An Automaton-based Formalism for Cooperative Augmented Reality Systems

Felix G. Hamza-Lup[1] Ferucio Laurenţiu Ţiplea[2]

[1]Department of Computer Science, Armstrong Atlantic State University, Savannah, Georgia 31419-1997, USA

[2]Department of Computer Science, "Al.I.Cuza" University, Iasi, Romania

NCMA 2009

Outline

Augmented Reality Systems

- The basic idea of augmented reality is to superimpose graphics, audio and other sense enhancements over a real-world environment in real-time, and to change them to accommodate a user's head- and eye-movements, so that the graphics always fit the perspective;
- Augmented reality is still in an early stage of research and development at various universities and high-tech companies;
- Three basic components needed to make an augmented-reality system work:
  - head-mounted display;
  - tracking system;
  - mobile computing power.

Augmented Reality Systems

There are hundreds of potential applications for augmented reality, such as:

- medicine;
- maintenance and construction;
- military;
- gaming;
- instant information

More details about AR systems and related projects:

http://cs.armstrong.edu/felix/

Example of Cooperative Augmented Reality System:

Endo-tracheal Intubation (ETI)

Figure: Instructors visualizing the 3D models relative position (left), while a remote student performs the ETI procedure (right)

Example of Cooperative Augmented Reality System:

Remote Telerobotic Manipulation

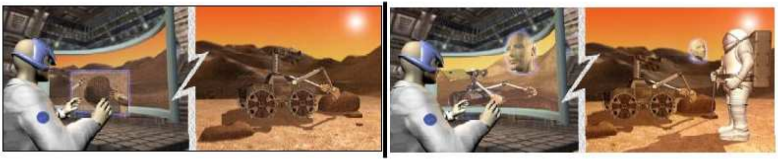


Figure: Multi-Modal Interaction System

Modeling CARSs

- Actors — entities that are able to perform complex operations on a given set of variables.
  $\mathcal{A} = \{A_1, \ldots, A_k\}$ is a given set of actors;

- Objectives — sequence of actions that actors are to perform in order to drive the system from its initial state γ_0 to some final state γ_f.
  An o-state (observation state) is a valuation γ of a given set $\mathcal{V} = \{x_1, \ldots, x_m\}$ of (typed) variables.
  Γ is the set of all o-states.

- Environments and actions —
  - $read_A : Q \to T$ is the read-time function of A;
  - $write_A : Q \to T$ is the write-time function of A;

- Modeling actors — An *actor* is a 4-tuple $A = (Q, \Sigma, \delta, q_0)$, where

$$\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q \times \Sigma)$$

(δ may be a partial function).
Infinite input sets are allowed;

- Time constraints — A time-constraint is any function

$$\mathcal{C} : \Gamma \rightarrow T \cup \{\infty\}$$

which gives the maximum delay permitted to the actors to trigger their actions in a state γ.
$\mathcal{C}(\gamma) = \infty$ means that no time-constraint is imposed.

Modeling CARSs

- Cooperative system (CS) — is a 5-tuple

$$\mathcal{S} = (\mathcal{V}, \mathcal{A}, read_{\mathcal{A}}, write_{\mathcal{A}}, \mathcal{C})$$

- Computation — transition relation between configurations

$$(t, q_1^1, \ldots, q_1^k, \gamma) \vdash (t', q_2^1, \ldots, q_2^k, \gamma')$$

iff there exists an i such that:

1. $read_{A_i}(q_1^i) \leq \mathcal{C}(\gamma)$ (i.e., A_i satisfies the time-constraint $\mathcal{C}(\gamma)$);
2. A_i performs an action, i.e.
   1. $\delta_i(q_1^i, \gamma) = (q_2^i, \gamma')$;
   2. $t' = t + read_{A_i}(q_1^i) + write_{A_i}(q_1^i)$;
3. $q_2^j = q_1^j$, for all $j \neq i$ (i.e., the other actors do not perform any action).

- Objectives again — variations of the reachability problem:
  - **Reachability Problem**

    Instance: cooperative system $\mathcal{S}$, initial o-state γ_0, and final o-state γ_f;

    Question: is γ_f reachable from γ_0?

  - **P-Reachability Problem**

    Instance: cooperative system $\mathcal{S}$, initial o-state γ_0, final o-state γ_f, and predicate P over Γ;

    Question: is γ_f P-reachable from γ_0?

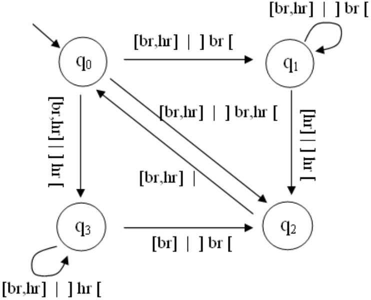
  - **Time-reachability Problem**

    Instance: cooperative system $\mathcal{S}$, initial o-state γ_0, final o-state γ_f, predicate P over Γ, and time value t;

    Question: is γ_f P-reachable from γ_0 in time $t' \leq t$?

Modeling CARSs

Example of an actor in ETI:



$$read_{A_2}(q) = write_{A_2}(q) = \tau, \ \forall q$$

q_0	–	br and hr normal
q_1	–	br at least once modified
q_2	–	br and hr at least once modified
q_3	–	hr at least once modified

Figure: Actor A_2

From PN to CS:

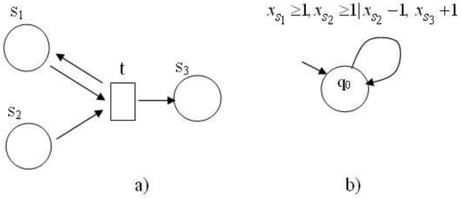


Figure: a) A transition t; b) The actor A_t

$$M[t\rangle M' \;\Leftrightarrow\; (0, q_0, \ldots, q_0, \gamma_M) \overset{A_t}{\vdash} (0, q_0, \ldots, q_0, \gamma_{M'})$$

From CS to PN:

- A cooperative system $\mathcal{S}$ is called monotonic if:
  - for any variable x, its domain is **N**;
  - for any actor A and any transition $(q', \gamma') \in \delta(q, \gamma)$ of A, the following property holds true

$$(q', \bar{\gamma} + (\gamma' - \gamma)) \in \delta(q, \bar{\gamma}),$$

    for any $\bar{\gamma} \geq \gamma$ (the inequality between functions is component-wise defined).

- A monotonic cooperative system $\mathcal{S}$ is called locally finite if for any actor A and any states q and q' of A, there exists a finite set of vectors with integer components, $\{V_1, \ldots, V_p\}$, such that for any transition $(q', \gamma') \in \delta(q, \gamma)$ of A there exists i with $\gamma' - \gamma = V_i$.

Theorem 1

For any monotonic and locally finite cooperative system $\mathcal{S}$ without time-constraints, there exists a Petri net Σ such that for any configurations c and c' of $\mathcal{S}$ there are two markings M_c and $M_{c'}$ and a transition $t_{c,c'}$ satisfying

$$c \vdash c' \quad \Leftrightarrow \quad M_c[t_{c,c'}\rangle M_{c'}.$$

From CS to PN (example):

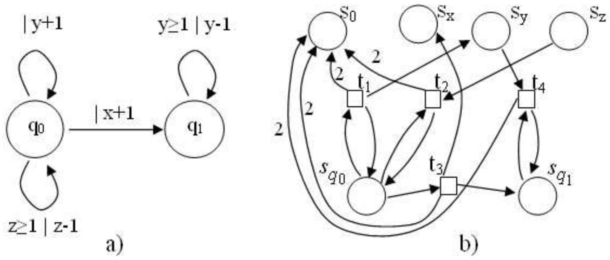


Figure: a) A CS with only one actor A; b) A Petri net associated to the CS in a)

Reachability Problem for Cooperative Systems

Theorem 2

The reachability problem for cooperative systems is undecidable.

Proof.

The halting problem for counter machines can be reduced to the reachability problem for CS. □

Theorem 3

The polynomial time-reachability problem for finite-domain cooperative systems is NP-complete.

Proof.

Membership to NP ($\mathcal{S}$, γ_0, γ_f, predicate P verifiable in polynomial time, and time value t of polynomial size (w.r.t. $||\mathcal{S}||$)):

`guess` a sequence of transitions of length at most t such that the first one rewrites γ_0 and the last one ends up with γ_f;
`if` the sequence induces a computation
`then` `if` each configuration in computation verifies P
`then` "yes" `else` "no";

NP-hardness: reduction from the Hamiltonian circuit problem. □

Conclusions

- This work proposes an automaton-based formalism for CARSs;
- Future work:
  - in-depth study of the basic properties of the model;
  - verification techniques (based on automata theory (reachability-based techniques, model checking etc.));
  - accommodate delays in the formalism.