A Substructural Modal Type Theory to handle Mobility Failures in Distributed Computing

Giuseppe Primiero

Department of Computer Science
Middlesex University
G.Primiero@mdx.ac.uk

Error detection and fault tolerance in software systems are well-studied notions since the 70’s in the context of engineered systems for resource sharing, resource typing and accessibility [9, 5, 22, 6, 7]. In these contexts, error-resolution consists in affected service identification and restart.

From the programming viewpoint, functional languages without side-effects limit the damage caused by failing computations and restarting them can be performed without multiple updates. A typical example of the standard strategy of re-evaluation of referentially transparent expressions to implement fault-tolerance is Erlang [3, 2].

Formal systems to study validity and satisfiability under failure conditions are limited in number and approaches. Semantically, Propositional Dynamic Error Logic, an extension of Propositional Dynamic Logic PDL [11], offers a labelled transition system including an error state, but it does not provide recovery means. Syntactically, typing systems are more common for implementations in distributed programming where failures can occur: type-checking and inference mechanisms for data sharing, [15]; systems for sound computations with explicitly distributed data structures, [14]; session types to mimic and check broadcasting with and without end-to-end reliable communication, [13, 10]; modal types to model mobile computing, [18]. In general, all failure mechanisms in type systems are expressed as forms of abortion procedures, reflecting the kill-and-restart strategy. In a typed natural deduction system, for example, the standard rule for falsehood (formally a negation introduction, corresponding to contradiction elimination) can be formulated as an instruction to abort a process, see e.g. [20]:

\[
\frac{\Delta; \Gamma \vdash m : \bot}{\Delta; \Gamma \vdash \text{ABORT}(m) : \alpha}
\]

This rule preserves correctness in a normalisation process, but it forbids any attempt at identifying and resolving the source of error.

Failures of mobility are an interesting and important sub-class of the several typologies of errors occurring in mobile and distributed computing. They do not seem to require necessarily abortion processes, as errors of typing do. Previous works on constructive modalities [4, 1] with applications to type theories [21, 19] and distributed systems [16, 8, 17, 18, 12] offer a possible basis for a syntactic investigation of failures in mobile distributed computing. We present a substructural modal type theory to reason about mobility failures and their resolutions. Our strategy is given in three steps:
1. we formulate a language with indexed processes for valid code and values;
2. we enrich the language with the means for mobility of code and values via modal-style functions;
3. we consider functions for mobility failures and their resolution, and for unresolvable errors.

The substructural nature of the language highlights some important aspects of failure-prone mobile computing:

– failure of commutativity indicates that resource allocation is relevant in view of mobility rules;
– failure of weakening indicates that (sufficiently complete) resource redundancy at locations guarantees availability of a resolution strategy;
– finally, failure of contraction shows that location redundancy is not trivial.

We present first a simple explanatory example, introduce rules for failures of mobility with appropriate handling procedures, and a standard abort rule. We formulate meta-theoretical properties of error expressions, show local soundness and completeness and prove termination of resolution or failure.

References


