Collaborative Planning with Privacy

Protocol eXchange

May 7, 2007

Max Kanovich\textsuperscript{1}, Paul Rowe\textsuperscript{2}, Andre Scedrov\textsuperscript{2}

\textsuperscript{1}Queen Mary, University of London
\textsuperscript{2}University of Pennsylvania
Many examples of collaboration

- Between distributor and retailer
- Between hospitals and insurance companies
- Distributed databases
- Social networking sites (MySpace, Facebook)

Temporary alignment of interests

Information sharing is necessary to collaborate, but full disclosure is not desired.
Our Work

- We provide a model of collaboration at an abstract level.

- We can model a large class of collaborations while being able to make conclusions about privacy.

- Focus is on the interplay between protecting and releasing information.
Our Work

- We draw on
  - Planning from AI literature
  - State transition systems and multiset rewriting

- We consider systems with well-balanced actions

- It is PSPACE-complete to decide the existence of a collaborative plan, and if the system preserves the privacy of all agents.
Outline

- Motivations from Classical Planning
- Our formalism: Local state transition systems
- Privacy in collaboration
- Complexity results and foundation in logic
- Related and future work
Classical Planning

- A robot manipulating its environment
- Description of the environment
  - Objects
  - Relations between the objects
- Actions
- Initial configuration
- Goal configuration
Initial State

\{ONTABLE(A), ON(B,A), CLEAR(B), ONTABLE(C), CLEAR(C), HANDEMPTY\}
Actions

- **take(𝑥):** \{HANDEMPLOYEE, CLEAR(𝑥), ONTABLE(𝑥)\}
  \[\rightarrow\] \{HOLDS(𝑥)\}

- **remove(𝑥,𝑦):** \{ON(𝑥,𝑦), HANDEMPLOYEE, CLEAR(𝑥)\}
  \[\rightarrow\] \{HOLDS(𝑥), CLEAR(𝑦)\}

- **stack(𝑥,𝑦):** \{HOLDS(𝑥), CLEAR(𝑦)\}
  \[\rightarrow\] \{HANDEMPLOYEE, CLEAR(𝑥), ON(𝑥,𝑦)\}

- **put(𝑥):** \{HOLDS(𝑥)\}
  \[\rightarrow\] \{ONTABLE(𝑥), CLEAR(𝑥), HANDEMPLOYEE\}
Blocks World: Plan

- remove(B,A)
- put(B)
- take(A)
- stack(A,C)
Outline

- Motivations from Classical Planning
- *Our formalism: Local state transition systems*
- Privacy in collaboration
- Complexity results and foundation in logic
- Related and future work
Collaboration and Planning

- Multiple agents: $A_1, \ldots, A_n$
- Each has private data $P_A(t)$ and public data $P'(u)$
- Each has a set of actions
- Initial state
- Goal state
- Find a sequence of actions leading from initial state to goal state
A local state transition system is a triplet

\[ T = (\Sigma, I, R_T) \]

where

- \( \Sigma \) is a signature of predicate symbols and terms
  (currently only constants and variables)
- \( I \) is a set of agents
- \( R_T \) is a set of (local) actions
Local State Transition Systems

- A **fact** is a closed atomic predicate over multi-sorted terms
- A syntactic convention distinguishes between private and public facts:

  Private   Public/Group
  \( P_A(t) \)   \( P'(u) \)
Security Labels

Figure 1: Security Levels.
System Configurations

- A *state* or *configuration* of the system is a multiset of private and public facts

\[ X_{A_1'}, X_{A_2'}, ..., X_{A_n'}, X' \]

- Each agent can affect only their own private data and the public data.
Actions

- Replace $X_A$ and $X'$ by $Y_A$ and $Y'$
  \[ r : X_A X' \rightarrow A Y_A Y' \]

Transforms $W = VX_A X'$ into $U = VY_A Y'$

- System transformation is written as
  \[ W \underset{r}{\rightarrow} U \]

- Reachability from a set $R$ of actions is denoted by
  \[ W \underset{R^*}{\rightarrow} U \]
Partial Goals

- The goal need not describe the complete configuration.

- Partial reachability is defined by

\[ W \overset{\mathbf{A}}{\underset{\mathbf{R}}{\rightarrow}} Z \quad \text{iff} \quad W \overset{\mathbf{B}}{\underset{\mathbf{R}}{\rightarrow}} ZU \quad \text{for some} \quad U \]

So with \( r : X \Rightarrow A Y \Rightarrow A Y' \) we find that

\[ UX \overset{\mathbf{A}}{\underset{\mathbf{R}}{\rightarrow}} r Y \Rightarrow A Y' \]
Collaborative Plans

A **collaborative plan** based on the action set $R$ which leads from $W$ to the partial goal $Z$ is a labeled, non-branching tree satisfying:

1. Edges are labeled with actions from $R$, and nodes are labeled with states.
2. The label of each node enables the label of the outgoing edge.
3. The label of the root is $W$.
4. The label of the leaf is $ZU$ for some $U$.

There exists a collaborative plan based on $R$, leading from $W$ to the partial goal $Z$ if and only if $W \ast_R^* Z$. 
Abstract Example

Alice’s actions include

\[ r_1 : P_A(t) \rightarrow A P_A(t)P'(t') \]
\[ r_2 : P_A(t) \rightarrow A P_A(t)P''(t'') \]

Bob’s actions include

\[ r_3 : Q_B(u)Q'(v)P'(t')P''(t'') \rightarrow B Q_B(t)Q'(v') \]

When \( R = \{r_1, r_2, r_3\} \) then

\[ P_A(t)Q_B(u)Q'(v) * R Q_B(t) \]
An Example

Example:

\[ r_1 : P_A(15\_A\_Pwd) S_A(7\_B\_Share) \rightarrow_A \]
\[ P_A(21\_A\_Pwd) S_A(7\_B\_Share) P'(8\_A\_Share) \]

\[ r_2 : Q_B(7\_B\_Share) P'(8\_A\_Share) \rightarrow_B Q_B(15\_A\_Pwd) \]
An Example

Example:

- $r_1 : P_A(15\_A\_Pwd) \rightarrow_A S_A(7\_B\_Share)$
  
  $P_A(21\_A\_Pwd) S_A(7\_B\_Share) \rightarrow_A P'(8\_A\_Share)$

- $r_2 : Q_B(7\_B\_Share) \rightarrow_B Q_B(15\_A\_Pwd)$

$P_A(15\_A\_Pwd) S_A(7\_B\_Share) Q_B(7\_B\_Share)$
An Example

- Example:
  - $r_1 : P_A(15_A_Pwd) S_A(7_B_Share) \rightarrow_A P_A(21_A_Pwd) S_A(7_B_Share) P'(8_A_Share)$
  - $r_2 : Q_B(7_B_Share) P'(8_A_Share) \rightarrow_B Q_B(15_A_Pwd)$

$P_A(15_A_Pwd) \cdot S_A(7_B_Share) \cdot Q_B(7_B_Share) \cdot r_1$

$P_A(21_A_Pwd) \cdot S_A(7_B_Share) \cdot P'(8_A_Share) \cdot Q_B(7_B_Share)$
An Example

Example:

\[ r_1 : P_A(15\_A\_Pwd) S_A(7\_B\_Share) \rightarrow_A P_A(21\_A\_Pwd) S_A(7\_B\_Share) P'(8\_A\_Share) \]

\[ r_2 : Q_B(7\_B\_Share) P'(8\_A\_Share) \rightarrow_B Q_B(15\_A\_Pwd) \]
An Example

- Example:
  - $r_1 : P_A(15\_A\_Pwd) S_A(7\_B\_Share) \rightarrow_A P_A(21\_A\_Pwd) S_A(7\_B\_Share) P'(8\_A\_Share)$
  - $r_2 : Q_B(7\_B\_Share) P'(8\_A\_Share) \rightarrow_B Q_B(15\_A\_Pwd)$

- In this case we see

\[
P_A(15\_A\_Pwd) S_A(7\_B\_Share) Q_B(7\_B\_Share) \cdot r_1
\]
\[
P_A(21\_A\_Pwd) S_A(7\_B\_Share) P'(8\_A\_Share) Q_B(7\_B\_Share) \cdot r_2
\]
\[
P_A(21\_A\_Pwd) S_A(7\_B\_Share) Q_B(15\_A\_Pwd)
\]

\[
\ast \cdot R^* Q_B(15\_A\_Pwd)
\]
Outline

- Motivations from Classical Planning
- Our formalism: Local state transition systems
- Privacy in collaboration
- Complexity results and foundation in logic
- Related and future work
Privacy Concerns

- If Alice starts with a secret term $t$, she wants to make sure it stays secret.

- Protect the secret from all possible behavior of other participants.

- Requires a global condition on reachable configurations.
Privacy Condition

- Local state transition system in initial configuration $W$, protects the privacy of agent A if every term $t$ which, in the initial configuration $W$, occurs only in private predicates of A, also occurs only in private predicates of A in any reachable configuration.

- Partial goals of the form $Q'(t)$ or $Q_B(t)$ are not reachable from the initial configuration.
Remarks on Privacy

- Local state transition systems define a space of plans or protocols.

- Privacy condition is global condition on entire space.

- Other participants may be viewed as a type of adversary.

- Provides a guarantee that if others don’t follow plan, or perform extra local computations then secrets are not revealed.
Remarks on Privacy

- Can express notions of knowledge of *current* information.

- Alice’s action may change her password, rendering the old password obsolete.

- Knowledge of old password without knowledge of *current* password may be useless.
The Collaborative Planning Problem with Privacy

Given a local state transition system, and given an initial state $W$ and a partial goal $Z$, does there exist a plan which leads from $W$ to $Z$, and does the system protect the privacy of all agents?
Well-Balanced Actions

- Actions are restricted to have the same number of facts in the pre- and post-conditions.

- Intuitively, actions serve to update fields and they do not create new ones.

- Introduce a special constant symbol to indicate an empty field: $P(*)$

- Not as restrictive as it seems.
Example Revisited

Example:

- \( r_1 : P_A(15\_A\_Pwd) \ S_A(7\_B\_Share) \ P'(\ast) \rightarrow_A P_A(15\_A\_Pwd) \ S_A(7\_B\_Share) \ P'(8\_A\_Share) \)

- \( r_2 : Q_B(7\_B\_Share) \ P'(8\_A\_Share) \rightarrow_B Q_B(15\_A\_Pwd) \ P'(\ast) \)

\[
P_A(15\_A\_Pwd) \ S_A(7\_B\_Share) \ P'(\ast) \ Q_B(7\_B\_Share) \ \ast \ r_1
\]
\[
P_A(15\_A\_Pwd) \ S_A(7\_B\_Share) \ P'(8\_A\_Share) \ Q_B(7\_B\_Share) \ \ast \ r_2
\]
\[
P_A(15\_A\_Pwd) \ S_A(7\_B\_Share) \ P'(\ast) \ Q_B(15\_A\_Pwd) \]

We still find that

\[
P_A(15\_A\_Pwd) \ S_A(7\_B\_Share) \ P'(\ast) \ Q_B(7\_B\_Share) \ \ast \ R^* \ Q_B(15\_A\_Pwd) \]
Outline

- Motivations from Classical Planning
- **Our formalism: Local state transition systems**
- Privacy in collaboration
- **Complexity results and foundation in logic**
- Related and future work
Complexity Results

- The Collaborative Planning Problem with Privacy, with well-balanced actions, is \textit{PSPACE-complete}.
- It is polynomial with respect to the following parameters:
  - The size of a program recognizing the actions
  - The number of facts in the initial configuration
  - The number of closed facts in the (finite) signature
Complexity Results

- For a *fixed* finite signature, the Collaborative Planning Problem with privacy, with well-balanced actions, is solvable in polynomial time.
  - It is polynomial with respect to the parameters:
    - The size of a program recognizing the actions
    - The number of facts in the initial configuration
  
(The number of closed facts in the signature is now viewed as a constant.)
Linear logic is a resource-sensitive refinement of traditional logic.

Linear implication mimics actions well by “consuming” antecedents.

We translate local state transition systems into a variant of linear logic called affine logic.
Logical Foundation

**Theorem**: Local state transition systems are sound and complete with respect to (our translation into) affine logic.

**Benefits include**:
- Possible insights from well established formalism
- Use of already existing tools
Outline

- Motivations from Classical Planning
- Our formalism: Local state transition systems
- Privacy in collaboration
- Complexity results and foundation in logic
- Related and future work
Related Work


- **Security Policies and Security Models** [J. A. Goguen and J. Meseguer ’82]

- **Conditional Rewriting Logic as a Unified Model of Concurrency** [J. Meseguer ’92]
Related Work

- **Enforcing Robust Declassification and Qualified Robustness** [A. C. Myers, A. Sabelfeld, S. Zdancewic 2004]

Future Work

- Extend to a richer language of functional terms.
- Explore the use of existentials in affine logic to model fresh values.
- Investigate behavior in the presence of actions with nondeterministic effects.
- Determine if our formalism provides traceability.
Future Work

- Investigate more completely the ability to distinguish between obsolete and current secrets.

- Explore the use of utility functions weighing the relative importance of protecting or releasing information.

- Explore a more complicated structure for security labels.
Summary

- Introduced local state transition systems
- Discussed notions of privacy in collaboration
- Formalized the collaborative planning problem with privacy
- Determined PSPACE-completeness in the well-balanced case
- Discussed foundation in logic
Thank You!
Collaborative Planning with Privacy

Protocol eXchange
May 7, 2007

Max Kanovich\textsuperscript{1}, Paul Rowe\textsuperscript{2}, Andre Scedrov\textsuperscript{2}

\textsuperscript{1}Quenn Mary, University of London
\textsuperscript{2}University of Pennsylvania