# Distributed Algorithms and Causally Consistent Observations (Abstract)

Friedemann Mattern

Department of Computer Science, Technical University of Darmstadt,
Alexanderstr. 6, D 64283 Darmstadt, Germany
Email: mattern@isa.informatik.th-darmstadt.de

Observing an asynchronous distributed system which consists of processes that communicate solely by messages, is non-trivial – not only from a technical point of view (instrumentation, intrusiveness), but also because of inherent conceptual problems: Since event notification messages sent to an observer are subject to unknown delays, it is generally *not possible to observe all processes simultaneously*. However, if we simply deny the existence of global time, does it then still make sense to consider *global predicates* (i.e., predicates of the global state) of a distributed system? This is a serious question since such predicates may reflect important properties of a distributed computation. Examples include *deadlock*, objects being *garbage*, and whether the number of mobile agents of a certain type circulating in a system is greater than a given threshold $k$.

Fortunately, there exist several algorithmic means to guarantee that an observer gets at least a *causally consistent view* (i.e., a linearly ordered sequence of events with respect to the causality relation) of a distributed computation. These solutions can easily be generalized in such a way that not only a single observer, but each process within the system gets a causally consistent view of all events it learns about. A simple, although not very efficient solution consists in preventing direct or indirect message overtakings – either by generalizing the sequence number approach known from FIFO channel implementations, or by using a handshake communication scheme as in synchronous communications [4].

Such a realization of the so-called causal order message delivery property, however, does not solve all conceptual problems with global predicates: If two or more causally consistent observers monitor a single computation, they *may or may not agree* on the value of such a predicate – which, for example, makes the notion of global (or "distributed") breakpoints rather doubtful! Fortunately, there exist *observer independent predicates* (i.e., "objective facts"), forming a non-trivial class [1]. The well-known *stable predicates* (i.e., the "monotone facts") are a subset of these predicates.

Many (perhaps too many?) distributed algorithms to *detect* such stable predicates (i.e., to decide whether the predicate already holds) have been reported in the literature. A prominent example is *termination detection* [3], for which a surprising variety of algorithms with various characteristics have been published in recent years. A distributed computation is said to be terminated when all processes are passive and no message is in transit. However, since a passive process may be reactivated when a message is received, and since in general it is impossible to inspect all processes at the same time, the detection of termination of a distributed computation is non-trivial. Distributed termination detection is a "prototype problem"; research on it has contributed much to the entire field of distributed algorithms. It is closely connected to other important problems such as determining a causally consistent global state (the so-called *distributed snapshot problem* [2]) and *distributed garbage collection* (i.e., *identification* of objects which can no longer be reached) without freezing the underlying computation.

Termination detection and garbage collection are important from a practical as well as from a theoretical point of view. For example, distributed garbage collection algorithms are gaining much interest because of current efforts to efficiently implement object-oriented languages on parallel distributed memory machines and because distributed hypertext schemes (such as Wold Wide Web) become ubiquitous. Surprisingly, any garbage collection algorithm can be systematically transformed into a termination detection algorithm [5]. The transformation is rather straightforward and yields a deeper understanding on the relationship of these two prominent stable property detection problems and their underlying structure.

## References

1. Charron-Bost, B., Delporte-Gallet, C., Fauconnier, H., *Local and Temporal Predicates in Distributed Systems.* Technical Report, LITP, Université Paris 7, France, April 1992
2. Chandy, K.M., Lamport, L., *Distributed Snapshots: Determining Global States of Distributed Systems.* ACM Trans. on Computer Systems 3 (1), 1985, pp. 63-75
3. Mattern, F., *Algorithms for Distributed Termination Detection.* Distributed Computing 2, 1987, pp. 161-175
4. Mattern, F., Fünfrocken, S., *A Non–Blocking Lightweight Implementation of Causal Order Message Delivery.* Technical Report TR-VS-95-01, Department of Computer Science, Technical University of Darmstadt, January 1995
5. Tel, G., Mattern, F., *The Derivation of Distributed Termination Detection Algorithms from Garbage Collection Schemes.* ACM Trans. on Prog. Lang. Sys. 15 (1), 1993, pp. 1-35