AS/NZS ISO/IEC 15437:2003
ISO/IEC 15437:2001

Australian/New Zealand Standard™

Information technology—Open distributed processing—Enhancements to LOTOS (E-LOTOS)

Standards Australia

STANDARDS NEW ZEALAND
Paerewa Aotearoa

This Joint Australian/New Zealand Standard was prepared by Joint Technical Committee IT-015, Software Engineering. It was approved on behalf of the Council of Standards Australia on 6 May 2003 and on behalf of the Council of Standards New Zealand on 22 April 2003. It was published on 4 June 2003.

The following are represented on Committee IT-015:

Australian Computer Society
Australian Information Industry Association
Australian Society for Technical Communication (NSW)
Australian Software Metrics Association
Griffith University
New Zealand Organisation for Quality
Quality Society of Australasia
Software Engineering Australia (Qld)
Software Quality Association (ACT)
Software Quality Association (NSW)
Software Verification Research Centre
Sydney SPIN Group (Software Process Improvement Network)
Systems Engineering Society of Australia
University of New South Wales
University of South Australia
University of Technology, Sydney

Keeping Standards up-to-date

Standards are living documents which reflect progress in science, technology and systems. To maintain their currency, all Standards are periodically reviewed, and new editions are published. Between editions, amendments may be issued. Standards may also be withdrawn. It is important that readers assure themselves they are using a current Standard, which should include any amendments which may have been published since the Standard was purchased.

Detailed information about joint Australian/New Zealand Standards can be found by visiting the Standards Australia web site at www.standards.com.au or Standards New Zealand web site at www.standards.co.nz and looking up the relevant Standard in the on-line catalogue.

Alternatively, both organizations publish an annual printed Catalogue with full details of all current Standards. For more frequent listings or notification of revisions, amendments and withdrawals, Standards Australia and Standards New Zealand offer a number of update options. For information about these services, users should contact their respective national Standards organization.

We also welcome suggestions for improvement in our Standards, and especially encourage readers to notify us immediately of any apparent inaccuracies or ambiguities. Please address your comments to the Chief Executive of either Standards Australia International or Standards New Zealand at the address shown on the back cover.

This Standard was issued in draft form for comment as DR 03092.

Australian/New Zealand Standard™

Information technology—Open distributed processing—Enhancements to LOTOS (E-LOTOS)

First published as AS/NZS ISO/IEC 15437:2003.

COPYRIGHT

© Standards Australia/Standards New Zealand

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the written permission of the publisher.

Jointly published by Standards Australia International Ltd, GPO Box 5420, Sydney, NSW 2001 and Standards New Zealand, Private Bag 2439, Wellington 6020

ISBN 0 7337 5284 5

PREFACE

This Standard was prepared by the Joint Standards Australia/Standards New Zealand Committee IT-015, Software Engineering.

This Standard is identical with, and has been reproduced from ISO/IEC 15437:2001, *Information technology—Enhancements to LOTOS (E-LOTOS)*.

The objective of this Standard is to define the syntax and semantics of the enhanced LOTOS language, 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.

As this Standard is reproduced from an international standard, the following applies:

(a) Its number appears on the cover and title page while the international standard number appears only on the cover.

(b) In the source text 'this International Standard' should read 'this Australian/New Zealand Standard'.

(c) A full point substitutes for a comma when referring to a decimal marker.

References to International Standards should be replaced by references to Australian or Australian/New Zealand Standards, as follows:

Reference to International Standard		*Australian/New Zealand Standard*	
ISO		AS/NZS	
8807	Information processing systems—Open systems interconnection—LOTOS—A formal description technique based on the temporal ordering of observational behaviour	4199	Information processing systems—Open systems interconnection—LOTOS—A formal description technique based on the temporal ordering of observational behaviour

CONTENTS

Page

1	Scope	1
2	Conformance	1
3	Normative reference	1
4	Terms, definitions and notation	1
5	E-LOTOS grammar	2
5.1	Lexical Structure	2
5.1.1	Character set	2
5.1.2	Comments and separators	3
5.1.3	Identifiers	3
5.1.4	Reserved words	3
5.1.5	Identifiers classes	4
5.1.6	Non-terminals classes	4
5.2	Syntax of the language	5
5.2.1	Specification	5
5.2.2	Top-level declaration	6
5.2.3	Module body	6
5.2.4	Module expression	6
5.2.5	Module formal parameters	6
5.2.6	Interface expressions	6
5.2.7	Record module expression	6
5.2.8	Interface body	7
5.2.9	Formal parameter list	7
5.2.10	Renaming/Instantiation	7
5.2.11	Equations declaration	7
5.2.12	Simple equations declaration	8
5.2.13	Declarations	8
5.2.14	Expressions	8
5.2.15	Expression Atoms	8
5.2.16	Behaviour expressions	9
5.2.17	Disabling behaviour expression	10
5.2.18	Synchronization behaviour expression	10

Page

5.2.19 Concurrency behaviour expression . . . 10
5.2.20 Selection behaviour expression . . . 10
5.2.21 Suspend/Resume behaviour expression . . . 10
5.2.22 Interleaving behaviour expression . . . 10
5.2.23 Behaviour term . . . 10
5.2.24 Behaviour atom . . . 11
5.2.25 Type expressions . . . 12
5.2.26 Record type expressions . . . 12
5.2.27 Value expressions . . . 12
5.2.28 Record value expressions . . . 13
5.2.29 Patterns . . . 13
5.2.30 Gate parameter list . . . 13
5.2.31 Actual parameter list . . . 13
5.2.32 Exception parameter list . . . 13
5.2.33 Record patterns . . . 14
5.2.34 Behaviour pattern matching . . . 14
5.2.35 Expression pattern matching . . . 14
5.2.36 In parameter list . . . 14
5.2.37 In parameter . . . 14
6 E-LOTOS abstract syntax . . . 14
6.1 Overview . . . 14
6.1.1 Syntactic sugar . . . 15
6.1.2 Abstract syntax . . . 15
6.2 Concrete to abstract syntactic translation . . . 16
6.2.1 Interface body . . . 16
6.2.2 Formal parameter list . . . 18
6.2.3 Declarations . . . 18
6.2.4 Expressions . . . 21
6.2.5 Expression atoms . . . 21
6.2.6 Behaviour expressions . . . 24
6.2.7 Interleaving behaviour expression . . . 24
6.2.8 Behaviour atom . . . 25
6.2.9 Type expressions . . . 31
6.2.10 Record type expressions . . . 32
6.2.11 Record value expressions . . . 32
6.2.12 Gate parameter list . . . 33
6.2.13 Actual parameter list . . . 33
6.2.14 Exception parameter list . . . 33
6.2.15 Record patterns . . . 33
6.2.16 In parameter list . . . 34
6.2.17 In parameter . . . 34
7 E-LOTOS semantics . . . 35
7.1 Overview . . . 35
7.2 Static Semantics . . . 36
7.2.1 Static semantic objects for Base . . . 36
7.2.2 Judgements on static semantics for Base . . . 37
7.2.3 Extended identifiers . . . 38
7.2.4 Static semantic objects for Modules . . . 39
7.2.5 Judgements on static semantics for Modules . . . 39
7.2.6 Cycle freedom . . . 40
7.2.7 Context morphism . . . 40
7.2.8 Realization . . . 40

Page

7.2.9 Interface Instantiation . . . 40
7.2.10 Interface Matching . . . 41
7.2.11 Renaming/Instantiation . . . 41
7.3 Untimed dynamic semantics . . . 44
7.3.1 Untimed dynamic semantic objects for Base . . . 44
7.3.2 Judgements on untimed dynamic semantics for Base . . . 45
7.3.3 Dynamic semantic objects for Modules . . . 46
7.3.4 Judgements on untimed dynamic semantics for Modules . . . 47
7.3.5 Environment morphism . . . 48
7.3.6 Signature Instantiation . . . 48
7.3.7 Renaming/Instantiation . . . 48
7.4 Timed dynamic semantics . . . 48
7.4.1 Judgements on timed dynamic semantics . . . 49
7.5 Write-many variables: the value substitution operator . . . 49
8 The E-LOTOS modules . . . 52
8.1 Specification . . . 52
8.1.1 Specification . . . 52
8.2 Top-level declaration . . . 53
8.2.1 Module not constrained by an interface . . . 53
8.2.2 Module constrained by an interface . . . 53
8.2.3 Generic module not constrained by an interface . . . 54
8.2.4 Generic module constrained by an interface . . . 55
8.2.5 Interface declaration . . . 55
8.2.6 Sequential top declaration . . . 56
8.3 Module body . . . 56
8.3.1 Block declaration . . . 56
8.3.2 Module Expression . . . 57
8.4 Module expression . . . 57
8.4.1 Module aliasing not constrained by an interface . . . 57
8.4.2 Module aliasing constrained by an interface . . . 58
8.4.3 Generic module actualization not constrained by an interface . . . 58
8.4.4 Generic module actualization constrained by an interface . . . 59
8.4.5 Generic module renaming/instantiation . . . 59
8.5 Module formal parameters . . . 60
8.5.1 Single . . . 60
8.5.2 Disjoint union . . . 60
8.6 Interface expressions . . . 61
8.6.1 Interface identifier . . . 61
8.6.2 Simple renaming . . . 61
8.6.3 Explicit body . . . 61
8.7 Interface body . . . 62
8.7.1 Type hiding the implementation . . . 62
8.7.2 Type synonym . . . 62
8.7.3 Constructed type . . . 62
8.7.4 Named record type . . . 63
8.7.5 Process declaration . . . 63
8.7.6 Equations . . . 64
8.7.7 Sequential declaration . . . 64
8.8 Record module expression . . . 64
8.8.1 Single . . . 64
8.8.2 Disjoint union . . . 65
8.8.3 Renaming tuple . . . 65

Page

8.9 Equation declarations . . . 65
8.9.1 Equations declaration . . . 65
8.9.2 Sequential . . . 66
8.10 Simple equation declaration . . . 66
8.11 Declarations . . . 66
8.11.1 Type synonym . . . 66
8.11.2 Type declaration . . . 66
8.11.3 Named record type . . . 67
8.11.4 Process declaration . . . 67
8.11.5 Sequential declarations . . . 68
9 The E-LOTOS base language . . . 69
9.1 Introduction . . . 69
9.2 Behaviours . . . 69
9.2.1 Disabling behaviour expression . . . 69
9.2.2 Synchronization behaviour expression . . . 70
9.2.3 Concurrency behaviour expression . . . 71
9.2.4 Selection behaviour expression . . . 73
9.2.5 Suspend/Resume behaviour expression . . . 73
9.2.6 Sequential composition . . . 74
9.2.7 Action . . . 75
9.2.8 Internal action . . . 76
9.2.9 Succesful termination without values . . . 76
9.2.10 Succesful termination . . . 76
9.2.11 Inaction . . . 77
9.2.12 Time block . . . 77
9.2.13 Delay . . . 77
9.2.14 Assignment . . . 78
9.2.15 Nondeterministic Assignment . . . 78
9.2.16 Choice over values . . . 79
9.2.17 Trap . . . 80
9.2.18 General parallel . . . 81
9.2.19 Parallel over values . . . 83
9.2.20 Variable declaration . . . 84
9.2.21 Gate hiding . . . 84
9.2.22 Renaming . . . 85
9.2.23 Process instantiation . . . 87
9.2.24 **loop** iteration . . . 87
9.2.25 Case . . . 88
9.2.26 Case with tuples . . . 88
9.2.27 Signalling . . . 89
9.3 Type expressions . . . 90
9.3.1 Type identifier . . . 90
9.3.2 Empty type . . . 90
9.3.3 Universal type . . . 90
9.3.4 Record type . . . 90
9.4 Record type expressions . . . 91
9.4.1 Singleton record . . . 91
9.4.2 Universal record . . . 91
9.4.3 Record disjoint union . . . 91
9.4.4 Empty record . . . 92
9.5 Value expressions . . . 92
9.5.1 Primitive constants . . . 92

Page

9.5.2 Variables . . . 92
9.5.3 Record values . . . 92
9.5.4 Constructor application . . . 93
9.6 Record value expressions . . . 93
9.6.1 Singleton record . . . 93
9.6.2 Record disjoint union . . . 93
9.6.3 Empty record . . . 93
9.7 Patterns . . . 94
9.7.1 Record pattern . . . 94
9.7.2 Wildcard . . . 94
9.7.3 Variable binding . . . 94
9.7.4 Expression pattern . . . 95
9.7.5 Constructor application . . . 95
9.7.6 Explicit typing . . . 96
9.8 Record patterns . . . 96
9.8.1 Singleton record pattern . . . 97
9.8.2 Record wildcard . . . 97
9.8.3 Record match . . . 97
9.8.4 Record disjoint union . . . 98
9.8.5 Empty record pattern . . . 98
9.9 Record of variables . . . 98
9.9.1 Singleton record variable . . . 98
9.9.2 Record disjoint union . . . 98
9.10 Behaviour pattern-matching . . . 99
9.10.1 Single match . . . 99
9.10.2 Multiple match . . . 100
10 Predefined library . . . 100
10.1 Booleans . . . 101
10.2 Natural Numbers . . . 103
10.3 Integral Numbers . . . 106
10.4 Rational Numbers . . . 109
10.5 Floating Point Numbers . . . 112
10.6 Characters . . . 113
10.7 Strings . . . 114
10.8 Enumerated Type Scheme . . . 114
10.9 Record Type Scheme . . . 116
10.10 Set Type Scheme . . . 117
10.11 List Type Scheme . . . 120
A Tutorial . . . 122
A.1 The base language . . . 122
A.1.1 Basic concepts . . . 124
A.2 The module language . . . 145
A.2.1 Basic concepts . . . 148
A.3 An E-LOTOS specification of the ODP trader . . . 156
A.3.1 Introduction . . . 156
A.3.2 An overview of the ODP Trader . . . 156
A.3.3 E-LOTOS Specification of the trader . . . 158
A.3.4 The complete specification . . . 161
B Guidelines for LOTOS to E-LOTOS translation . . . 179
B.1 Introduction . . . 179
B.1.1 Specification and process definition . . . 180
B.1.2 Basic LOTOS . . . 180

Page

B.1.3 Data Types 181
B.1.4 Full LOTOS 184

AUSTRALIAN/NEW ZEALAND STANDARD

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:

COPYRIGHT