrippo L. , The University of Ottawa LOTOS Toolkit. In [Quemada 91].

[27] Maras J.A., de Miguel T. and van Thienen H., The Implementation of a Specification Language for OSI Systems. In [van Eijk 89] 409-421.

[28] Mayr J., Specification of Object-oriented Systems in Lotos, in [Turner 89]. 121-134.

[29] Milner R., A Calculus of Communicating Systems, volume 92 Lecture Notes in Computer Science. Springer-Verlag, 1980.

[30] Obaid A., Guevel G. et Saumure R., EGL: Un outil de conception des protocoles de communication, Rapport de Recherche RR 90/10-01, Departement d'informatique, Universite du Quebec a Hull, octobre 1990.

[31] Pecheur C. , An Overview of the LOTOS Specification of Chorus V3. Report No. S.A.R.T. 89 - 03 -13, Universite de Liege, B28, D partement de systemes et Automatique, mai 1990.

[32] Queiroz J., Sehrouchni A. , Cunha P. and Jajm E., PIL: A Tool for Pre-implementation of LOTOS. In [Quemada 91].

[33] Quemada J., Azcorra A. and D. Frutos, TIC: A Timed Calculus for LOTOS. In [Brinksma 90] 195-209.

[34] van Eijk P.H.J., The Design of a Simulator Tool. In [van Eijk 89] 351-390.

[35] Vissers C. , Scollo G. and van Sinderen M., Architecture and Specification Styles in Formal Distributed Systems. In [Aggarwal 89].

[36] Wezman C.P, The CO-OP Method for Compositional derivation of Conformance Testers, in [Turner 89].

[37] Aggarwal S., Sabnani K. (eds.), Protocol Specification, Testing, and Verification, VIII. North-Holland, 1989.

[38] Brinksma E., Scollo G. and Visser C. (eds.), Protocol Specification, Testing, and Verification, IX. North-Holland, 1990.

[39] Logrippo L. , Probert R.L. and Ural H. (eds.), Protocol Specification, Testing, and Verification, X. North-Holland, 1990.

[40] Pehrson B. , Jonsson B. and Parrow J. (eds.), Protocol Specification, Testing, and Verification, XI. North-Holland. (To appear).

[41] Maras J. , Vazquez E. (eds.), Formal Description Techniques, III. North Holland. (To appear).

[42] Turner K. (ed.), Formal Description Techniques I. North-Holland, 1989.

[43] van Eijk P.H.J. , Vissers C.A., and Diaz M. (eds.), The Formal Description Technique LOTOS. North-Holland, 1989.

[44] Vuong S.T. (ed.), Formal Description Techniques, II. North-Holland,1 990.