STELLINGEN

Propositions belonging to the Ph.D. dissertation:

Actors at Work

by *Behrooz Nobakht*

1. Application-level priorities is one of the cross-functional requirements of concurrent systems as important as functional requirements. (Chapter 2)
2. Co-operative scheduling of application-level priorities is resourceful and costly in thread-based programming languages. (Chapter 2)
3. Real-time programming with deadlines and timeouts requires fine granularity and flexibility of the programming language. (Chapter 3)
4. Separation of invocation from execution of asynchronous messages drastically minimizes resourcefulness of co-operative scheduling in a multi-threaded runtime. (Chapter 4)
5. Java 8 features enable to create a more efficient abstraction for asynchronous messages. (Chapter 4)
6. Runtime verification of multi-thread Java applications through non-intrusive code annotations processing improves correctness and de-coupling in comparison to bytecode instrumentation. (Chapter 5)
7. Actors are a natural fit for a distributed monitoring model based on observation and reaction to guarantee composable multi-objective service characteristics. (Chapter 6)
8. Bridging modelling and programming for the purpose of verification and reasoning is a necessity for concurrent and distributed programming.
9. Explicit transfer of control in co-operative scheduling in ABS makes programs easier to understand and reason about compared to the implicit behavior in multi-threaded models.
10. Supporting hybrid thread mapping model in object-oriented languages such as Java is a desirable but currently unrealized feature because of implementation complexity.
11. The orthogonal properties of object orientation, co-operative scheduling, and actor model has given rise to numerous challenges to devise a new programming model at their intersection.
12. The increasing importance of concurrent and distributed programming, similar to the shift of structured programming to object orientation, requires more of mentality change rather than languages and tools.
13. Concurrency could be hard even for a computer despite the presence of an experienced programmer.