

Research Statement

Ionut Buricea

1 Brief Overview

In general terms, I am interested in automated verification and composition techniques for software systems and modules. Arguably the biggest problem that hampers the usability of verification methods in the case of concurrent software is the computational complexity caused by the state explosion phenomenon. Two of the major approaches to alleviate this problem are (1) reducing the interleavings of execution paths belonging to the various concurrent modules, and (2) improving the scalability of the analysis process by composition of already analysed modules.

My research has been centred around these two approaches, applied to communication protocols, with a strong emphasis on real-time protocols. Specifically, I have studied modular reasoning for timed and untimed protocols (represented as annotated finite state machines) with respect to safety and liveness properties. In related research work, I developed a reachability procedure for real-time concurrent modules which avoids executing all path interleavings of the system.

I am as much interested in investigating the theoretical foundation of different verification and design approaches as in actually building tools based on such approaches and evaluating and comparing their performances. For example, I have implemented my reachability procedure in Standard ML and measured its performance indicators on a wide array of test cases.

2 Research Work Description

- **Constraint-based Protocol Composition.** Protocol composition has been advocated as an attractive way to design and verify complex protocols, as it renders these processes more scalable. In [3], we present an approach to composing untimed protocols represented as Extended Finite State Machines.

In our approach, we consider the composition of protocols P and Q using constraint set SC . The elements in SC are constraints between events in P and Q respectively. Typical examples of constraints would be: “ a enables b ”, by which event a (say, from protocol P) allows event b (from protocol Q) to execute; “ a disables b ”, by which b cannot execute once a has. Then the composition of P and Q using SC consists of the set of all concurrent executions of the two components which obey each of the constraints in SC . We provide an implementation of this kind of composition which is based on guard variables. Such an approach allows a flexible and disciplined composition of protocol modules, and, just as important, can be performed in the exact same way the level of both protocol specification and the more compact level of service specification. In fact we show that service specification composition preserves the properties of the protocol specification composition. The implication is that the analysis of the composition at service specification level is enough to establish if a certain invariant or liveness property is satisfied by the composition at the protocol specification level. This results in efficient verification as the state space of the service specification is typically much smaller than that of the protocol specification.

We have extended this work for the case of **timed protocols**, represented as **Timed Extended Finite State Machines**. TEFSM are a subclass of Timed Automata in which there is one clock for each automaton, which is reset by every event execution. In [2] we present similar theoretical results to the ones in the untimed case: if a property holds for the composed service specification, then it is guaranteed to hold for the composed protocol specification (the converse is not true in the real-time case).

- **Reachability Analysis of Timed Protocols.** Pursuing the same goal of overcoming the verification complexity, we have developed a reachability algorithm for a network of timed extended finite state machines executing in parallel [1]. The approach works for discrete time and it skips some of the path interleavings by the use of a parallel-step technique, which generates a new analysis state by allowing independent events in different automata to be triggered independently, rather than a single event in the whole system at each step. The technique tends to be more efficient when the system of automata is “loosely coupled” (when the automata interact through relatively few constraints), and when events stay enabled for shorter periods of time. In the form presented in the dissertation, the algorithm is geared to detect *deadline violations* (situations where time cannot progress any more) in the system, but it can easily be adjusted for safety properties in general. I have written a full implementation of the reachability algorithm in Standard ML, and experimental data and comparison tables are available in [2].
- **Interleaving Reduction by Virtual Coarsening.** Finally, in another attempt to counter the effects of the state explosion phenomenon, we have developed a procedure ([2]) to compress timed extended finite state machines around the special transitions which are involved in composition constraints (i.e., interactions with other automata), for the same setting of a network of concurrent automata. Each automaton is individually processed, so that consecutive “internal” transitions (the ones not involved in composition constraints) are coalesced into a coarse transition which will be part the new automaton used in the verification. The resulting automata could end up having much more transitions than the original machines (they will typically have fewer states), but the coarsening helps the verification process in that a sequence of transitions in the original automaton will be passed in one step of the verification.

Used in conjunction, the graph compressing procedure and the reachability analysis can constitute an efficient methodology for the verification of real-time protocols. On the other hand, the graph compression can be used together with other verification techniques as well, and the basic procedure can be immediately adapted to any context that involves concurrent processes with specific points of interaction between one another.

3 Future Interests

In my future research I would like to further explore ways to improve design and verification techniques, especially in the field of real-time systems (in that respect I am planning to shift my research focus closer to the theory of timed automata and hybrid systems). A few of the near-term research goals derive directly from my current work:

- **Optimisation of the Virtual Coarsening Procedure.** The procedure we have developed can stand further improvements (for example some loops in the resulting automaton can be turned into single transitions with infinite time upper bound). I am looking to bring the procedure to an

optimal form, and be able to formally make a case for its optimality relative to the verification process.

- **Dynamic coarsening.** The coarsening procedure we have mentioned above is static in nature. Further coarsening might be possible dynamically while running the reachability algorithm. We would have to refine the notion of *locality* for transitions into a state-sensitive definition, much in the same way such refinements are applied to partial order techniques. Both static and dynamic coarsening could be used in the verification process; having applied the former would reduce opportunities for the latter, but not completely. Also, I hope to be able to adapt the static and dynamic techniques to a network of concurrent timed automata (rather than timed extended finite state machines), where the main difference to overcome will be that a timed automaton admits more than one clock.
- **Partial Order Techniques.** The technique of partial-order reduction is a well-established approach, which cuts down time and memory usage in the model-checking process by skipping unnecessary interleavings of independent transitions. It has been successfully applied to finite-state systems. In the case of timed systems, much less headway has been made. This is due to the fact that the advance of time by the same rate in all concurrent processes leaves little room for independence between transitions: different interleavings of concurrent transitions will produce different combinations of clock values in the subsequent global states. I believe that more work is called for in trying to break the implicit dependence that time effects on concurrent processes.

I am also interested in other aspects of verification and design, such as finding appropriate BDD-like data structures for efficient representation and comparison of explored states during the analysis of timed systems, component-based design and verification of concurrent systems, predicate abstraction as a means to enhance the effectiveness of reachability computation techniques for timed/hybrid systems and concurrent systems in general. I am equally interested in applying formal techniques in the areas of software concurrency, communication protocols, hardware verification and embedded systems.

In addition, I am always excited to get involved in the actual implementation of formal techniques and methods. I like developing end-to-end solutions and I would enjoy opportunities to apply and assess them against real-life systems.

References

- [1] I. Buricea and G. Singh. Compositional design of real-time protocols. In *International Conference on Parallel and Distributed Computing Systems*, Nov. 2000.
- [2] Ionut Buricea. *Analysing Composed Real-time Protocols*. PhD Thesis, Kansas State University, 2003.
- [3] G. Singh, Z. Mao, and I. Buricea. Composition of service specifications. In *IEEE International Conference on Network Protocols*, 1998.