

**Author:** Carlos Henrique Cabral Duarte (carlos.duarte@computer.org)

**Title:** *Proof-Theoretic Foundations for  
the Design of Extensible Software Systems*

**Language of presentation:** English

**Promotor:** Prof. Dr. Thomas S. E. Maibaum

**Date of defence:** 8th of February, 1999

**Institution:** Department of Computing  
Imperial College of Science, Technology and Medicine  
University of London

**Abstract:**

Extensible software systems have been increasingly demanded as a means of supporting in a more faithful way constantly changing user requirements and also as a necessary logical counterpart to rapidly evolving networking architectures. Such terms as open, reconfigurable, mobile and reflexive have been used to attempt to describe relevant facets of this kind of reactive system with dynamically varying functionality or structure. In this thesis, we not only characterise extensible systems but also study their rigorous design.

We advocate a proof-theoretic step-by-step approach to the development of extensible systems as a means of ensuring correctness, modularity and incrementability. By spelling out their characteristics and identifying corresponding logical constructions, we present as an original foundational contribution a first-order branching time logical system that seems to be appropriate as a basis for specification and verification. Even though our software process approach is proof-theoretic, we provide both model and proof theories for the proposed system, studying in the context of general logics important properties such as soundness, completeness and expressiveness. We argue that other logical systems proposed in the literature are not adequate to achieve the same desirable effects in design.

We also study particular software development approaches based on the actor model, on dynamic sub-classing and on meta-level architectures which could best underpin the rigorous design of extensible systems. Specific design principles are proposed in the form of derived inference rules with their application guidelines and composability notions are studied in terms of categories of theory presentations. We show that reasoning about their local properties can be carried out based only on such constructions but global properties may not be verified without the additional aid of a rely-guarantee discipline. A series of helpful theorems and realistic examples are developed to support and illustrate how our ideas can be effectively applied in practice.

## Table of contents:

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	What is Extensibility? . . . . .	2
1.1.1	A Classification of Software Changes . . . . .	2
1.1.2	Related Terminology . . . . .	4
1.1.3	Approaches to Support Extensibility . . . . .	4
1.2	Formal Design of Extensible Systems . . . . .	5
1.2.1	Process Calculi and Extensibility . . . . .	6
1.2.2	Temporal Logic and Extensibility . . . . .	8
1.3	Aims of the Thesis . . . . .	9
1.4	Outline of the Thesis . . . . .	9
<b>2</b>	<b>Proof Theory and Software Development</b>	<b>11</b>
2.1	The Proof-Theoretic Approach . . . . .	13
2.2	Logic in General . . . . .	19
2.3	Classical Propositional Logic . . . . .	30
2.4	Propositional Linear Time Logic . . . . .	40
2.5	Propositional Branching Time Logic . . . . .	47
2.6	Classical First-Order Logic . . . . .	54
2.6.1	Many-Sorted Logic with Equality . . . . .	58
2.7	First-Order Temporal Logic . . . . .	62
2.8	A Particular Model Theory . . . . .	72
2.9	Some General Logical Results . . . . .	77
2.10	Summary and Related Work . . . . .	91
<b>3</b>	<b>Designing Open Reconfigurable Systems</b>	<b>93</b>
3.1	Issues in the Design of a Proof Theory for the Actor Model . . . . .	95
3.2	An Axiomatisation of the Actor Model . . . . .	96
3.2.1	Representing Actors . . . . .	96
3.2.2	Axiomatising Actor Behaviours . . . . .	102
3.3	Verification of Local Properties . . . . .	108
3.4	Composition of Actor Specifications . . . . .	111
3.5	A Rely-Guarantee Design Discipline . . . . .	117
3.6	Verification of Global Properties . . . . .	121
3.7	A Plethora of Modes of Interaction . . . . .	130
3.8	Actors and Dynamic Subclassing . . . . .	136
3.9	Summary and Related Work . . . . .	140
<b>4</b>	<b>Reflection and the Design of Meta-Level Architectures</b>	<b>145</b>
4.1	Meta-level Considered Necessary: The Consensus Problem . . . . .	147
4.2	The Design of Meta-Level Architectures . . . . .	151
4.3	Computational Reflection . . . . .	155
4.4	Summary and Related Work . . . . .	156

<b>5</b>	<b>Case Study: Location Management for Mobility</b>	<b>157</b>
5.1	Location Management: Requirements . . . . .	158
5.2	Location Management in a Formal Setting . . . . .	159
5.3	Verifying Location Management Properties . . . . .	166
5.3.1	Location Space . . . . .	166
5.3.2	Location Service . . . . .	167
5.3.3	Other Properties of the Mobile Architecture . . . . .	169
5.4	Summary and Related Work . . . . .	169
<b>6</b>	<b>Concluding Remarks</b>	<b>171</b>
6.1	Contributions . . . . .	171
6.2	Further Work . . . . .	173
<b>I</b>	<b>Useful Theorems</b>	<b>175</b>
I.1	Classical Propositional Logic . . . . .	175
I.2	Propositional Linear Time Logic . . . . .	176
I.3	Propositional Branching Time Logic . . . . .	178
I.4	Classical First-Order Logic . . . . .	179
I.5	Many-Sorted Logic with Equality . . . . .	180
I.6	First-Order Temporal Logic . . . . .	180
<b>II</b>	<b>Remaining Cases in the Proof of Soundness</b>	<b>181</b>
	<b>Bibliography</b>	<b>185</b>
	<b>Notation Index</b>	<b>199</b>
	<b>Subject Index</b>	<b>200</b>

**Author address:** Banco Nacional de Desenvolvimento Econômico e Social  
 Av. Chile 100, Centro  
 Rio de Janeiro, RJ  
 20001-970 Brasil