A Distributed Logic for Networked Cyber-Physical Systems

Minyoung Kim, Mark-Oliver Stehr, and Carolyn Talcott

SRI International,
mkim,stehr,clt@csl.sri.com

Abstract. A distributed logical framework designed to serve as a declarative semantic foundation for Networked Cyber-Physical Systems provides notions of facts and goals that include interactions with the environment via external goal requests, observations that generate facts, and actions that achieve goals. Reasoning rules are built on a partially ordered knowledge-sharing model for loosely coupled distributed computing. The logic supports reasoning in the context of dynamically changing facts and system goals. It can be used both to program systems and to reason about possible scenarios and emerging properties.
The underlying reasoning framework is specified in terms of constraints that must be satisfied, making it very general and flexible. Inference rules for an instantiation to a specific local logic (Horn clause logic) are given as a concrete example. The key novel features are illustrated with snippets from an existing application—a theory for self-organizing robots performing a distributed surveillance task. Traditional properties of logical inference and computation are reformulated in this novel context, and related to features of system design and execution. Proofs are outlined for key properties corresponding to soundness, completeness, and termination. Finally, the framework is compared to other formal systems addressing concurrent/distributed computation.
Keywords: Distributed declarative logic, partially ordered knowledge, networked cyber-physical systems.

1 Introduction

We present a novel distributed logic framework intended to serve as a semantic foundation for Networked Cyber-Physical Systems (NCPS). NCPS present many challenges that are not suitably addressed by existing distributed computing paradigms. They must be reactive and maintain an overall situation awareness that emerges from partial distributed knowledge. They must achieve system goals through local, asynchronous actions, using (distributed) control loops through which the environment provides essential feedback. NCPS should be resilient to failures of individual elements, readily adapt to changing situations, and often need to be rapidly instantiated and deployed for a given mission.

To address these challenges, we are developing a logical framework for NCPS that combines distributed reasoning and asynchronous control in space and time.

The purpose of logic in this context is many-fold. First of all, it provides a language to express and communicate system goals. Dually, it allows expressing and communicating facts about the current system state. In both cases, communication includes communication with the users but also communication among the system components themselves. At the level of an individual cyber-physical component, the logic provides a declarative interface for goal-oriented control and feedback through observations that are represented as logical facts. Finally, it provides a framework for inference and computation, which allows facts and goals to interact with each other and form new facts or goals. Our aim is a solution to declarative control that covers the entire spectrum between cooperation and autonomy, makes opportunistic use of networking resources, and adapts to changing resource constraints.

In the following we present a distributed inference system that is a significant step toward this goal. Our logical framework is based on partially ordered knowledge sharing, a distributed computing paradigm for loosely coupled systems that does not require continuous network connectivity. We use Horn clause logic to illustrate our approach, which we expect to generalize to more expressive logics. The features of the framework are illustrated using a theory of self-organizing robots. A simplified version of the inference system was presented in [11]. The main contributions of this paper are

- the fully general inference system with explicit derivations,
- the identification of conditions under which key properties such as soundness, completeness, termination, and confluence hold, and
- the application of our results to a theory for self-organizing robots

2 Case Study: Self-Organizing Robots

We focus on networked cyber-physical systems S with a finite set of cyber-nodes. Two cyber-nodes have the capability to communicate whenever the network conditions permit. Each cyber-node can have sensors that can generate observations at arbitrary time points, and actuators driven by goals. S may operate under arbitrary conditions, so there is no guarantee that goals will be achieved. Consider a self-organizing network of mobile robots deployed in a building, e.g., for situational awareness during an emergency. In this paper, we use an abstract topological *mobility model* where a robot is located in some area and can move to any adjacent area. Each area is equipped with acoustic or motion sensors. The robots use a common logical theory that specifies a language (constants, functions, and predicates) and local inference rules based on Horn clause logic. A robot's local knowledge (state) consists of a set of facts and a set of goals. Facts are formulas derived by logical inference or by *observation* of the environment. Goals are formulas expressing what the system should achieve and drive the inference process. Goals can arrive from the environment at any time. They can also be generated as subgoals during local inference. Robots can exchange knowledge (i.e., facts and goals) opportunistically if they reside in the same or adjacent rooms.

Forward Clauses:

$F1: Noise(T, A) \Rightarrow Trigger(T, A).$
$F2: Motion(T, A) \Rightarrow Trigger(T, A).$
$F3: Adjacent(A, B) \Rightarrow Adjacent(B, A).$

Backward Clauses:

$B1: Interest(T_I, I, R) \Leftarrow Result(T_I, T_T, 0, I), Deliver(T_I, T_T, 1, I, R).$
$B2: Deliver(T_I, T_T, N_D, I, R) \Leftarrow Delivered(T_I, T_T, N_D, I, R).$
$B3: Deliver(T_I, T_T, N_D, I, R) \Leftarrow$
$\quad Position(T_P, R, A), Position(T'_P, R', A'), R' \neq R,$
$\quad MoveTo(T_I, T_T, N_D, 0, \infty, R', A), Deliver(T_I, T_T, N_D, I, R).$
$B4: Result(T_I, T_T, N_D, I') \Leftarrow CompImage(T_I, T_T, N_D, I), I' = Extract(I).$
$B5: CompImage(T_I, T_T, N_D, I') \Leftarrow RawImage(T_I, T_T, N_D, I), I' = Compress(I).$
$B6: RawImage(T_I, T_T, N_D, I) \Leftarrow Trigger(T_T, A), T_I \leq T_T,$
$\quad MoveTo(T_I, T_T, N_D, 0, T_T + \Delta t_{sd}, R, A),$
$\quad TakeSnapshot(T_I, T_T, N_D, T_T + \Delta t_{sd}, A, I).$
$B7: TakeSnapshot(T_I, T_T, N_D, D, A, I) \Leftarrow$
$\quad Snapshot(T_I, T_T, N_D, T_S, A, I), T_T \leq T_S, T_S \leq D.$
$B8: MoveTo(T_I, T_T, N_D, W', D, R, B) \Leftarrow Position(T_P, R, B), T_P \leq D.$
$B9: MoveTo(T_I, T_T, N_D, W', D, R, B) \Leftarrow Adjacent(A, B), W' > -b_w, W = W' - 1,$
$\quad MoveTo(T_I, T_T, N_D, W, D, R, A), Move(T_I, T_T, N_D, W', D, R, A, B).$

Replacement Ordering: (f denotes a fact and g a goal and x denotes either)

$O1: f: Position(t_P, r, \ldots) \prec f: Position(t'_P, r, \ldots)$ if $t_P < t'_P$.
$O2: x: X(t_I, \ldots) \prec g: Interest(t'_I, \ldots)$ if $t_I < t'_I$.
$O3: x: X(t_I, t_T, 0, \ldots) \prec f: Result(t_I, t_T, 0, \ldots)$ if $x: X \neq f: Result$.
$O4: x: X(t_I, t_D, 1, \ldots) \prec f: Deliver(t_I, t_D, 1, \ldots)$ if $x: X \neq f: Deliver$.

Variables: T: time, D: snapshot deadline, A and B: area, R: robot,
I: image or derived information, N: identifier, W: weight
Constants: Δt_{sd}: relative snapshot deadline (max. delay from trigger event),
b_w: bound for weight (diameter of the floor plan)

Fig. 1. Logical Theory for Self-Organizing Robots

Figure 1 shows the logical theory that is used to specify the possible behaviors of our self-organizing robots. The clauses are partitioned into forward and backward rules, providing a means for controlling inference/execution. Forward clauses such as the trigger conditions $F1$ and $F2$ can be applied at any time when the conditions are met. Backward clauses are applied only when the conclusion formula matches (unifies with) an existing goal. Goal atoms appearing as premises in forward or backward clauses generate new goals to be satisfied in an execution. The primary goal is delivery of images I to a node r, $Interest(T_I, I, r)$. Figure 2, shows a possible execution of the theory of Figure 1 achieving an instance of the $Interest$ goal. The variables (T_I,T_T,N_D) are suppressed, as they are fixed for an execution solving primary goal instance. For example, $Result(I)$ abbreviates $Result(t_I, t_T, n_D, I)$ where t_I is the session value of T_I and so on. At the top of Figure 2, the user injects a cyber-goal $Interest(I, r)$ at the root node r. Backward reasoning with clause $B1$ is used to add the first

subgoal, $Result(I)$, to the local knowledge base. Then clauses $B4, B5$ for solving $Result$ goals, are used to add subgoals, $CompImage(I)$ and $RawImage(I)$. Meanwhile, at the bottom of Figure 2, the cyber-fact $Noise(0.0, a)$ is observed by the sensor in area a, and forward reasoning using clause $F1$ leads to the fact $Trigger(0.0, a)$. Clause $B6$ for $RawImage(I)$ has three subgoals involving $Trigger$, $MoveTo$, and $TakeSnapshot$. The leftmost subgoal can be matched with the fact $Trigger(0.0, a)$. Suppose the above reasoning is carried out by robot r in area a and further that a camera robot, x, is in adjacent area b. Then by communication with r, x can learn the $RawImage$ goal, and the $Trigger$ fact and use $B6$ to add a $MoveTo$ goal to its knowledge base and $B8, B9$ to satisfy the goal. Then using its camera, robot x can take a snapshot adding $Snapshot(10.0, a, i)$ to the set of facts and apply $B7, B6$ to realize the $TakeSnapshot(t_D, a, I)$, and the $RawImage(I)$ goals. The goals $CompImage(I)$ and $Result(I)$ can be solved by the robot x, since it has the fact $RawImage(i)$. Alternatively, it could be satisfied by another robot, possibly r, depending on available computational resources. The backward clause $B3$ is used to steer a robot toward the root node r to deliver the image, and $B2$ can be applied once a $Delivered$ fact is available. Then $Interest(I, r)$ can be satisfied.

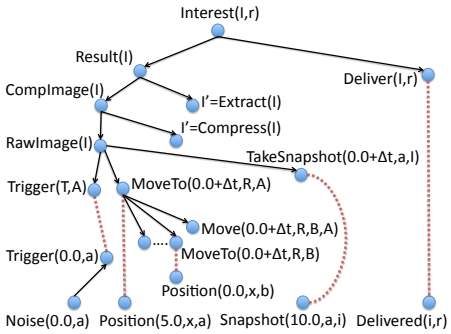

Fig. 2. Example Robot Execution

Unlike traditional logics, new facts and/or goals can arrive at any time, interleaved with local inference processes. For example, a robot can observe its position at different times, and possibly get different answers. Two features of the logical framework help to avoid potential confusion. Certain predicates, called *cyber-predicates*, have time stamps as part of their argument list. For example, position readings are time stamped and thus different readings can be distinguished logically by their time stamp. In addition, the logical theory is augmented by a partial ordering on facts and goals, called the *replacement ordering*. A fact or goal can be replaced by one that is higher in the ordering. This provides a means of removing outdated knowledge from the distributed system state, without any need for synchronization.

The clauses (O1-4) at the end of Figure 1 axiomatize the replacement ordering of the robot theory. Suppose some robot has a fact $Position(0.0, r, b)$ in its knowledge base, stating that robot r is in area b at time 0.0, and later the robot receives the fact $Position(1.0, r, a)$. The replacement rule can be used to remove $Position(0.0, r, b)$ from its set of facts since $Position(0.0, r, b) \prec Position(1.0, r, a)$.

3 The Distributed Logical Framework

Constraints on the Local Theory Let Σ be a signature, $\mathcal{V}$ a countably infinite set of variables, and Ω a fixed finite theory of Horn clause logic over Σ. The sets of *terms* $\mathcal{T}(\Sigma, \mathcal{V})$, and *atoms* $\mathcal{A}(\Sigma, \mathcal{V})$, ground terms $\mathcal{T}(\Sigma)$ and ground atoms $\mathcal{A}(\Sigma)$ are defined as usual. We use P and Q to range over atoms. A (ground) substitution is a mapping from variables to (ground) terms. We let σ range over substitutions, and $\sigma(e)$ denotes the application of a substitution, that is the result of replacing variables in e by their image under σ.

Σ contains built-in constants for natural numbers and names of cyber-nodes. Additional *built-in functions*, and *built-in predicates* can be included in Σ, and the application of a built-in predicate cannot be the conclusion of a clause in Ω. Σ also contains a distinguished set of predicates (distinct from built-ins) called *cyber-predicates*. These predicates define the interface of the logic with the outside world. We use p_c to range over such predicates. The first argument of a cyber-predicate is a natural number interpreted as a timestamp. In the robot theory, $\leq$, *Compress* and *Extract* are built in, while *Snapshot* and *Position* are cyber-predicates. Clauses in Ω are assigned unique labels, for example $l : P_1, \ldots, P_n \Rightarrow Q$ is a clause with label l. In addition $\Omega = \Omega_\mathrm{f} \cup \Omega_\mathrm{b}$, where Ω_f and Ω_b are sets of clauses that we refer to as *forward and backward clauses*, respectively. We use $\vdash$ to denote the *standard derivability* in Horn clause logic with all the built-ins in Σ.

A *fact* is a ground atom. The definition of *goal* is more complex. A subset of the predicates, designated as *goal predicates*, includes at least the built-in predicates and all predicates that appear in the conclusion of a clause from Ω_b. The set of *goals* can be any set of (not necessarily ground) atoms that are applications of goal predicates satisfying the following closure properties: **(1)** If G is a goal then $\sigma(G)$ is a goal. **(2)** If $l : P_1, \ldots, P_n \Rightarrow Q \in \Omega_\mathrm{f}$, $j \in 1, \ldots, n$, P_j is the application of a goal predicate, and $\sigma(P_i)$ is a fact for $i \leq 1 < j$, then $\sigma(P_j)$ is a goal. **(3)** If $l : P_1, \ldots, P_n \Rightarrow Q \in \Omega_\mathrm{b}$, P_j is the application of a goal predicate, $\sigma(P_i)$ is a fact for all $1 \leq i < j \leq n$, and $\sigma(Q)$ is a goal, then $\sigma(P_j)$ is a goal. In the robot theory, $Interest(t_I, I, R)$ is a goal only for ground terms t_I, and $MoveTo(t_I, t_T, n_D, W, D, R, B)$ is a goal for ground terms t_I, t_T, n_D. The capitalized arguments are variables.

We further require the *variable restriction*: **(1)** For $l : P_1, \ldots, P_n \Rightarrow Q \in \Omega_\mathrm{f}$, each variable in Q appears in at least one of $P_1, \ldots, P_n$. **(2)** For $l : P_1, \ldots, P_n \Rightarrow Q \in \Omega_\mathrm{b}$, if $\sigma(Q)$ is a goal, then each variable in $\sigma(Q)$ appears in at least one of $\sigma(P_1), \ldots, \sigma(P_n)$. It is easy to check that our example satisfies this restriction.

Derived Atoms as Knowledge Derived facts and derived goals are objects of the form $f : F$ and $g : G$ that constitute units of knowledge, atoms equipped with an indication of their role and an explanation of their origin. The set of *(atomic) derived facts* and *(atomic) derived goals* together is inductively defined as follows: **(1)** $\mathtt{B}_\sigma(g) : \sigma(G)$ is a derived fact if G is a built-in goal, $\vdash \sigma(G)$, and $g : G$ is a derived goal. **(2)** $\mathtt{O}(F) : F$ is an atomic derived fact, also called an *observation*, for each cyber-fact F, **(3)** $\mathtt{C}(G) : G$ is an atomic derived goal, also called a *control*, for each cyber-goal G; **(4)** $l_\sigma(f_1, \ldots, f_n) : \sigma(Q)$ is a de-

rived fact if $l : P_1, \ldots, P_n \Rightarrow Q \in \Omega_\mathrm{f}$, $\sigma(Q)$ is a fact, and $f_i : \sigma(P_i)$ are derived facts; **(5)** $l_\sigma^{-1}(f_1, \ldots, f_{j-1}) : \sigma(P_j)$ is a derived goal if $l : P_1, \ldots, P_n \Rightarrow Q \in \Omega_\mathrm{f}$, $j \in 1, \ldots, n$, $\sigma(P_j)$ is a goal, and $f_i : \sigma(P_i)$ are derived facts; **(6)** $l_\sigma(f_1, \ldots, f_n; g') : \sigma(Q)$ is a derived fact if $l : P_1, \ldots, P_n \Rightarrow Q \in \Omega_\mathrm{b}$, $\sigma(Q)$ is a fact, $f_i : \sigma(P_i)$ are derived facts, and $g' : G'$ is a derived goal with $\sigma(G') = \sigma(Q)$; and **(7)** $l_\sigma^{-1}(f_1, \ldots, f_{j-1}; g') : \sigma(P_j)$ is a derived goal if $l : P_1, \ldots, P_n \Rightarrow Q \in \Omega_\mathrm{b}$, $j \in 1, \ldots, n$, $\sigma(P_j)$ is a goal, $f_i : \sigma(P_i)$ are derived facts, $g' : G'$ is a derived goal, and $\sigma(G') = \sigma(Q)$.

A *derived atom* is either a derived fact or a derived goal. This is different from standard approaches to explicit proof objects where derivations of goals are not considered. We let $f : F$ range over derived facts with derivation f and underlying fact F. Similarly $g : G$ ranges over derived goals and $d : P$ ranges over derived atoms. Goals may have variables, and we consider two derived goals that differ only by renaming of the variables to be the same. Given a derived atom $d : P$, it is easy to see that P is uniquely determined by d. We write $at(d : P)$ to denote the atom of $d : P$, i.e., P.

We say that $d : P$ is an *immediate subderivation* of $d' : P'$, written $d : P \triangleright d' : P'$, iff d' is of the form $L(\ldots, d, \ldots)$, where L represents any of the above constructors of derivations. $\triangleright^+$ and $\triangleright^*$ denote the transitive and reflexive transitive closure of $\triangleright$, respectively. We let K range over derived atoms and $\mathcal{K}$ range over sets of derived atoms. The *knowledge entailment* relation $\vdash$ is defined inductively by: **(1)** $K \in \mathcal{K}$ implies $\mathcal{K} \vdash K$, and **(2)** $\mathcal{K}' \vdash_1 K''$ and $\mathcal{K} \vdash K'$ for all $K' \in \mathcal{K}'$ implies $\mathcal{K} \vdash K''$, where $\mathcal{K} \vdash_1 K'$ is defined by $K \triangleright K'$ for some $K \in \mathcal{K}$.

We assume that the set of derived atoms is equipped with a quasi-order $\leq$, the so-called *subsumption order*, and a strict partial order $\prec$, the so-called *replacement order*. These relations must not make use of the structure of the derivations other than distinguishing between facts and goals, they must not relate distinct built-in derived atoms, and $\leq$ must not relate derived facts and derived goals. For derived goals $g : G$ and $g' : G'$ with $G = \sigma(G')$ we require $g : G \leq g' : G'$. The induced *subsumption equivalence* $K \equiv K'$ is defined as $K \leq K' \wedge K' \leq K$ and strict subsumption is defined by $K < K'$ iff $K \leq K'$ and $K' \not\leq K$. We require that the replacement order is a *compatible extension* of strict subsumption, that is, **(1)** $K < K'$ implies $K \prec K'$, and **(2)** $K \leq K'$, $K' \prec K''$, and $K'' \leq K'''$ implies $K \prec K'''$. In addition, the relations must satisfy the *ordering consistency* requirements, that is, **(1)** $K \prec K'$ implies $K \not\equiv K'$, and **(2)** $K'_1 \leq K_1 \prec K_2 \leq K'_2$ and $K'_1 < K'_2$ implies $K_1 < K_2$.

Distributed Proofs as Interactive Executions The local state of a cyber-node is of the form $\Gamma \vdash \Delta \ @ \ t, x$, where x is the unique name of the node, t is a natural number representing its local time, and Γ,Δ constitutes the knowledge at the node. Γ is a finite set of derived facts, and Δ is a finite set of derived goals. A configuration of a cyber-physical system S is a set of local states $\Gamma \vdash \Delta \ @ \ t, x$, one for each cyber-node x of S. Given a configuration c containing $\Gamma \vdash \Delta \ @ \ t, x$, we write $\mathcal{F}_x(c)$ and $\mathcal{G}_x(c)$ to denote Γ and Δ, respectively.

Figure 3 gives the proof rules of our logic. The rule (Control) represents the addition of a new user-level objective to the set of system goals. The rule (Ob-

$$\frac{\Gamma \vdash \Delta \ @ \ t, x}{\Gamma \vdash \Delta, \mathtt{C}(G) : G \ @ \ t', x} \quad \text{if } G = p_c(t, \ldots) \text{ is a cyber-goal} \qquad \text{(Control)}$$

$$\frac{\Gamma \vdash \Delta \ @ \ t, x}{\Gamma, \mathtt{O}(F) : F \vdash \Delta \ @ \ t', x} \quad \text{if } F = p_c(t, \ldots) \text{ is a cyber-fact} \qquad \text{(Observation)}$$

$$\frac{\Gamma, f : F \vdash \Delta \ @ \ t, x}{\Gamma \vdash \Delta \ @ \ t', x} \quad \text{if } f : F \prec \Gamma, \Delta \qquad \text{(Replacement1)}$$

$$\frac{\Gamma \vdash \Delta, g : G \ @ \ t, x}{\Gamma \vdash \Delta \ @ \ t', x} \quad \text{if } g : G \prec \Gamma, \Delta \qquad \text{(Replacement2)}$$

$$\frac{\Gamma_x \vdash \Delta_x \ @ \ t_x, x \quad \Gamma_y, f : F \vdash \Delta_y \ @ \ t_y, y}{\Gamma_x, f : F \vdash \Delta_x \ @ \ t'_x, x} \qquad \text{(Communication1)}$$

if $x \neq y$, $t'_x \geq t_y$, and $f : F$ is fresh at x.

$$\frac{\Gamma_x \vdash \Delta_x \ @ \ t_x, x \quad \Gamma_y \vdash \Delta_y, g : G \ @ \ t_y, y}{\Gamma_x \vdash \Delta_x, g : G \ @ \ t'_x, x} \qquad \text{(Communication2)}$$

if $x \neq y$, $t'_x \geq t_y$, and $g : G$ is fresh at x

$$\frac{\Gamma \vdash \Delta, g : G \ @ \ t, x}{\Gamma, \mathtt{B}_\sigma(g) : \sigma(G) \vdash \Delta, g : G \ @ \ t', x} \qquad \text{(Built-in)}$$

if G is a built-in goal with a solution $\sigma(G)$ such that $\mathtt{B}_\sigma(g) : \sigma(G)$ is fresh.

$$\frac{\Gamma, f_1 : \sigma(P_1), \ldots, f_n : \sigma(P_n) \vdash \Delta \ @ \ t, x}{\Gamma, f_1 : \sigma(P_1), \ldots, f_n : \sigma(P_n), f : \sigma(Q) \vdash \Delta \ @ \ t', x} \qquad \text{(Forward1)}$$

if $l : P_1, \ldots, P_n \Rightarrow Q$ is a clause from Ω_f,
$f = l_\sigma(f_1, \ldots, f_n)$, $\sigma(Q)$ is a fact, and $f : \sigma(Q)$ is fresh.

$$\frac{\Gamma, f_1 : \sigma(P_1), \ldots, f_{j-1} : \sigma(P_{j-1}) \vdash \Delta \ @ \ t, x}{\Gamma, f_1 : \sigma(P_1), \ldots, f_{j-1} : \sigma(P_{j-1}) \vdash \Delta, g : \sigma(P_j) \ @ \ t', x} \qquad \text{(Forward2)}$$

if $l : P_1, \ldots, P_n \Rightarrow Q$ is a clause from Ω_f,
$g = l_\sigma^{-1}(f_1, \ldots, f_{j-1})$, $\sigma(P_j)$ is a goal, and $g : \sigma(P_j)$ is fresh.

$$\frac{\Gamma, f_1 : \sigma(P_1), \ldots, f_n : \sigma(P_n) \vdash \Delta, g' : G' \ @ \ t, x}{\Gamma, f_1 : \sigma(P_1), \ldots, f_n : \sigma(P_n), f : \sigma(Q) \vdash \Delta, g' : G' \ @ \ t', x} \qquad \text{(Backward1)}$$

if $l : P_1, \ldots, P_n \Rightarrow Q$ is a clause from Ω_b,
$f = l_\sigma(f_1, \ldots, f_n; g')$, $\sigma(Q) = \sigma(G')$, $\sigma(Q)$ is a fact, and $f : \sigma(Q)$ is fresh.

$$\frac{\Gamma, f_1 : \sigma(P_1), \ldots, f_{j-1} : \sigma(P_{j-1}) \vdash \Delta, g' : G' \ @ \ t, x}{\Gamma, f_1 : \sigma(P_1), \ldots, f_{j-1} : \sigma(P_{j-1}) \vdash \Delta, g' : G', g : \sigma(P_j) \ @ \ t', x} \qquad \text{(Backward2)}$$

if $l : P_1, \ldots, P_n \Rightarrow Q$ is a clause from Ω_b,
$g = l_\sigma^{-1}(f_1, \ldots, f_{j-1}; g')$, $\sigma(Q) = \sigma(G')$, $\sigma(P_j)$ is a goal, and $g : \sigma(P_j)$ is fresh.

$$\frac{\Gamma \vdash \Delta \ @ \ t, x}{\Gamma \vdash \Delta \ @ \ t', x} \qquad \text{(Sleep)}$$

Notes. An implicit side condition $t < t'$ is omitted in all proof rules ($t_x < t'_x$ in the communication rules). In the context of a proof rule that has a premise $\Gamma \vdash \Delta \ @ \ t, x$ we say that K is *fresh* (at x) if there is no $K' \in \Gamma, \Delta$ such that $K \equiv K'$ or $K \prec K'$. In the condition of proof rules we use σ to range over all most general (not necessarily ground) substitutions that satisfy the condition of the proof rule.

Fig. 3. Proof Rules of our Distributed Logical Framework for NCPS

servation) captures the generation of information from the environment, spontaneously or triggered by a goal. The (Replacement) rules are used to overwrite subsumed and obsolete facts and goals. The (Communication) rules allow cyber-nodes to exchange facts or goals by means of asynchronous communication. The time constraints in the rule achieve a minimal level of temporal consistency. The forward and backward rules implement forward and backward reasoning. The rule (Forward1) is the usual Horn clause rule. The rule (Forward2) covers the case where the available facts are not sufficient to apply the clause so that a new subgoal $\sigma(P_j)$ needs to be generated for a missing fact. The backward rules are analogous to the two forward rules, but in addition require the Horn clause conclusion to unify with an existing goal. Finally, the (Sleep) rule allows the system to be inactive, for example to save energy or wait for new knowledge.

The proof rules determine a labeled transition relation $\rightarrow_r$ on configurations of the cyber-physical system S: For configurations c and c', we have $c \rightarrow_r c'$ iff there exists an instance of proof rule r such that c contains the premises of the instance, and c' is obtained by an update of c with the conclusion, i.e., by replacing $\Gamma \vdash \Delta \ @ \ t, x$ by the conclusion $\Gamma' \vdash \Delta' \ @ \ t', x$. In this case, we also say that r is *applicable* at x in c. An execution of the networked cyber-physical system S is a finite or infinite sequence $\pi = c_0, r_0, c_1, r_1, c_2, \ldots$ of configurations such that $c_i \rightarrow_{r_i} c_{i+1}$ for all i, and we say that $c_i \rightarrow_{r_i} c_{i+1}$, or briefly r_i, is the ith step of π. We say that a rule r is *applied* in π at j iff $r = r_j$.

For a given execution π, we denote by $\mathcal{F}^O(\pi)$ all derived facts of the form $\mathtt{O}(F) : F$ generated in π by the observation rule and by $\mathcal{G}^C(\pi)$ all derived goals of the form $\mathtt{C}(G) : G$ generated in π by the control rule.

4 Properties of the Logical Framework

For a logical framework to be a useful semantic foundation it is important that we understand the guarantees provided by the framework. Here we discuss properties of executions, $\pi = c_0, r_0, c_1, r_1, c_2, \ldots$, where c_0 is an initial configuration in which each node has an empty set of facts and goals. Most of these properties are independent of the underlying communication system. Several of the properties only require the Horn clause theory and/or the execution strategy to satisfy additional conditions. Specifically, we consider notions of *Monotonicity*, *Soundness*, *Completeness*, *Termination*, and *Confluence*. These are analogs of properties of traditional inference and computation systems and important for ensuring desired properties of specific cyber-physical systems. In the following, $\pi_{|i}$ denotes the prefix $c_0, r_0, c_1, r_1, c_2, \ldots, c_i$ of π, and $\mathcal{K} \vdash Q$ denotes $at(\mathcal{K}) \vdash Q$ where $at(\mathcal{K})$ is the set of atoms of the derived facts of $\mathcal{K}$ (i.e., ignoring derivations).

Monotonicity is the property that for all steps $i \leq j$ of π and for every cyber-node x, $\mathcal{F}_x(c_i) \subseteq \mathcal{F}_x(c_j)$ and $\mathcal{G}_x(c_i) \subseteq \mathcal{G}_x(c_j)$. *Monotonicity* holds if no replacement rules are applied in π, because only replacement rules remove facts or goals from a node's state.

Soundness expresses that any derived fact appearing in an execution π is provable in Horn clause logic (with built-ins) from the previous observations. It

holds because derived atoms that appear in π are entailed by previous observations and controls of π, and entailment on derived atoms implies entailment in Horn clause logic.

Theorem 1 (Soundness). *For every step i of π, and for each $f : F \in \mathcal{F}(c_i)$, we have $\mathcal{F}^O(\pi_{|i}), \mathcal{G}^C(\pi_{|i}) \vdash f : F$, which in turn implies $\mathcal{F}^O(\pi_{|i}) \vdash F$.*

Proof. By Lemmas 1 and 2 below. □

Lemma 1 (Derivability implies provability). *If $f : F$ is a derived fact and $\mathcal{F}$ is the set of facts underlying the atomic subderivations of $f : F$ then $\mathcal{F} \vdash F$*

Proof. We show $\mathcal{F} \vdash F$ by cases on f. If $f : F$ is $B_\sigma(g) : \sigma(G)$, then $\vdash \sigma(G)$ by definition of derived facts. If $f : F$ is $\mathtt{O}(F) : F$ we have $\mathtt{O}(F) \vdash \mathtt{O}(F)$. If $f : F$ is $l_\sigma(f_1, \ldots, f_n, [g']) : \sigma(Q)$, with $l : P_1, \ldots, P_n \Rightarrow Q$ in Ω, then by induction we have $\mathcal{F} \vdash f_i : \sigma(P_i), 1 \leq i \leq n$ and $\mathcal{F} \vdash \sigma(Q)$, applying clause l. □

Lemma 2 (Derivations are derivable). *If $f : F \in \mathcal{F}(c_i)$ and $g : G \in \mathcal{G}(c_i)$, then $\mathcal{F}^O(\pi_{|i}), \mathcal{G}^C(\pi_{|i}) \vdash f : F$ and $\mathcal{F}^O(\pi_{|i}), \mathcal{G}^C(\pi_{|i}) \vdash g : G$.*

Proof. The proof is by induction on i. Note that $\mathcal{F}^O(\pi_{|i-1}), \mathcal{G}^C(\pi_{|i-1}) \vdash f : F$ implies $\mathcal{F}^O(\pi_{|i}), \mathcal{G}^C(\pi_{|i}) \vdash f : F$ (monotonicity of $\vdash$). We only need to consider rules r_i that introduce a new derived fact $f : F$ or goal $g : G$ at some cyber-node x. There are five cases for facts and four for goals. Here we show a few cases to illustrate the arguments (see [8] for the full proof).

(Observation) $f : F$ is $\mathtt{O}(F) : F$, which is in $\mathcal{F}^O(\pi_{|i})$.

(Forward1) $f : F$ is $l_\sigma(f_1, \ldots, f_n) : \sigma(Q)$, $l : P_1, \ldots, P_n \Rightarrow Q \in \Omega_f$, $f_j : \sigma(P_j) \in \mathcal{F}(c_{i-1})$, $1 \leq j \leq n$. By induction $\mathcal{F}^O(\pi_{|i-1}), \mathcal{G}^C(\pi_{|i-1}) \vdash f_j : F_j$ for $1 \leq j \leq n$ and so $\mathcal{F}^O(\pi_{|i}), \mathcal{G}^C(\pi_{|i}) \vdash f : F$.

(Forward2) $g : G$ is $l_\sigma^{-1}(f_1, \ldots, f_{j-1}) : \sigma(P_j), l : P_1, \ldots, P_n \Rightarrow Q \in \Omega_f$, and $f_k : \sigma(P_k) \in \mathcal{F}(c_{i-1})$, $1 \leq k < j$. By induction $\mathcal{F}^O(\pi_{|i-1}), \mathcal{G}^C(\pi_{|i-1}) \vdash f_k : F_k$ for $1 \leq k < j$ and thus $\mathcal{F}^O(\pi_{|i}), \mathcal{G}^C(\pi_{|i}) \vdash g : G$. □

Note that *Monotonicity* and *Soundness* are independent of the specific theory; in particular, they hold for the robot theory.

Completeness gives conditions under which a fact provable in the logic will eventually be covered (either directly or by subsumption). These conditions include fairness conditions on executions and consistency conditions between the theory and the subsumption and replacement orderings.

Definition 1 (Weak Fairness). *A rule instance contains the parameters that determine whether a rule applies in a configuration and if so, what the result is. It is given by the rule name, the node(s), the clause label, substitution, and all derived facts or goals involved in the application. For example,* $\mathsf{Forward1}(x, l, \sigma, f_1 : \sigma(P_1), \ldots, f_n : \sigma(P_n), l_\sigma(f_1, \ldots, f_n) : \sigma(Q))$ *represents an instance of the first forward rule. A rule instance ρ is* permanently applicable *in π at i iff ρ is applicable*

to c_j for $j \geq i$. An execution is logically fair *iff each instance of a* reasoning rule*, i.e., either a built-in, forward, or backward rule, that is permanently applicable at i is applied at some $j \geq i$. Similarly, an execution is* replacement fair *iff each instance of a* replacement rule *that is permanently applicable at i is applied at some $j \geq i$. An execution is* communication fair *iff each instance of a* communication *rule that is permanently applicable at i is applied at some $j \geq i$. An execution is* globally fair *iff it is logically, replacement, and communication fair.*

Definition 2 (Subsumption Preservation). *We say subsumption is preserved iff whenever $K_i \leq K_i'$ and $K_1, \ldots, K_n \vdash_1 K$, then there exists K' such that $K_1', \ldots, K_n' \vdash_1 K'$ and $K \leq K'$ (recall that K ranges over derived atoms).*

Definition 3 (Replacement Conditions). *Replacement is restricted iff the following conditions hold:* ***(1)*** *If $K_1 \prec K_2$, then $K_2 \not\rhd^+ K_1$.* ***(2)*** *If $K_1 \prec K_2$, $K_1 \not\rhd^+ K_2$ and $K_1 \not\leq K_2$, then there exists atomic K_1', K_2' such that $K_1' \rhd^* K_1$, $K_2' \rhd^* K_2$ and $K_1' \prec K_2'$.* ***(3)*** *If $K_1 \prec K_2$, $K_1 \rhd^+ K_2$, $K_1 \rhd^+ K_3$, $K_3 \not\rhd^+ K_2$, and $K_2 \not\rhd^+ K_3$, then $K_3 \leq K_2$.* ***(4)*** *If $K_1 \leq K_2$ and there is an atomic $K_2' \rhd^* K_2$ with $K_2' \prec K$, then there is an atomic $K_1' \rhd^* K_1$ with $K_1' \prec K$.*

We say that a derived fact $f : F$ is *eventually covered in* π there is some i and $f' : F' \in \mathcal{F}(c_i)$ such that $f : F \leq f' : F'$. The essence of completeness is that if $\mathcal{F} \vdash F$ for a subset of the observed facts of an execution, then some derivation of F will be eventually covered in the execution. The completeness theorem statement refines this, beginning with sufficient constraints for completeness to hold. The statement is broken into two parts, first showing provability implies derivability, and second showing that if a derived fact $f : F$ is entailed by subset of the observations of an execution, $f : F$ will eventually be covered. This is further split into two cases depending whether the final rule in the Horn clause derivation is a forwards or backwards rule. This is needed to account for the requirement that there must be a goal that unifies with a backwards rule conclusion before the rule can be applied, and thus in the backwards case, the theorem only applies to instances of goals.

Theorem 2 (Completeness). *Let π be a logically and communication fair execution, and let $\mathcal{F} \subseteq \mathcal{F}^O(\pi)$ and $\mathcal{G} \subseteq \mathcal{G}^C(\pi)$ be such that each element in $\mathcal{F} \cup \mathcal{G}$ is maximal in $\mathcal{F}^O(\pi) \cup \mathcal{G}^C(\pi)$ w.r.t. the replacement ordering. Assume subsumption is preserved, upward well-founded, and that replacement is restricted. If $at(\mathcal{F}) \vdash_{\mathrm{f}} F$ then there exists a derived fact $f : F$ such that $\mathcal{F} \vdash f : F$, which in turn implies that $f : F$ is eventually covered in π. If $G \in at(\mathcal{G})$ and $at(\mathcal{F}) \vdash_{\mathrm{b}} \sigma(G)$ then there exists a derived fact $f : \sigma(G)$ such that $\mathcal{F}, \mathcal{G} \vdash f : \sigma(G)$, which in turn implies that $f : \sigma(G)$ is eventually covered in π. Here $\vdash_{\mathrm{f}}$ ($\vdash_{\mathrm{g}}$) denote Horn clause derivability where the last clause applied is from Ω_f (Ω_g).*

Proof. As for soundness the proof has two parts: (a) showing that entailment in the Horn logic sense implies entailment in derived-atom sense, and (b) showing that a derived-atom derivable from the observed facts and injected (control) goals will eventually be covered in an execution. The proof of (a) is similar

to the proof of Lemma 1. The proof of (b) is structured using cases from the definition of replacement restriction. Maximality of the observed facts is needed as part of dealing with replacement rules. For details we refer to [8]. □

Completeness implies that all solutions for a goal are eventually generated, which is not always a desirable property in practice. For instance, in our robot example, the specification states that the user interest is satisfied as soon as one suitable snapshot is available, and further snapshots (and related activities) can be suppressed by means of the replacement ordering. Specifically, consider a situation where one goal $TakeSnapshot(t_I, t_T, n_D, t_D, A, I)$ leads to multiple $Snapshot(t_I, t_T, n_D, t_S, a, i)$ facts. Suppose there are two *Snapshot* facts; the logic will solve the *Result* goal with the first and discard both using replacement. In this execution one *Snapshot* fact will be ignored, but there is another execution where it is not.

Termination constrains the local inference system to avoid infinite regression in the attempt to achieve a goal. To state the theorem we need to define the finite closure property for a set of derived atoms, which by the correspondence between Horn clause derivability and the derivability relation on derived atoms is in fact a constraint on the Horn clause theory. We use a special case of the general definition for simplicity.

Definition 4 (Finite Closure). *We say that a set $\mathcal{F} \cup \mathcal{G}$ of derived facts and goals has the* finite closure property *iff there exists a well-founded quasi-order $(\mathcal{K}, \leq)$ such that $\mathcal{F} \cup \mathcal{G} \subseteq \mathcal{K}$, for each induced equivalence class $\mathcal{K}'$ the projection on atoms $at(\mathcal{K}')$ is finite, and the following conditions are satisfied:* ***(0)*** *If $g : G \in \mathcal{K}$ is a built-in goal and $\vdash \sigma(G)$ then $\mathtt{B}_\sigma(g) : \sigma(G) \in \mathcal{K}$ and $\mathtt{B}_\sigma(g) : \sigma(G) \leq g : G$.* ***(1)*** *If $l : P_1, \ldots, P_n \Rightarrow Q$ in Ω_f and $\mathcal{K} \vdash f_1 : \sigma(P_1), \ldots, f_n : \sigma(P_n)$, then $l_\sigma(f_1, \ldots, f_n) : \sigma(Q) \in \mathcal{K}$, and $f_i : \sigma(P_i) \in \mathcal{K}$ implies $l_\sigma(f_1, \ldots, f_n) : \sigma(Q) \leq f_i : \sigma(P_i)$ for $1 \leq i \leq n$.* ***(2)*** *If $l : P_1, \ldots, P_n \Rightarrow Q$ in Ω_f with a goal $\sigma(P_j)$ and $\mathcal{K} \vdash f_1 : \sigma(P_1), \ldots, f_{j-1} : \sigma(P_{j-1})$, then $l_\sigma^{-1}(f_1, \ldots, f_{j-1}) : \sigma(P_j) \in \mathcal{K}$, $f_i : \sigma(P_i) \in \mathcal{K}$ implies $l_\sigma^{-1}(f_1, \ldots, f_{j-1}) : \sigma(P_j) \leq f_i : \sigma(P_i)$ for $1 \leq i < j$.* ***(3)*** *If $l : P_1, \ldots, P_n \Rightarrow Q$ in Ω_b and $\mathcal{K} \vdash f_1 : \sigma(P_1), \ldots, f_n : \sigma(P_n)$, and $g' : G' \in \mathcal{K}$ with $\sigma(Q) = \sigma(G')$, then $l_\sigma(f_1, \ldots, f_n; g') : \sigma(Q) \in \mathcal{K}$, and $f_i : \sigma(P_i) \in \mathcal{K}$ implies $l_\sigma(f_1, \ldots, f_n; g') : \sigma(Q) \leq f_i : \sigma(P_i)$ for $1 \leq i \leq n$.* ***(4)*** *If $l : P_1, \ldots, P_n \Rightarrow Q$ in Ω_b with a goal $\sigma(P_j)$ and $\mathcal{K} \vdash f_1 : \sigma(P_1), \ldots, f_{j-1} : \sigma(P_{j-1})$, and $g' : G' \in \mathcal{K}$ with $\sigma(Q) = \sigma(G')$, then $l_\sigma^{-1}(f_1, \ldots, f_{j-1}; g') : \sigma(P_j) \in \mathcal{K}$, $l_\sigma^{-1}(f_1, \ldots, f_{j-1}; g') : \sigma(P_j) \leq g' : G'$, and $f_i : \sigma(P_i) \in \mathcal{K}$ with $i < j$ implies $l_\sigma^{-1}(f_1, \ldots, f_{j-1}; g') : \sigma(P_j) \leq f_i : \sigma(P_i)$ for $1 \leq i < j$.*

Intuitively, the set $\mathcal{K}$ over-approximates the set of all derived facts and goals that could be generated in response to an element from this set. Condition (0) corresponds to the built-in rule, conditions (1) and (2) correspond to the forward rules (which can be applied to solutions of goals), and conditions (3) and (4) correspond to the backward rules. We note that $\mathcal{K}$ may be infinite, but due to the use of most general substitutions σ in the proof rules, only a finite subset of $\mathcal{K}$ will be generated in any actual execution.

Theorem 3 (Termination). *If $\mathcal{F}^O(\pi) \cup \mathcal{G}^C(\pi)$ is finite and has the finite closure property then π is terminating, that is, either π is finite or there is some n such that r_i is the sleep rule for all $i > n$.*

Proof. Define depth $d(K)$ of a derived fact or goal K such that if $K \in \mathcal{K}$ then there is a descending $\lessdot$-chain in $\mathcal{K}$ of length $d(K)$, where $\lessdot$ is the relation inductively generated by the conditions (0)–(4) above (replacing $\leq$ by $\lessdot$). We then argue (a) that if π is nonterminating then due to the freshness condition of the proof rules the set of derived facts and goals grows without bound; and (b) that there is a finite bound on the set of facts and goals of a given finite depth. This means that in a nonterminating proof there is a descending $\lessdot$-chain and hence a descending $<$ chain that grows without bound, which contradicts well-foundedness. For details we refer to [8]. □

Our robot theory does satisfy the conditions for termination. Intuitively, the cases to check involve recursive calls: F3, B3, B9. Recursive calls using the clause F3, axiomatizing commutativity, lead to cycles with two facts in the equivalence class for any pair of areas. Calls to B9 will terminate because the argument W decreases on each until it reaches the lower bound b_w. The recursive call in B3 will never happen, by freshness constraints, but even without freshness the recursive call results in an equivalent derived fact.

Theorem 4 (Confluence). *If π is a globally fair and terminating execution then π is* confluent*, i.e., there exists a suffix π' such that $\mathcal{F}_x(c) = \mathcal{F}_y(c)$ and $\mathcal{G}_x(c) = \mathcal{G}_y(c)$ for all cyber-nodes x, y and $c \in \pi'$.*

Proof. It is easy to see that in a globally fair and terminating system, the replacement and communication rules will eventually ensure that all cyber-nodes will reach the same logical state (disregarding time and name) after no new knowledge is produced by reasoning rules.

5 Related Work

Knowledge sharing is a well-known idea that has been investigated by Halpern in [6] and in much subsequent work. Understanding knowledge sharing in distributed environments has led to a complementary view providing new insights into distributed algorithms and a logical justification for their fundamental limitations. For instance, attaining common knowledge, i.e., complete knowledge about the knowledge of other agents (and hence about the global state) in a distributed system is not feasible in a strict sense, and hence problems such as coordinated attack are unsolvable in asynchronous systems. In practice, approximations of common knowledge can be used by making assumptions of (sufficient) synchrony, but the fundamental problem in asynchronous systems remains. Halpern's concept of knowledge is based on modal logic, which expresses facts and the state of knowledge of individual agents. A key axiom is the knowledge axiom, which states that if an agent knows a fact, it must be true. Such an axiom is problematic in a distributed setting without a global view of the

world. Furthermore, such logics do not deal with changes in the facts during the reasoning process, or the ability to discard facts that are no longer useful, nor do they take goals into account.

The idea of applying declarative techniques in communication and networking is not new. A large body of work exists in the areas of specification, analysis, and synthesis of networking policies and protocols, e.g., in the context of security, routing, or dynamic spectrum access. Declarative querying of sensor networks has been studied through several approaches, for instance in [13], which composes services on the fly and in a goal-driven fashion using a concept of semantic streams. Declarative techniques to specify destinations have been used in disruption-tolerant networking [2]. A variant of Datalog has been applied to the declarative specification of peer-to-peer protocols [9]. Based on this work, [3] develops a very interesting approach to declarative sensor networks that can transmit generated facts to specific neighbors and can also utilize knowledge about neighbors to specify, e.g., routing algorithms. The idea of providing an abstraction that views a system as a single asset (an ensemble) rather then programming its individual components has been explored in several projects. Most interesting, the approach in Meld [1] extends the ideas from declarative sensor networks to modular robots, i.e., ensembles of robots with inter-robot communication limited to immediate neighbors. As an example, the movement of a composite robot emerges as a result of the coordinated interaction between its homogeneous robot modules. Most of the existing work focuses not on the theoretical foundations, but on efficient compilation into a conventional programming language. Another approach is the use of an efficient reasoning engine in embedded systems such as software-defined radios [5] or routers [12] as explored in the context of disruption-tolerant networking.

6 Conclusion and Future Directions

We have presented first steps toward combining local forward and backward reasoning in a fully distributed fashion with knowledge that is transparently shared. A fixed or known neighborhood is not assumed in our more abstract approach, and the use and dissemination of both facts and goals aims at general cyber-physical systems with distributed actuation, and hence leads us beyond sensor networks, in particular to dynamic sensor/actuator networks that are, unlike ensembles, inherently heterogeneous.

The partial order structure of knowledge enables distributed knowledge sharing and replacement. The subsumption relation has a logical interpretation, which in a sufficiently expressive logic can be defined in terms of a logical implication. The replacement ordering, on the other hand, allows the user to specify when knowledge becomes obsolete. The use of knowledge is not limited to facts; knowledge can also represent goals. We do not use a modal logic, which means that knowledge about knowledge must be explicitly represented if needed.

We have developed a prototype of our distributed logical framework based on an implementation of the partially ordered knowledge-sharing model and an application programming interface (API) for cyber-physical devices that enables interaction with the physical world [7]. Our framework provides a uniform

abstraction for a wide range of NCPS applications, especially those concerned with distributed sensing, optimization, and control. Key features of our framework are that (i) it provides a generic service to represent, manipulate, and share knowledge across the network under minimal assumptions on connectivity, (ii) it enables the same application code to be used in various environments including simulation models and real-world deployments, (iii) it adapts to a wide range of operating points between autonomy and cooperation to overcome limitations in connectivity and resources, as well as uncertainties and failures.

The proof system that we have presented in this paper focuses on a few core ideas, but the work can be generalized in many directions. One step is the generalization of the underlying logic, for example, incorporation of equational features as in Maude [4]. Another possibility is introducing stochastic events and/or probablistic reasoning.

The logical framework should be thought of as a means of expressing the space of logically sound behaviors, which can be further constrained by more quantitative techniques. Our inference rules force a proof strategy that proceeds according to the ordering of atoms in the conditions of a Horn clause. More general proof strategies are possible, and could potentially lead to a higher degree of parallelism. Solved or unsolved goals that cannot generate further solutions could be removed — for example, by equipping goals with an expiration time to allow removal in a controlled manner. Several conflicting goals can be active at the same time, and strategies guided by prioritization and more generally distributed optimization techniques need to be developed.

This paper presents an interleaving semantics, but a true concurrency semantics, such as the semantics of rewriting logic [10], where the concurrent application of proof rules is represented explicitly, might be more appropriate.

In this paper derivations are used for meta-level reasoning. However, the explicit representation of derivations could be made available to applications. Possible uses include the following.

(1) *Faulty Facts Elimination.* If the initial sensor data (e.g., noise detected in area A) is wrong — for example, due to a faulty or malicious sensor — and is detected, this (meta) fact should be disseminated to other robots and the inference system should exclude reasoning based on faulty data.

(2) *Situation Awareness.* Noise was correctly detected, but ceases when one of the robots arrives in area A. In this case, new facts will be disseminated and decisions based on the obsolete facts might need to be canceled.

(3) *Uncertainty Management.* Derivations can be used to indicate whether a decision was made based on reliable information. If a decision is made based on an uncertain observation (e.g., a sensor with some error margin), the degree of uncertainty needs to be propagated through the derivation so that decisions can be based on the quality of derivations as well as the conclusions.

(4) *Post-Examination.* After a goal is satisfied, one can examine whether further optimization is possible (e.g., in terms of delay, energy consumption). For example, one can examine why a certain robot decided to move in a certain direction. This can be related to (3) if the less optimal decision was made due to data

uncertainty that was not correctly evaluated (e.g., data fusion from two sensors with equal weight is suboptimal if one of the sensors has a larger error).

Acknowledgments Support from National Science Foundation Grant 0932397 (A Logical Framework for Self-Optimizing Networked Cyber-Physical Systems) and Office of Naval Research Grant N00014-10-1-0365 (Principles and Foundations for Fractionated Networked Cyber-Physical Systems) is gratefully acknowledged. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF or ONR.

References

1. Michael P. Ashley-Rollman, Seth Copen Goldstein, Peter Lee, Todd C. Mowry, and Padmanabhan Pillai. Meld: A declarative approach to programming ensembles. In *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS '07)*, October 2007.
2. P. Basu, R. Krishnan, and D. W. Brown. Persistent delivery with deferred binding to descriptively named destinations. In *Proc. of IEEE MILCOM*, 2008.
3. David Chu, Lucian Popa, Arsalan Tavakoli, Joseph M. Hellerstein, Philip Levis, Scott Shenker, and Ion Stoica. The design and implementation of a declarative sensor network system. In *SenSys '07: Proc. of the 5th International Conference on Embedded Networked Sensor Systems*, pages 175–188, 2007.
4. Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, and Carolyn L. Talcott, editors. *All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic*, volume 4350 of *LNCS*. Springer, 2007.
5. Daniel Elenius, Grit Denker, and Mark-Oliver Stehr. A semantic web reasoner for rules, equations and constraints. In *RR '08: Proc. of the 2nd International Conference on Web Reasoning and Rule Systems*, pages 135–149. Springer, 2008.
6. Joseph Y. Halpern and Yoram Moses. Knowledge and common knowledge in a distributed environment. *Journal of the ACM*, 37:549–587, 1984.
7. Minyoung Kim, Mark-Oliver Stehr, Jinwoo Kim, and Soonhoi Ha. An application framework for loosely coupled networked cyber-physical systems. In *8th IEEE Intl. Conf. on Embedded and Ubiquitous Computing (EUC-10)*, 2010.
8. Minyoung Kim, Mark-Oliver Stehr, and Carolyn Talcott. A distributed logic for networked cyber-physical systems (extended version). In preparation., November 2010.
9. Boon Thau Loo, Tyson Condie, Minos Garofalakis, David E. Gay, Joseph M. Hellerstein, Petros Maniatis, Raghu Ramakrishnan, Timothy Roscoe, and Ion Stoica. Declarative networking. *Commun. ACM*, 52(11):87–95, 2009.
10. J. Meseguer. Conditional Rewriting Logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73–155, 1992.
11. Mark-Oliver Stehr, Minyoung Kim, and Carolyn L. Talcott. Toward distributed declarative control of networked cyber-physical systems. In *Ubiquitous Intelligence and Computing - 7th International Conference, UIC 2010, Xi'an, China, October 26-29, 2010. Proceedings*, volume 6406 of *LNCS*, pages 397–413. Springer, 2010.
12. Mark-Oliver Stehr and Carolyn Talcott. Planning and learning algorithms for routing in disruption-tolerant networks. In *Proc. of IEEE MILCOM*, 2008.
13. Kamin Whitehouse, Feng Zhao, and Jie Liu. Semantic streams: A framework for composable semantic interpretation of sensor data. In *Proc. of the European Workshop on Wireless Sensor Networks*, pages 5–20. EWSN, 2006.