Temporal Golog with Execution Monitoring

Mikhail Soutchanski
Department of Computer Science
University of Toronto
http://www.cs.toronto.edu/~mes/index.html

Imagine a robot that is executing a program on-line, and, insofar as it is reasonable to do so, it wishes to continue with this on-line program execution, no matter what exogenous events occur in the world. *Execution monitoring* is the robot's process of observing the world for discrepancies between the actual world and its internal logical representation of it, and recovering from such discrepancies.

We provide logical specifications of on-line program executions (with monitoring) formulated in a version of the sequential situation calculus that includes time (Reiter 1998). In that version, all primitive actions have an additional temporal argument. Our account relies on specification of a single-step interpreter for the logic programming language Golog (De Giacomo, Lespérance, & Levesque 1997). The main contribution of this paper is an adaptation of the framework developed in (De Giacomo, Reiter, & Soutchanski 1998) to the temporal domain. Our interpreter for sequential temporal Golog computes a schedule (which is a solution of temporal constraints) for a remaining part of a Golog program whenever it makes a single step of execution. In reality, some unexpected exogenous actions may delay execution of robot's actions. For this reason, before performing an action A that was scheduled for execution at time T_1, robot senses its internal clock to determine the current time T_2. If $T_2 \leq T_1$ then robot waits until T_1 and then performs A. If $T_1 < T_2$, but there is an alternative solution of the system of temporal constraints such that a rescheduled remainder of a Golog program can be successfully completed, then the execution monitor reschedules the rest of the Golog program (in particular, it assigns new execution time $T_3 \geq T_2$ to the action A), and (after waiting for some time, if necessary) performs A at the time T_3. Otherwise, if there is no alternative schedule, then either it is too late to perform the action A selected by the single-step interpreter, or one of the subsequent actions will miss its deadline. However, if the action A was selected as one of possible alternatives of a nondeterministic choice, and there is another possible action B that can be scheduled at this moment and there is a schedule for all remaining actions, then the execution monitor abandons A (and actions that follow after it) in favor of B and subsequent actions. This loop of parsing a Golog program, sensing, monitoring and executing repeats until the Golog program finishes or until it fails if no continuing schedule can be found.

The theory is supported by an implementation that is illustrated by monitoring of Reiter's coffee delivery program (Reiter 1998). The interpreter for temporal Golog based on transitional semantics and an execution monitor have been implemented in Eclipse Prolog. Our execution monitor is able to control an arbitrary Golog program in which primitive actions depend on time. The interpreter and monitor were tested in the simulation mode using the coffee delivery program. Ongoing work along these lines includes controlling an RWI B21 autonomous robot to perform temporal scheduling tasks in an office environment. In this setting, it would be unrealistic to expect the robot to execute a schedule. Frequently, it will be impossible to meet the exact times in such a schedule, for example, if the robot is unexpectedly delayed in traveling to the coffee machine (e.g., by people walking in a corridor).

A paper providing a logical account, connections with related research and the description of our implementation will be available from the WWW home page mentioned above.

References

De Giacomo, G.; Lespérance, Y.; and Levesque, H. 1997. Reasoning about concurrent executions, prioritized interrupts, and exogenous actions in the situation calculus. In *15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, volume 2, 1221–1226. Nagoya, Japan: Morgan Kaufmann.

De Giacomo, G.; Reiter, R.; and Soutchanski, M. 1998. Execution monitoring of high-level robot programs. In *Principles of Knowledge Representation and Reasoning: Proceedings of the 6th International Conference (KR'98)*, 453–464.

Reiter, R. 1998. Sequential, temporal golog. In *Principles of Knowledge Representation and Reasoning: Proceedings of the 6th International Conference (KR'98)*, 547–556.