# An Automata-based Formalism for Cooperative Augmented Reality Systems


Felix Hamza- Lup
School of Computing
Armstrong Atlantic State University
Savannah, GA, USA
felix@cs.armstrong.edu

Ferucio Laurenţiu Ţiplea *
University of Central Florida
School of Computer Science
Orlando, Florida 32816, USA,
tiplea@cs.ucf.edu



## Abstract

*The aim of the paper is to propose a formal model for Cooperative Augmented Reality Systems (CARSs). We motivate it by two examples, one of them applying Augmented Reality (AR) paradigms to medical training, and the other one to telerobotic manipulation. The model is based on automata theory and objectives are formulated as reachability-like decision problems. We show that reachability, which plays an important role in analyzing CARSs, is undecidable in general, but it is NP-complete for finite-domain CARSs. The relationship with Petri nets, as models of distributed and concurrent systems, is also provided.*


## 1 Introduction

Augmented Reality (AR) systems [11] use computers and specific visualization devices to overlay virtual information in the real world. They enhance the perception of, and the interaction with the real world. Visually, the real scene a person sees is augmented with computer-generated objects. These virtual objects are placed (*registered*) in the real scene in such a way that the information they carry appears in the correct location with respect to the real objects they augment.

Several AR systems were proposed in the mid '90s as tools to assist in different fields such as medicine [5], complex assembly labeling [4], and construction labeling [18]. With advances in computer graphics, networking, and hardware (i.e., 3D displays, haptic devices etc.) the research community has shifted attention to distributed environments that use extensively the AR paradigm [1, 14]. Furthermore, a Cooperative Augmented Reality System (CARS) can substantially facilitate experts' interactions, especially during quick-response conditions such as medical emergencies [15], and has the potential to provide efficient training. An important aspect regarding these systems is that augmentation can occur for multiple sensory modalities (haptic, visual, auditive).

The main challenge encountered in designing CARSs is the dynamic nature of the environment. The attributes of the virtual components of the scene are changing as an effect of the participants' interactions. These interactions and information exchanges generate a state referred to as the *dynamic shared state* [16] that has to be maintained consistent at all sites for all participants and in the presence of inevitable network latency and jitter. In spite of the fact that several problems inherent in such environments have been investigated over the years, no generally accepted formalism allowing an in-depth reasoning for such systems has been developed.

The aim of this paper is to propose a formal model for CARSs. We motivate it by two examples, one of them applying AR paradigms to medical training, and the other one to telerobotic manipulation. The model is based on automata theory and objectives are formulated as reachability-like decision problems. We show that reachability, which plays an important role in analyzing CARSs, is undecidable in general, but is NP-complete for finite-domain CARSs. A relationship with Petri nets, as models of distributed and concurrent systems, is also provided.

The paper is organized into six sections. Section 2 describes two complex CARSs as motivating examples for this work. In Section 3 we raise the abstraction level by introducing the main components of a formal model intended to capture the behavior of CARS. In Section 4, the model is applied to the first CARS example, an AR-based Endotracheal Intubation training system. A few basic properties of our model are studied in Section 5. We end the paper with conclusions followed by near future work.

*On leave from "Al.I.Cuza" University of Iaşi, Department of Computer Science, Iaşi, Romania, e-mail: `fltiplea@mail.dntis.ro`

## 2 Examples of Cooperative Augmented Reality Systems (CARSs)

The complexity of modeling and reasoning about systems that involve cooperation between tasks and data distribution plays an important role in CARSs development.

In the following paragraphs, two examples of CARSs are briefly discussed. These examples will be used to motivate our formal model and to exemplify the main problems one encounters when designing and planning the development of such complex systems.

### 2.1 AR Systems for Training Endo-Tracheal Intubation

Endo-tracheal Intubation (ETI) is a frequent medical procedure encountered in Emergency Rooms (ER). For a successful intubation, a trained physician must insert an endotracheal tube through the patient's mouth or nose into the trachea to assure lungs ventilation. The most important reason for training clinicians in ETI is the inherent difficulty associated with the procedure. In case of severe trauma patients, emergency airway management is classified as a major cause of pre-hospital death trauma by the American Heart Association [17]. Many anesthesiologist believe that the main cause of failure in applying the ETI is the difficulty for the clinician to visualize the vocal cords [3].

A common training methodology is based on the Human Patient Simulator (HPS) [8] a plastic mannequin, used as a severe trauma victim. A medical student in a practice ER is responsible for stabilizing the simulated patient. One of the first procedures the student must perform is the ETI. The standard training procedure is executed locally (i.e., both the student and his/her's instructor are at the same location).

Augmenting the real environment with virtual 3D models that participants may interact with remotely, has the potential to enhance training effectiveness allowing participants to visualize and better understand various complex medical procedures with frequent exposure to the techniques and without the cost of traveling. In [6] the ETI procedure based on the HPS system was enhanced using AR paradigms (i.e., virtual models superimposed dynamically on the HPS), allowing the student to visualize the internal anatomy (i.e., trachea and lungs). During the intubation procedure the instructors located remotely visually assess the student skills based on the relative position of the virtual models displayed and the associated simulation parameters [9].

Using the above CARS, a student in a practice ER can perform the ETI while two or more instructors located remotely visualize, interact during the training procedure, and trigger difficult intubation scenarios (e.g. blocked airway and halo cervical traction) by changing the simulations parameters (e.g., breathing rate, hearth rate) as illustrated in Figure 1.

**Figure 1. Instructors visualizing the 3D models relative position (left), while a remote student performs the ETI procedure (right)**

Using this training CARS, the student has the ability to visualize the changes in the HPS behavior (e.g., blocked airway) and the associated virtual models (relative position of the virtual endotracheal tube and virtual trachea).Based on this visual feedback the student will adopt different procedures on the simulator in order to correctly accomplish the intubation.

Enhancement of the HPS with cooperative AR capability has the potential to:

- Simultaneously train local and remotely located students;
- Allow students to actually "see" and therefore better understand their actions on the HPS which affects the behavior of the simulator;
- Allow an instructor to change the simulation parameters and confront students with different emergency scenarios (e.g. blocked airway, different ventilatory patterns).

This CARS example will be formally modeled and detailed in Section 4.

### 2.2 AR Systems for Remote Telerobotic Manipulation

Another example that would take advantage of our formal reasoning model which will be proposed in the next section is an AR Remote Telerobotic Manipulation system []. Although this system will not be discussed in detail it is interesting to emphasize the potential and the wide range of collaborative applications that can benefit from the formal model.

With the advances in multi-modal interaction devices, integrated multi-modal interfaces can enable a team to visually and haptically (e.g., force feedback) sense the environment of a robot and to remotely control a robot's navigation/manipulation in hazardous environments. Such a cooperative system will extend the human visual and haptic telepresence to near and far space via human-robotic systems and by interconnecting the visual senses and decision making of remote participants [2].

Systems for remote surface exploration or orbit assembly and repair tasks and maneuvers are under continuous investigation by the National Space Agency (NASA) []. Such systems help "extend human presence throughout the solar system" by extending human telepresence to near and far space via robotic systems and across astronauts during surface exploration or during station repair.

In what follows we are briefly presenting the functionality of such a prototype based on the following real scenario. During a space mission repairs have to be executed on the International Space Station located on the Earth's orbit. An astronaut located in the Space Shuttle attached to the station uses the system to manipulate a robotic arm during repairs. At the same time he receives repair instructions/protocols from the Earth's surface base station through an AR-based system (Figure 2).

**Figure 2. Multi-Modal Interaction System**

The system has the potential to:

- Provide haptic feedback to the user regarding the force applied by the robotic arm while repair actions are executed (e.g., rotating a valve until a certain force threshold has been reached);
- Allows the user and the team of engineers located on the land station to actually "see" and therefore better understand their actions on the component being repaired.
- Allows the user and the team of engineers located on the land to communicate in real-time on their actions.

## 3 Modeling CARSs

In this section we propose an automata-based formalism for CARSs and formulate objectives as reachability-like decision problems in the formalism.

**Actors** *Actors* are entities that are able to perform complex operations on a given set of variables. These operations are specified with respect to a concrete application. In what follows we assume that a set $\mathcal{A} = \{A_1, \dots, A_k\}$ of $k \geq 1$ actors is given.

**Objectives** Let $\mathcal{V} = \{x_1, \dots, x_m\}$ be a set of (typed) variables, each of which has associated a *type* $\tau_i$ and a *domain* $D_{\tau_i}$. In our approach, each domain is at most countable. Because of this, the notation $[a, b]$ will be mostly used to denote finite intervals (e.g., $[1, 3]$ may denote the set $\{1, 2, 3\}$ or the set $\{1, 1.5, 2, 2.5, 3\}$; the distinction will be clear from the context).

An *observation state over* $\mathcal{V}$ (*o-state*, for short) is any assignment $\gamma : \mathcal{V} \rightarrow \bigcup_\tau D_\tau$ such that $\gamma(x) \in D_\tau$, for any type $\tau$ and variable $x$ of type $\tau$. $\Gamma(\mathcal{V})$, or simply $\Gamma$, will stand for the set of all o-states over $\mathcal{V}$.

o-states represent discrete observations of the behavior of a given system. Actors may interact with the system and guide its behavior. Therefore, o-states are controllable up to some extent. Given an *initial o-state* $\gamma_0$ and a *final o-state* $\gamma_f$, an *objective* can be roughly defined as a sequence of actions that actors are to perform in order for the system to reach $\gamma_f$ from $\gamma_0$.

**Environments and Actions** Each actor acts in some *environment* and performs some actions. Each action performed by an actor $A$ presupposes:

- a time $\tau$ required by $A$ to read the current o-state $\gamma$;
- a time $\tau'$ required by $A$ to perform an action.

The time values $\tau$ and $\tau'$ depend on the environment in which the actors act. For example, a satellite orbiting Earth reads the current state (from an Earth base) faster than a satellite orbiting March (Figure 3 illustrates this).

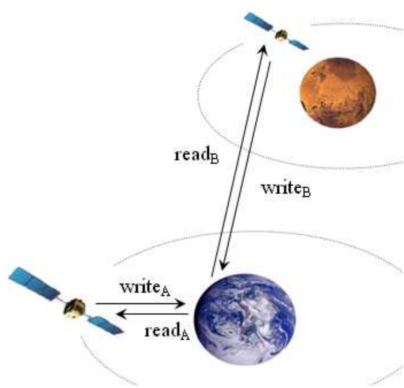


**Figure 3. Read- and write-times associated with different positions in space**

If a set $Q$ of states is associated with an actor $A$, then the time values needed by $A$ to read the current o-state and to perform an action can be given by two functions, the *read-time function* $read_A : Q \rightarrow T$ and the *write-time function* $write_A : Q \rightarrow T$, where $T$ is a set of time values (each of which is a non-negative real number).

Of course, many variations of these two functions can be defined, depending on the system we want to model:

1. $\tau$ is constant for each actor and each action;
2. $\tau$ depends on actors;
3. $\tau$ depends on actors and actions (an actor may not need to read the entire state in order to be able to perform a required action).
4. similar discussion on $\tau'$.

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

- $Q$ is a finite non-empty set of *states*;
- $\Sigma$ is a set of *inputs*;
- $\delta$ is the *transition function*, $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q \times \Sigma)$, which may be a partial function ($\mathcal{P}(X)$ stands for the powerset of $X$);
- $q_0 \in Q$ is the *initial state*.

As we can see, an actor is a kind of a non-deterministic automaton, the only difference consisting in the fact that infinite input sets are also allowed.

The *size of the actor* $A$, denoted $||A||$, is the number of transitions of $A$.

In most cases, $\Sigma$ will be the set $\Gamma(X)$ of all o-states over some subset $X \subseteq \mathcal{V}$ of variables. In such a case, the actor $A$ will be called *local* if $X \subset \mathcal{V}$, and *global* if $X = \mathcal{V}$. A local actor has access only to a proper subset of variables. Moreover, to have a flexible notation, especially with computations which are going to be defined in what follows, we will extend the transition function of local actors from $\Gamma(X)$ to $\Gamma(\mathcal{V})$ by:

$$(q', \gamma') \in \delta(q, \gamma)$$

whenever there exists $(q', \theta') \in \delta(q, \theta)$ such that $\gamma(x) = \theta(x)$ and $\gamma'(x) = \theta'(x)$ for all $x \in X$, and $\gamma(y) = \gamma'(y)$ for all $y \in \mathcal{V} - X$.

We emphasize that such an extension is just for technical purposes (see below the definition of a computation).

**Time-constraints** If an actor $A$ is in a state $q$ and it is not able to read the current o-state in due time, then no action specific to $q$ can be triggered by $A$.

*Time-constraints* impose time restrictions on triggering actions. A time-constraint is any function

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

$\mathcal{C}(\gamma)$ gives the maximum delay permitted to the actors to trigger their actions in state $\gamma$. When $\mathcal{C}(\gamma) = \infty$, we will say that no time-constraint is imposed.

**Cooperative Systems** Now, we introduce our model, called *cooperative system* (CS). Such a system is defined as a 5-tuple

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

where

- $\mathcal{V}$ is a set of (typed) variables;
- $\mathcal{A}$ is a set of actors over $\mathcal{V}$ (i.e., their inputs are o-states over subsets of variables);
- $read_{\mathcal{A}} = \{read_A | A \in \mathcal{A}\}$ is a set of read-time functions;
- $write_{\mathcal{A}} = \{write_A | A \in \mathcal{A}\}$ is a set of write-time functions;
- $\mathcal{C}$ is a time-constraint.

If only finite domains are associated to variables, then we will say that $\mathcal{S}$ is a *finite-domain CS*, and if all actors are local, then $\mathcal{S}$ will be called a *local CS*.

The *size* of a CS $\mathcal{S}$ with $k$ actors is

$$||\mathcal{S}|| = \sum_{i=1}^{k} ||A_i||,$$

where $A_i$ is the $i$th actor of $\mathcal{S}$.

**Computations** A *configuration* of a cooperative system $\mathcal{S}$ is any $(k+2)$-tuple

$$(t, q_1^1, \ldots, q_1^k, \gamma),$$

where:

- $t$ is the current time;
- $q_1^i$ is the current state of $A_i$, for any $i$;
- $\gamma$ is the current o-state.

The *transition relation* $\vdash$ is given by:

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

iff there exists $i$ 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.
   (a) $\delta_i(q_1^i, \gamma) = (q_2^i, \gamma')$;
   (b) $t' = t + read_{A_i}(q_1^i) + write_{A_i}(q_1^i)$;
3. $q_2^j = q_1^j$, for all $j \neq i$ (i.e., the other actors do not perform any action).

As usual, $\overset{+}{\vdash}$ is the transitive closure of $\vdash$, and $\overset{*}{\vdash}$ is the reflexive and transitive closure of $\vdash$.

A computation of $\mathcal{S}$ usually starts with the initial configuration

$$(0, q_0^1, \ldots, q_0^k, \gamma_0),$$

where $\gamma_0$ is the initial o-state and $q_0^i$ is the initial state of $A_i$, for any $i$. If

$$(0, q_0^1, \ldots, q_0^k, \gamma_0) \overset{*}{\vdash} (t, q^1, \ldots, q^k, \gamma)$$

then we will say that $\gamma$ *is reachable in time* $t$.

**Objectives Again** Objectives with respect to a cooperative system can be often defined as variations of the reachability problem which asks to decide whether the actors of a cooperative system can cooperate in order to complete a job. They start with an initial description of the problem they have to solve (which is an initial o-state) and should end up with a final o-state. Formally, the problem is as follows:

**Reachability Problem**

| | |
|---|---|
| Instance: | cooperative system $\mathcal{S}$, initial o-state $\gamma_0$, and final o-state $\gamma_f$; |
| Question: | is $\gamma_f$ reachable from $\gamma_0$? |

Knowing that a final o-state $\gamma_f$ can be reached from an initial o-state $\gamma_0$ is important. However, in many practical cases it is not enough. One might also want to know how this o-state is reachable. For example, if a cooperative system models an ETI and during the intubation the patient dies, then it is not important at all that the intubation was successful. The intubation should be done in such a way that the patient is kept alive. Therefore, each intermediate state should satisfy some property.

Given a predicate $P$ over $\Gamma$, we say that an o-state $\gamma'$ is $P$*-reachable* in $\mathcal{S}$ from an o-state $\gamma$ if there exists a computation

$$(t_0, q_0^1, \ldots, q_0^n, \gamma_0) \vdash \cdots \vdash (t_i, q_i^1, \ldots, q_i^n, \gamma_i) \vdash \cdots$$
$$\vdash (t_k, q_k^1, \ldots, q_k^n, \gamma_k)$$

such that $t_0 = 0$, $\gamma = \gamma_0$, $\gamma' = \gamma_k$, and $P(\gamma_i)$ holds true, for all $i$.

**P-Reachability Problem**

| | |
|---|---|
| Instance: | cooperative system $\mathcal{S}$, initial o-state $\gamma_0$, final o-state $\gamma_f$, and predicate $P$ over $\Gamma$; |
| Question: | is $\gamma_f$ $P$-reachable from $\gamma_0$? |

It is obvious that reachability is a particular case of P-reachability (the case where $P$ is satisfied by all o-states).

Many practical cases require that specific jobs be completed in a given amount of time. For example, a robot sent in space to fix a satellite should complete the job in a given time interval, a medical procedure should be applied to a patient in a given time interval etc. Thus, we define the following version of the P-reachability problem.

**Time-reachability Problem**

| | |
|---|---|
| Instance: | cooperative system $\mathcal{S}$, initial o-state $\gamma_0$, final o-state $\gamma_f$, predicate $P$ over $\Gamma$, and time value $t$; |
| Question: | is $\gamma_f$ P-reachable from $\gamma_0$ in time $t' \leq t$? |

## 4 Endotracheal Intubation from a Formal Point of View

The main cause of failure in applying ETI is the inability to visualize the larynx during laryngoscopy after neck flexion and external cricoid pressure was applied [10]. Therefore, training scenarios with corresponding recuperative actions should take into consideration the following two main cases:

1. *Blocked airway*. In such cases the airway is blocked by a foreign object or by the inflated tongue. An anti-inflammatory solution is usually administered and/or an alternative route for the air is found. In the AR intubation scenario (Section 2), the blocked airway can be simulated by changing one of the HPS parameters. Let us define the parameter $x \in \{0, 1\}$, where 0 denotes a normal condition for intubation and 1 denotes a blocked airway condition;

2. *Halo cervical traction* condition is usually caused by fractured vertebrae. As a consequence, the head cannot be aligned in the correct intubation position. In such cases, an alternative cuffed pharyngeal tube is employed. The halo cervical traction scenario can be simulated by changing one of the HPS parameters. Let us denote this parameter by $y \in \{0, 1\}$, where 0 denotes a normal condition for intubation and 1 denotes a halo cervical situation.

ETI can be formalized as a reachability problem in our model of cooperative systems. We shall consider a very simplified scenario just as a running example for the paper:

1. *Parameters* that characterize the system:
    - HPS mechanical parameters that allow modification of the mechanical properties of the simulator:
        - blocked airway (through swollen tongue) parameter $x \in \{0, 1\}$;

- halo cervical traction condition parameter $y \in \{0, 1\}$;

- HPS soft parameters whose values are obtained by monitoring the patient:
  - breathing rate $br$, measured as breathing cycles per minute (e.g. adult 12-20 cycles/min). $br$ must be maintained in a given interval $[br_{min}, br_{max}]$ during any ETI procedure;
  - heart rate $hr$, measured as beats per minute (e.g adult 60-80 beats/min). $hr$ must be maintained in a given interval $[hr_{min}, hr_{max}]$ during any ETI procedure;

- Position and orientation information for the virtual models superimposition given as the tube dynamic tracking parameters. This parameter, denoted $po$, is an array of 8 values, the first 4 represent the orientation quaternion, followed by the 3 values for translation and one error term [12];

- Pressure $p$ measured on the tube tip during intubation and which must be maintained in a given interval $[0, p_{max}]$.

For each parameter, its domain is a finite set;

2. An *o-state* is of the form $\gamma = (x, y, br, hr, po, p)$. The initial o-state $\gamma_0$ characterizes the fact that $HPS$ is in the horizontal position, the endotracheal tube is ready for intubation, and the other parameters have some initial values. The final o-state $\gamma_f$ is characterized by the fact that the $HPS$ is in the horizontal position and the endotracheal tube is correctly placed in the trachea;

3. *Actors* in the system are two medical doctors $A_1$ and $A_2$ which are the instructors, and one student $A_3$. They collaborate (interact) through their AR-based interface. The instructors can modify the system parameters to simulate difficult cases for the student. Therefore, the set of actors is $\mathcal{A} = \{A_1, A_2, A_3\}$ (their actions will be described later);

4. The *environment* in which the actors perform is subjected to several constraints such as:
   - Network communication delays (e.g., transmission, propagation, and buffering delay). These delays depend on the communication infrastructure;
   - System delays (e.g., rendering virtual components and tube tracking delay). These delays depend of the complexity of each local system (e.g., for more tracking sensors connected that the node the update cycle will increase). A complex system can produce delays that will affect the rendering cycle of the virtual components to unacceptable frame rates (i.e., the frame rate drops below 30 frames per second);

5. The main *objective* is to train the student to perform a correct ETI, characterized by:
   - a short period of time $t$ for performing the intubation ($t \leq t_{max}$);
   - keeping the vocal cords and other internal tissue intact (pressure on the internal tissue should be $p \leq p_{max}$);
   - the HPS parameters $br$ and $hr$ should be maintained in the appropriate intervals $[br_{min}, br_{max}]$ and $[hr_{min}, hr_{max}]$, respectively, during the intubation procedure.

During the ETI procedure, instructors may change the current values of these parameters facing the student with more or less difficult cases. We are focusing our attention on the following four cases:

- Normal case ($x = 0$ and $y = 0$);
- Blocked airway ($x = 1$ and $y = 0$);
- Halo cervical condition ($x = 0$ and $y = 1$);
- Halo cervical condition and blocked airway ($x = 1$ and $y = 1$).

We assume that $A_1$ and $A_2$ are global actors; they can monitor all parameters but $A_1$ can only change the parameters in $\{x, y\}$, and $A_2$, the ones in $\{br, hr\}$. The student is a local actor and, indirectly (by his actions), can modify the parameters in the set $\{br, hr, po, p\}$.

A possible scenario for $A_1$ is the one given by the automaton in Figure 4. Each state corresponds exactly to one of the four case mentioned above. $A_1$ can modify $x$ or $y$ in an obvious way as described by his automaton. For instance, "$y = 0 | y = 1$" says that $A_1$ can trigger a halo cervical condition by setting $y = 1$, whenever $y = 0$ and independently of $x$'s value (the other actions are similarly interpreted). The only restriction is that once the halo cervical condition has been set it is irreversible for one simulation cycle.

A corresponding scenario for $A_2$ is described in Figure 5. The arc labeled "$[br, hr] | ]br[$" says that $A_2$ can modify the breathing rate to bring it outside the normal breathing rate interval, denoted "$]br[$", whenever $br$ and $hr$ are in their associated normal intervals, denoted "$[br, hr]$" (the other labels are similarly interpreted). The purpose of this transition is to train the student to handle additional complications arising during the intubation procedure.

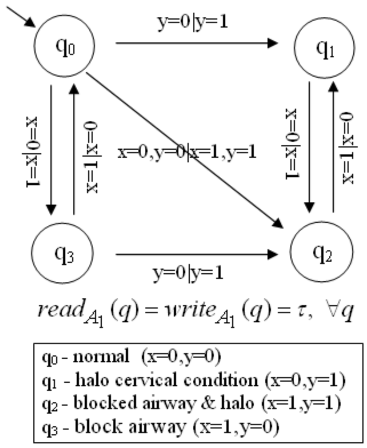


**Figure 4. Actor** $A_1$

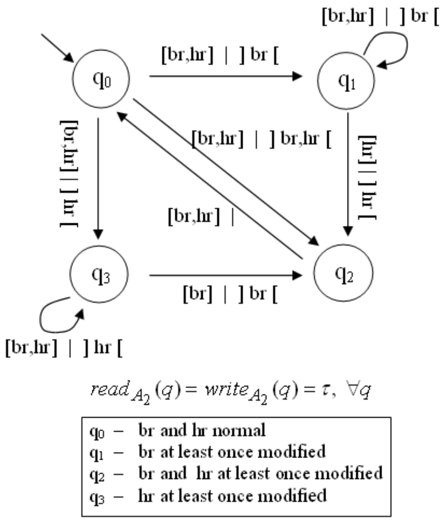


**Figure 5. Actor** $A_2$

Both automata are characterize by constant read- and write-times $\tau$.

At a remote site, $A_3$ performs the intubation procedure and works toward ameliorating the conditions imposed by $A_1$ and $A_2$, as part of his training exercise. Figure 6 describes the possible states for the HPS as an effect of $A_3$'s interaction. All actions performed by the student can be grouped into three classes according to $x$ and $y$'s values. Only the actions for the normal intubation procedure ($x = 0$, $y = 0$) are illustrated. The other three difficult cases are grouped into two dashed boxes $A_{32}$ and $A_{33}$. The arc labels have similar interpretation. For instance, "$[br, hr]|po + \Delta$" means that the student will proceed with the intubation procedure by advancing the tube insertion by a value $\Delta$, provided that the HPS's breathing and heart rate are normal.

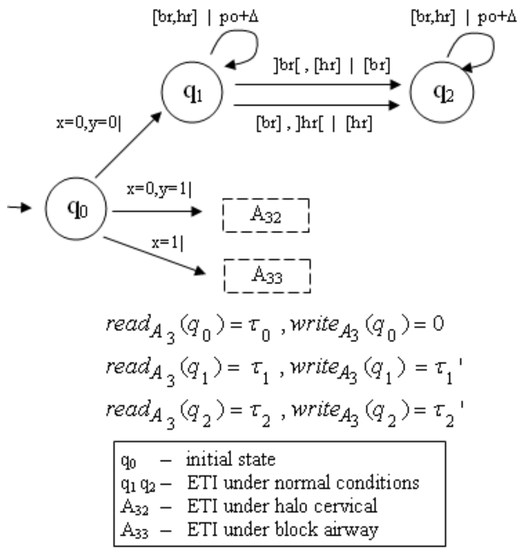


**Figure 6. Actor** $A_3$

We will now consider an example of computation. Let $\gamma_0 = (0, 0, 15, 65, po, 0)$, where $po$ gives the initial parameters of the tube. The following sequence of transitions define a computation:

$$\begin{array}{lll}
(0, q_0, q_0, q_0, \gamma_0) & \vdash & (\tau_0, q_0, q_0, q_1, \gamma_0) \\
& \vdash & (\tau_0 + \tau_1 + \tau_1', q_0, q_0, q_1, \gamma_1) \\
& \vdash & (\tau_0 + \tau_1 + \tau_1' + \tau, q_0, q_1, q_1, \gamma_2) \\
& \vdash & (\tau_0 + 2\tau_1 + 2\tau_1' + \tau, q_0, q_1, q_2, \gamma_3) \\
& \vdash & (\tau_0 + \tau_1 + \tau_1' + \tau, q_0, q_1, q_2, \gamma_4)
\end{array}$$

where

$$\begin{array}{lll}
\gamma_1 & = & (0, 0, 15, 65, po + \Delta, 0) \\
\gamma_2 & = & (0, 0, 25, 65, po + \Delta, 0) \\
\gamma_3 & = & (0, 0, 18, 65, po + \Delta, 0) \\
\gamma_4 & = & (0, 0, 15, 65, po + 2\Delta, 0)
\end{array}$$

The first two steps in the computation above are performed by the student who reads the initial configuration in time $\tau_0$ and then advances the tube into the trachea by $\Delta$ units in time $\tau_1 + \tau_1'$. Now, the second instructor faces the student with a difficult situation by bringing the parameter $br$ out of its normal values ($\gamma_2$). This case is managed by the student whose actions will hopefully bring the parameter back into its normal interval; his new state is $q_2$. Then, the student continues the intubation.

Each biological parameter is characterized by two intervals: a *normal interval* which gives the normal values of the parameters, and a *survival interval* which extends the normal interval with *critical values* (i.e., values which are critical for the patient's life). Values outside the survival interval are regarded as *fatal values*. We assign each parameter ($br$ and $hr$) a survival interval and define the predicate $P$ which is satisfied by an o-state $\gamma$ if and only if the parameters are in the corresponding survival intervals. For instance, if we assume that 25 is in the survival interval for $br$, then the computation above is a P-computation.

## 5 Basic Properties of Cooperative Systems

We have introduced CSs and showed how they can be used to model CARSs. In this section we will conduct a short investigation of a few basic properties of CSs.

### 5.1 Petri nets and Cooperative Systems

It is important to know the relationship between cooperative systems and other models focusing on concurrency, distribution, and cooperation. One of these models is that of a Petri net [1] [13].

Petri nets can be viewed as cooperative systems without time-constraints. Indeed, let $\Sigma$ be a Petri net. To each place $s$ an integer variable $x_s$ is associated, and to each transition $t$ an actor $A_t$ is associated, as follows:

- $A_t$ has exactly one state $q_0$, which is also the initial state of the actor;
- $A_t$'s transition function $\delta$ is defined for any pair $(q_0, \gamma)$ satisfying $\gamma(x_s) \geq W(s,t)$ for all $s$. Moreover, $\delta(q_0, \gamma)$ is defined by $(q_0, \gamma')$, where

  $$\gamma'(x_s) = \gamma(x_s) - W(s,t) + W(t,s),$$

  for all $s$.

For each automaton $A_t$, $read_{A_t}(q_0) = write_{A_t}(q_0) = 0$. This construction is illustrated in Figure 7 (the inscription on the arc in Figure 7(b) says that the transition can be applied only if $x_{s_1} \geq 1$ and $x_{s_2} \geq 1$ and, in this case, $x_{s_2}$ will be decremented and $x_{s_3}$ will be incremented).

Time-constraints can be added in an arbitrary but fixed way.

Given a marking $M$ of $\Sigma$, define the o-state $\gamma_M$ by $\gamma_M(x_s) = M(s)$, for all $s$. Now, it is easy to see that for any two markings $M$ and $M'$, and any transition $t$, we have

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

where $\overset{A_t}{\vdash}$ specifies that the transition is performed by $A_t$.

[1] A *Petri net* is a tuple $\Sigma = (S, T, F, W)$, where $S$ and $T$ are two finite sets (of *places* and *transitions*, respectively), $S \cap T = \emptyset$, $F \subseteq (S \times T) \cup (T \times S)$ is the *flow relation*, and $W : (S \times T) \cup (T \times S) \to \mathbf{N}$ is the *weight function* of $\Sigma$ verifying $W(x,y) = 0$ iff $(x,y) \notin F$).

The *transition relation* of a Petri net $\Sigma$ states that a transition $t$ is *enabled* at a marking $M$, denoted $M[t\rangle$, if $M(s) \geq W(s,t)$ for all $s \in S$. If $t$ is enabled at $M$, then it can occur yielding a new marking $M'$ given by $M(s) = M(s) - W(s,t) + W(t,s)$ for all $s \in S$; we denote this by $M[t\rangle M'$.

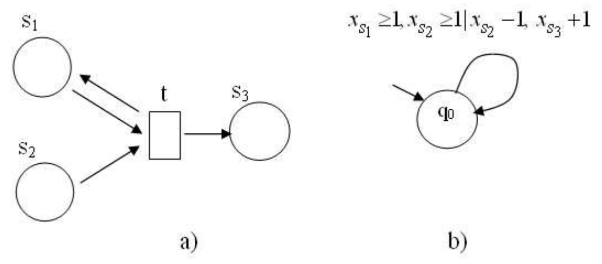


**Figure 7. a) A transition $t$; b) The actor $A_t$**

Under monotonicity and local finiteness restrictions, cooperative systems without time-constraints can be simulated by Petri nets. First, we say that a cooperative system $\mathcal{S}$ is *monotonic* if:

- for any variable $x$, its domain is $\mathbf{N}$;
- 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 $A$, the following property holds true:

$$\gamma' - \gamma = V_i,$$

for some $i$.

Now, for a monotonic and locally finite cooperative system $\mathcal{S}$ without time-constraints, define a Petri net $\Sigma$ as follows:

- to each variable $x$ associate a place $s_x$;
- consider a new place $s_0$ which is going to keep track of the time;
- to each actor $A$ and state $q$ of $A$ associate a place $s_{A,q}$. A token in $s_{A,q}$ denotes that $A$ is in state $q$;
- let $A$ be an actor, $q$ and $q'$ states in $A$, and let $V_1, \ldots, V_p$ be vectors as above. For each vector $V_i$ define

  $$\Gamma(A, q, q', V_i) = \{\gamma | \exists \gamma' : (q', \gamma') \in \delta_A(q, \gamma),\; \gamma' - \gamma = V_i\}$$

  and let $min(\Gamma(A, q', q, V_i))$ be the set of minimal elements of $\Gamma(A, q', q, V_i)$. $min(\Gamma(A, q', q, V_i))$ is a finite set; it is non-empty if $\Gamma(A, q', q, V_i)$ is.

  To each $\gamma \in min(\Gamma(A, q', q, V_i)$ associate a transition $t_{A,q,q',V_i,\gamma}$ and connect it to places as follows:

– $W(s_x, t_{A,q,q',V_i,\gamma}) = \gamma(x)$,
– $W(s_{A,q}, t_{A,q,q',V_i,\gamma}) = 1$,
– $W(t_{A,q,q',V_i,\gamma}, s_x) = \gamma(x) + V_i(x)$,
– $W(t_{A,q,q',V_i,\gamma}, s_{A,q'}) = 1$, and
– $W(t_{A,q,q',V_i,\gamma}, s_0) = read_A(q) + write_A(q)$,

for all variables $x$.

Given a configuration $c = (t, q^1, \ldots, q^k, \gamma)$ of $\mathcal{S}$, define the marking $M_c$ by:

- $M_c(s_x) = \gamma(x)$ for all variables $x$,
- $M_c(s_{q^i}) = 1$ for all $1 \leq i \leq k$,
- $M_c(s_q) = 0$ for all $q \notin \{q^1, \ldots, q^k\}$, and
- $M_c(s_0) = t$.

Now, due to the monotonicity and local finiteness properties, one can easily prove that

$$c = (t_1, q_1^1, \ldots, q_1^k, \gamma) \vdash (t_2, q_2^1, \ldots, q_2^k, \gamma') = c'$$

if and only if $M_c[t_{A,q_1,q_2,V,\bar{\gamma}}\rangle M_{c'}$, for some $t_{A,q_1,q_2,V,\bar{\gamma}}$.

The construction above is illustrated in Figure 8. The

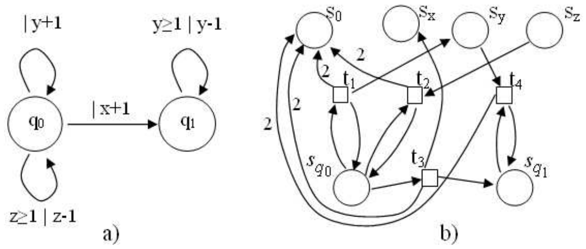


**Figure 8. a) A CS with only one actor $A$; b) The Petri net associated to the CS in a)**

automaton $A$ is monotonic and satisfies the local finiteness condition. All read- and write-time values are considered 1. The $\Gamma$-sets are:

- $\Gamma(q_0, q_0, (0,1,0)) = \{(x,y,z) | x, y, z \geq 0\}$,
- $\Gamma(q_0, q_0, (0,0,-1)) = \{(0,0,z) | z \geq 1\}$,
- $\Gamma(q_0, q_1, (1,0,0)) = \{(x,y,z) | x, y, z \geq 0\}$, and
- $\Gamma(q_1, q_1, (0,-1,0)) = \{(0,y,0) | y \geq 1\}$.

Based on these sets, the Petri net in Figure 8(b) can be easily obtained. The notation for transitions is

- $t_1 = t_{A,q_0,q_0,(0,1,0),(0,0,0)}$, $t_2 = t_{A,q_0,q_0,(0,0,-1),(0,0,1)}$,
- $t_3 = t_{A,q_0,q_1,(1,0,0),(0,0,0)}$, $t_4 = t_{A,q_1,q_1,(0,-1,0),(0,1,0)}$.

## 5.2 The Reachability Problem

As we have mentioned in Section 3 and also exemplified in Section 4, objectives of CARSs can be formulated as reachability problems for CSs. In this section we will investigate this problem.

First, we remark that counter machines [7] can be easily simulated by cooperative systems with just a single global actor, if infinite domains are allowed. Counters are modeled by integer variables and the actor simulates the transition relation in a straightforward way. If no global actor is allowed, but infinite domains are, the simulation of a counter machine can be done by associating local actors to counters (one for each counter). If a transition of the counter machine modifies simultaneously more than one counter, then the corresponding actors should be correlated with each other by variables in a straightforward way.

As a conclusion, the halting problem for counter machines can be reduced to the reachability problem for cooperative systems, and the following result directly follows.

**Theorem 1** The reachability problem for cooperative systems is undecidable.

We consider now the case of finite-domain CSs (i.e., each variable has associated a finite domain).

Instances of the time-reachability problem where $t$ is of polynomial size and $P$ can be verified in polynomial time (w.r.t. the size of the cooperative system), play an important role in practice. The problem consisting of all these instances will be called the *polynomial time-reachability problem* for cooperative systems.

**Theorem 2** The polynomial time-reachability problem for finite-domain cooperative systems is NP-complete.

**Proof** First, we show that the problem is in NP. Consider the algorithm

**input:** CS $\mathcal{S}$, initial o-state $\gamma_0$, final o-state $\gamma_f$, predicate $P$ verifiable in polynomial time and time value $t$ of polynomial size (w.r.t. $||\mathcal{S}||$);
**output:** "yes", if $\gamma_f$ is $P$-reachable in time $t' \leq t$, and "no", otherwise;

```
begin
  guess a sequence of transitions of length at most t
        such that the first one rewrites γ0 and the
        last one ends up with γf;
  if each transition in the sequence verifies P
    then if the sequence induces a computation
            then "yes" else "no";
end.
```

It is straightforward to see that the algorithm runs in nondeterministic polynomial time w.r.t. the size of $\mathcal{S}$ and decides the polynomial time-reachability problem. Therefore, this problem is in NP.

In order to prove that the problem is NP-hard, we will exhibit a reduction from the Hamiltonian circuit problem.

Let $G = (V, E)$ be a directed graph and $v_0$ be an arbitrary but fixed node in $G$. Without loss of the generality we may assume that $|V| \geq 2$. Define the following instance of the time-reachability problem for finite-domain CSs:

- to each node $v$ we associate a variable $x_v$ whose domain is $\{0, 1, 2\}$. The value 0 for a variable $x_v$ means that "$v$ has not yet been visited", the value 1 means that "$v$ is the current node", and the value 2 means that "$v$ has already been visited";
- to each arc $(v, v')$ with $v \neq v'$ we associate an actor $A_{v,v'}$, with only one state $q_0$, and whose transitions are $(q_0, \gamma') \in \delta(q_0, \gamma)$ iff either
  - $\gamma(x_v) = 1$, $\gamma(x_{v'}) = 0$, $\gamma'(x_v) = 2$, $\gamma'(x_{v'}) = 1$, and $\gamma(x) = \gamma'(x)$, for all $x \neq x_v, x_{v'}$, or
  - $v' = v_0$, $\gamma(x_v) = 1$, $\gamma'(x_v) = 2$, and $\gamma(x) = \gamma'(x)$, for all $x \neq x_v$;
- $read_{A_{v,v'}}(q_0) = write_{A_{v,v'}}(q_0) = 1/2$, for all $v \neq v'$;
- let $\gamma_0 = (1, 0, \ldots, 0)$, and $\gamma_f = (2, 2, \ldots, 2)$ (the first coordinate corresponds to $x_{v_0}$);
- $\mathcal{C}(\gamma)$ can be any number greater than or equal to $1/2$;
- let $P$ be the *true* predicate (satisfied by all o-states).

It is easily seen that the CS above can be constructed in polynomial time with respect to $|V|$ (there are at most $|V|^2$ actors and each actor has exactly one state and one transition). Moreover, there exists a Hamiltonian circuit in $G$ if and only if $(|V|, q_0, \ldots, q_0, \gamma_f)$ is $P$-reachable. □

## 6 Conclusion and Future Work

CARSs enable collaborative work spaces separated by miles of distance to appear as one; remote and local teams appear to be one team co-present in a space that is both physical and virtual. For communities of distributed emergency teams, medical teams, and engineering design groups, these networked spaces will open the door to a pattern of work and collaboration that can bridge both distances and make creative, collaborative ideas feel physical, tangible, and most of all, shared. As we enter a period where work teams are distributed internationally and travel is deeply uncertain and vulnerable to disruption, this is a vision not only worth pursuing, but one that calls out for research.

This work proposes an automata-based formal model for CARSs that allows and in-depth analysis of such cooperative systems. The model seems to be a good fit and its properties indicate a high potential for further investigation. Thus, we were able to establish a connection between cooperative systems and Petri nets, a well-studied model of distribution and concurrency. We also showed that reachability for cooperative systems is in general undecidable, and it is NP-complete for the finite-domain ones.

Many problems remain to be investigated. First of all, an in-depth study of the basic properties of the model is necessary by exploiting the rich and mature automata theory apparatus. Verification techniques should be considered too. Many such techniques developed nowadays are based on automata and, therefore, we anticipate the application of these techniques to cooperative AR systems through the proposed model. While we are currently focusing on these problems, a software tool allowing simulation, testing, and validation of CARSs is under development.


### Acknowledgments

We would like to thank Dr. Charles Hughes for a careful review of the manuscript, and Dr. Jannick Rolland for continuous support.


## References


[1] M. Billinghurst and H. Kato. *Collaborative Augmented Reality*. Communications of the ACM, Vol. 45, 64–70, 2002.

[2] F. Biocca, and J.P. Rolland. *Teleportal Face-to-face System: Teleconferencing and Telework Augmented Reality System*. Patent 6550-00048, MSU 99-029, US:Michigan State University and University of Central Florida, 2000.

[3] P. Biro, and M. Weiss. *Difficult Airway Management: Clinical Evaluation of Video-assisted Tracheal Intubation*. University of Zurich, www.research-projects.unizh.ch/med/unit40100/area145/p2213.htm, 2001.

[4] T.P. Caudell, and D.W. Mizell. *Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes*. IEEE International Conference on Systems Sciences, Hawaii, 1992.

[5] H. Fuchs, A. State, M. Livingston, W. Garrett, G. Hirota, M. Whitton, and E. Pisano *Virtual Environments Technology to Aid Needle Biopsies of the Breast: An Example of Real-Time Data Fusion*. Medicine Meets Virtual Reality (MMVR), Amsterdam, NL, 1996.

[6] F.G. Hamza-Lup, J.P. Rolland and C. Hughes. *A Distributed Augmented Reality System for Medical Training and Simulation*. Energy, Simulation-Training, Ocean Engineering and Instrumentation: Research Papers of the Link Foundation Fellows, Vol.4, Rochester Press, 2004.

[7] J.E. Hopcroft, R. Motwani, and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison Wesley, 2nd Ed., 2001.

[8] METI Medical Educational Technologies. *Human Patient Simulator.* Technical Guide, `http://www.meti.com/Art/downloads/HPSCF.pdf`, 2005.

[9] F.G. Hamza-Lup, A. Santhanam, C. Fidopiastis and J.P. Rolland. *Distributed Training System with High-Resolution Deformable Virtual Models*. 43rd Annual ACM Southeast Conference (ACMSE), Kennesaw, GA, 2004.

[10] C.K. Koay. *Difficult Tracheal Intubation - Analysis and Management in 37 Cases*. Singapore Medical Journal, Vol.39(3), 112–114, 1998.

[11] P. Milgram and F. Kishino. *A Taxonomy of Mixed Reality Visual Displays*. IECE Transactions on Information and Systems, Vol.E77-D(12), 1321–1329, 1994.

[12] Northern Digital Inc. *6DOF Optical Tracking System - Optotrak 3020*. Technical Guide, 2002.

[13] W. Reisig. *Petri Nets. An Introduction*. Springer-Verlag, 1985.

[14] D. Schmalstieg and G. Hesina. *Distributed Applications for Collaborative Augmented Reality*. IEEE Virtual Reality, Orlando, FL, 2002.

[15] B.H. Thomas, G. Quirchmayr, and W. Piekarski. *Through-Walls Communication for Medical Emergency Services*. International Journal of Human-Computer Interaction, Vol.16(3), 477–496, 2003.

[16] M. Zyda and S. Singhal. *Networked Virtual Environments, Design and Implementation*. New York: Addison Wesley, 1999.

[17] R.M. Walls, E.D. Barton, and A.T. McAfee. *2,392 Emergency Department Intubations: First Report of the ongoing National Emergency Airway Registry Study*. Annals of Emergency Medicine, Vol.26, 364–403, 1999.

[18] A. Webster, S. Feiner, B. MacIntyre, W. Massie, and T. Krueger. *Augmented Reality in Architectural Construction, Inspection and Renovation*. ASCE 3rd Congress on Computing in Civil Engineering, Anaheim, CA, 1996.