The Sigali Tool Box

Hervé Marchand
IRISA/INRIA Rennes
Campus Univ. de Beaulieu, 35042 Rennes,France
Herve.Marchand@irisa.fr

11th April 2005

1 Overview of the tool

SIGALI is a model-checking tool-based which manipulates *ILTS: Implicit Labeled Transition Systems* (which can be seen as an equational representation of an automaton) as intermediate models for discrete event systems. It offers functionalities for verification of reactive systems and discrete controller synthesis.

The techniques used consist in manipulating the system of equations instead of the sets of solution, which avoids the enumeration of the state space. Each set of states is uniquely characterized by a polynomial and the operations on sets can be equivalently performed on the associated polynomials. Therefore, a wide spectre of properties, such as liveness, invariance, reachability and attractivity can be checked or ensured. Many algorithms for computing predicates states are also available.

2 The Controller Synthesis methodology

Control theory of discrete event systems allows to use constructive methods, that ensure, a priori, required properties on the system behavior. In this approach, the validation phase is reduced to properties not guaranteed by the programming process.

Starting from a representation of the possible behaviors of the system(e.g. in the form of a finite state automaton) and the properties that have to be satisfied by the controlled system, the synthesis produces directly the constrained automaton, i.e., the one that presents only those behaviors that satisfy the required properties. In our framework, The system is represented by an ILTS while the control of the system is performed by restricting the controllable input values to values suitable for the control goal. This restriction is obtained by incorporating new algebraic equations into the initial system.

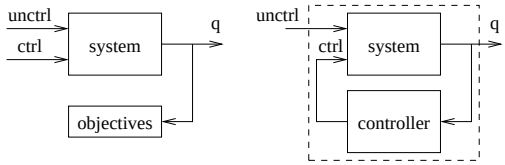

Figure 1: Discrete control synthesis: from uncontrolled system (left) to closed-loop (right).

Using symbolic methods [3] (based on BDD techniques, avoiding state space enumeration), the various control objectives for which we are able to synthesize a controller are the following:

- "Traditional" control objectives such that:
  - the *reachability* of a set of states from the initial states of the system,
  - the *attractivity* of a set of states E from a set of states F.
  - the *persistence* of a set of states E.
  - the *reccurence* of a set of states E.
- Control objectives expressed as partial order relations over the states of the system such that:
  - the *minimally restrictive control* (choice of a command such that the system evolves, at the next instant, into a state where the maximum number of uncontrollable events is admissible)
  - the *stabilization of a system* (choice of a command such that the system evolves, at the next instant, into a state with minimal change for the state variable values)
  - the *optimal control* (minimization of the cost of the trajectories between a set of initial states and a set of final states)

Note that if the property is expressed by means of an observer ω, with a sink state $Error$, then it is sufficient to perform the synchronous product between the ILTS modeling the plant and the observer and to compute a controller that avoids states of the form $(q, Error)$ to be reachable in this new system.

3 Tools implementing the models and synthesis

The current implementation of the method relies on the chain of Figure 2. Centrally, we use SIGALI [2], which is a tool that performs model-checking, controller synthesis for logical goals, and optimal controller synthesis.

The model of the system can be described using a synchronous formalism. The equational language SIGNAL is the synchronous language originally connected with the synthesis tool SIGALI [2]. Another such formalism is Mode Automata [1], for which the tool MATOU provides for compilation and has been adapted to generate the input format of *Sigali*.

The result of the synthesis in SIGALI is a controller, in the form of a logical relation, which can be interpreted by a resolver module: for a given state and uncontrollable input, the constraints on controllable signals are solved, for example in an incremental, interactive way following the manual valuation of signals. The resolver can be coupled with the original specification of the uncontrolled system, using either the Polychrony environment, or the tool *SigalSimu* in the case of Mode Automata.

SIGALI is freely available for non-commercial use on the web page [8]. Note that you will need part of the Polychrony [8] or Matou [7] environment as front-end in order to specify your systems.

Some academic examples are explained in[2]. In [6], the synthesis methodology have been applied to the incremental design of a power transformer station controller, whereas in [5, 4, 9], we have been interested in the automatic control of systems with multiple tasks, each with multiple modes, implementing a functionality with different levels of quality (e.g., computation approximation), and cost (e.g., computation time, energy) and we made the use of SIGALI in order to control the switching of modes in order to insure properties like bounding cost while maximizing quality.

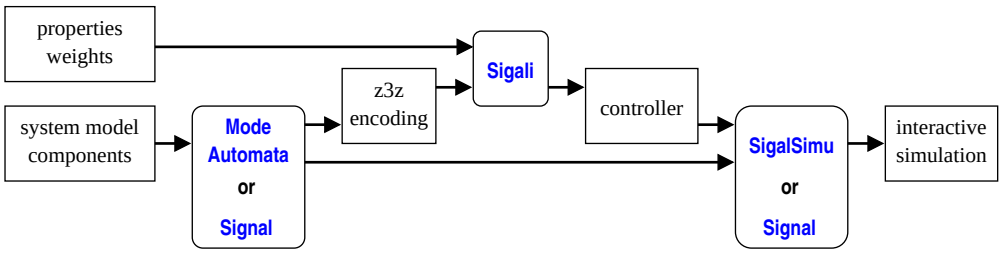

Figure 2: Implementation of the approach: the tools involved.

References

[1] F. Maraninchi and Y. Rémond. Mode-automata: a new domain-specific construct for the development of safe critical systems. *Science of Computer Programming*, 46(3):219–254, 2003.

[2] H. Marchand, P. Bournai, M. Le Borgne, and P. Le Guernic. Synthesis of discrete-event controllers based on the signal environment. *Discrete Event Dynamic System : Theory and Applications*, 10(4):347–368, October 2000.

[3] H. Marchand and M. Le Borgne. The supervisory control problem of discrete event systems using polynomial methods. Research Report 1271, Irisa, October 1999.

[4] H. Marchand and E. Rutten. A case study in applying discrete control synthesis to excavator operation. In *IEEE International Conference on Systems, Man and Cybernetics (IEEE SMC)*, Hammamet, Tunisia, October 2002.

[5] H. Marchand and E. Rutten. Managing multi-mode tasks with time cost and quality levels using optimal discrete control synthesis. In *14th Euromicro Conference on Real-Time Systems (ECRTS'02)*, June 2002.

[6] H. Marchand and M. Samaan. Incremental design of a power transformer station controller using controller synthesis methodology. *IEEE Transaction on Software Engineering*, 26(8):729–741, August 2000.

[7] Matou. www-verimag.imag.fr/people/ /florence.maraninchi/matou/.

[8] Polychrony. www.irisa.fr/espresso/polychrony/.

[9] E. Rutten and H. Marchand. Automatic generation of safe handlers for multi-task systems. Technical Report 5345, INRIA, October 2004.