In Processes, We Believe!
Marrying Process Algebra and Epistemic Logic

Francien Dechesne and Mohammad Mousavi

TbILLC: Special Session on Logic, Information and Agency
1 Introduction: Operational vs Epistemic
Outline

1. Introduction: Operational vs Epistemic
2. Bridging the Gap: Specification Framework
Outline

1. Introduction: Operational vs Epistemic
2. Bridging the Gap: Specification Framework
3. Linking to the existing epistemic temporal framework of ISs
1. Introduction: Operational vs Epistemic
2. Bridging the Gap: Specification Framework
3. Linking to the existing epistemic temporal framework of ISs
4. Formal Results
Outline

1. Introduction: Operational vs Epistemic
2. Bridging the Gap: Specification Framework
3. Linking to the existing epistemic temporal framework of ISs
4. Formal Results
5. Conclusions
VEMPS project (2006-2009)
using epistemic logic for verification of security protocols
Background of the research

VEMPS project (2006-2009)
using epistemic logic for verification of security protocols

TU Eindhoven & CWI Amsterdam:
Yanjing Wang, Jan van Eijck, Francien Dechesne,
Erik de Vink, Simona Orzan, Mohammad Mousavi
### VEMPS project (2006-2009)

Using epistemic logic for verification of security protocols

TU Eindhoven & CWI Amsterdam:
Yanjing Wang, Jan van Eijck, Francien Dechesne, Erik de Vink, Simona Orzan, Mohammad Mousavi

(dynamic) epistemic logic

\[ \uparrow \downarrow \]

Process algebra, formal semantics, verification of distributed processes, security
The prototypical example: Dining Cryptographers
### Operational vs. Epistemic

**Approach**

<table>
<thead>
<tr>
<th>Approach</th>
<th>Protocol</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operational</td>
<td>Intuitive</td>
<td>Non-trivial; Difficult with Knowledge Properties</td>
</tr>
<tr>
<td>Epistemic</td>
<td>Laborious</td>
<td>Intuitive; combination of epistemic and temporal constructs</td>
</tr>
</tbody>
</table>
Operational Approach: basic process algebra

Simple PA: Syntax

\[
\begin{align*}
  a & ::= a[?, !](\vec{k}) \\
  p, q & ::= a | a; p | p + q | p || q
\end{align*}
\]

1. receive: \(a?(\vec{k})\), send: \(a!(\vec{k})\),
   individual actions or synchronizations: \(a(\vec{k})\);
Operational Approach: basic process algebra

Simple PA: Syntax

\[
\begin{align*}
a & ::= a[?, !][\vec{k}] \\
p, q & ::= a | a; p | p + q | p || q
\end{align*}
\]

1. receive: \(a?(\vec{k})\), send: \(a!(\vec{k})\),
   individual actions or synchronizations: \(a(\vec{k})\);

2. action prefixing: \(a; q\)
Operational Approach: basic process algebra

Simple PA: Syntax

\[
\begin{align*}
  a &::= a[?, !](\overrightarrow{k}) \\
  p, q &::= a \mid a; p \mid p + q \mid p \parallel q
\end{align*}
\]

1. receive: \(a?(\overrightarrow{k})\), send: \(a!(\overrightarrow{k})\),
   individual actions or synchronizations: \(a(\overrightarrow{k})\);
2. action prefixing: \(a; q\)
3. nondeterministic choice: \(p + q\) (or \(\sum_{i \in I} p_i\));
Operational Approach: basic process algebra

**Simple PA: Syntax**

\[
a \ ::= \ a[?, !](\overrightarrow{k}) \\
p, q \ ::= \ a \mid a; p \mid p + q \mid p \parallel q
\]

1. receive: \(a?(\overrightarrow{k})\), send: \(a!(\overrightarrow{k})\),
   individual actions or synchronizations: \(a(\overrightarrow{k})\);
2. action prefixing: \(a; q\)
3. nondeterministic choice: \(p + q\) (or \(\sum_{i \in I} p_i\));
4. parallel composition: \(p \parallel q\)
   where send and receive synchronize.

Dechesne & Mousavi
In Processes, We Believe!
Operational Approach: basic process algebra

Simple PA: Syntax

\[
\begin{align*}
    a & ::= a[?, !](\overrightarrow{k}) \\
    p, q & ::= a | a; p | p + q | p || q
\end{align*}
\]

1. receive: \( a?(\overrightarrow{k}) \), send: \( a!(\overrightarrow{k}) \),
   individual actions or synchronizations: \( a(\overrightarrow{k}) \);
2. action prefixing: \( a; q \)
3. nondeterministic choice: \( p + q \) (or \( \sum_{i\in I} p_i \));
4. parallel composition: \( p || q \)
   where send and receive synchronize.

Semantics:
labelled transition system generated by syntactic rules
(SOS: Structural Operational Semantics)
Simple PA: Intuitive Semantics

\[ \sum_{i \in \{\top, \bot\}} (a!(i) + b?(i)) \]
Operational Approach: basic process algebra

Simple PA: Intuitive Semantics

\[
\sum_{i \in \{\top, \bot\}} (a! (i) + b? (i)) \parallel (a? (\top) + b! (\bot))
\]
Our approach

We try to bridge the gap by specifying *action visibilities* in the process algebraic protocol specification. This creates the epistemic component in the behavior model.
We try to bridge the gap by specifying *action visibilities* in the process algebraic protocol specification. This creates the epistemic component in the behavior model.

Structural Operational Semantics: derived formally from PA-term syntax through set of rules.

We extend actions in a simple process algebra with identity-annotations.
We extend actions in a simple process algebra with identity-annotations.

To be able to capture different views of what happens.
Bridging the gap: our proposal

- We extend actions in a simple process algebra with identity-annotations
- To be able to capture different views of what happens
- SOS-rules generate an *Annotated* LTS
We extend actions in a simple process algebra with identity-annotations.

To be able to capture different views of what happens,

SOS-rules generate an **Annotated** LTS

which can be decomposed into LTS and Kripke model.
Bridging the gap: our proposal

- We extend actions in a simple process algebra with identity-annotations.
- To be able to capture different views of what happens.
- SOS-rules generate an *Annotated* LTS.
- Which can be decomposed into LTS and Kripke model.
- On these ALTSs we can check properties in our epistemic temporal language.
We extend actions in a simple process algebra with identity-annotations.

To be able to capture different views of what happens.

SOS-rules generate an Annotated LTS.

Which can be decomposed into LTS and Kripke model.

On these ALTSs we can check properties in our epistemic temporal language.

(Spoiler) these ALTSs are like Interpreted Systems.
Decorated actions:

$\left( J \right) \alpha$: action $\alpha$ is perceived as $\alpha$ by $i \in J$ and as $\rho(\alpha)$ by others.
Decorated actions:

\((J)\alpha: \text{action } \alpha \text{ is perceived as } \alpha \text{ by } i \in J \text{ and as } \rho(\alpha) \text{ by others.}\)

where

- \(J \subseteq Id: \text{the intended audience of } a\)
- \(\rho: A \rightarrow A \cup \{\tau\} \text{ a public appearance function}\)
- \(\tau: \text{the invisible action}\)
Decorated actions:

$\langle J \rangle \alpha$: action $\alpha$ is perceived as $\alpha$ by $i \in J$ and as $\rho(\alpha)$ by others.

where

- $J \subseteq Id$: the intended audience of $a$
- $\rho : A \rightarrow A \cup \{\tau\}$ a public appearance function
- $\tau$: the invisible action

Simple PA with views: Syntax

\[
\begin{align*}
    d &::= (J)a[?,!](k) \\
    p, q &::= d \mid d; p \mid p + q \mid p \parallel q
\end{align*}
\]
Decorated actions:

\((J)\alpha\): action \(\alpha\) is perceived as \(\alpha\) by \(i \in J\) and as \(\rho(\alpha)\) by others.

where

- \(J \subseteq Id\): the intended audience of \(a\)
- \(\rho: A \rightarrow A \cup \{\tau\}\) a public appearance function
- \(\tau\): the invisible action

Simple PA with views: Syntax

\[
\begin{align*}
d &::= (J)a[?, !](\xrightarrow{k}) \\
p, q &::= d \mid d; p \mid p + q \mid p || q
\end{align*}
\]

(⇒ \(\rho\) is now part of the protocol specification!)
Suppose \( \rho(b) = c \). Then in the left branch, \( \{3, \ldots, n\} \) see \( c \).
A Process Algebraic Framework with epistemics

Simple PA with views: example

Suppose $\rho(b) = c$. Then in the left branch, $\{3, \ldots, n\}$ see $c$. 
Formal semantics: operational part (Summarized)

\[ (a) \quad (d, \pi) \xrightarrow{d} \sqrt{\pi \bowtie d} \]

\[ (s0) \quad (x_0, \pi) \xrightarrow{d} (y_0, \pi') \]
\[ (x_0; x_1, \pi) \xrightarrow{d} (y_0; x_1, \pi') \]

\[ (s1) \quad (x_0, \pi) \xrightarrow{d} \sqrt{\pi'} \]
\[ (x_0; x_1, \pi) \xrightarrow{d} (x_1, \pi') \]

\[ (n0) \quad (x_0, \pi) \xrightarrow{d} (y_0, \pi') \]
\[ (x_0 + x_1, \pi) \xrightarrow{d} (y_0, \pi') \]

\[ (p0) \quad (x_0, \pi) \xrightarrow{d} (y_0, \pi') \]
\[ (x_0 \parallel x_1, \pi) \xrightarrow{d} (y_0 \parallel x_1, \pi') \]

\[ (p4) \quad (x_0, \pi) \xrightarrow{(J)?a} (y_0, \pi') \]
\[ (x_1, \pi) \xrightarrow{(J')!a} (y_1, \pi'') \]
\[ (x_0 \parallel x_1, \pi) \xrightarrow{(J \cup J')a} (y_0 \parallel y_1, \pi \bowtie (J \cup J')a) \]
Formal semantics: epistemic part (Summarized)

\[ \pi \overset{i}{\equiv} \pi \]
\[ \pi \mathrel{\bowtie} J(a) \overset{i}{\equiv} \pi \mathrel{\bowtie} J(a) \]
\[ \pi \cdot \cdot \cdot \pi' \mathrel{\bowtie} J(a) \overset{i}{\not\equiv} \pi \]
\[ \pi \mathrel{\bowtie} J(a) \overset{i}{\equiv} \pi \]

\[ \pi \overset{i}{\equiv} \pi' \quad i \not\in J \quad \rho(a) = \rho(b) \]
\[ \pi \mathrel{\bowtie} J(a) \overset{i}{\equiv} \pi' \mathrel{\bowtie} J(b) \]
\[ \pi \cdot \cdot \cdot \pi' \mathrel{\bowtie} J(a) \overset{i}{\not\equiv} \pi' \mathrel{\bowtie} J(a) \]
\[ \pi \overset{i}{\equiv} \pi' \mathrel{\bowtie} J(a) \]

Dechesne & Mousavi
In Processes, We Believe!
Seminal book:


Interpreted Systems as semantics for epistemic temporal logic.
Interpreted Systems

Seminal book:


Interpreted Systems as semantics for epistemic temporal logic.

Transition systems with rich states:

- global state is $n$-tuple of local states
- indistinguishability relations between global states generated on the basis of local state for each agent.
Agents $\mathcal{I} = \{1, \ldots, n\}$

Local states $L_i$, global states: $L = \prod_{i=1}^{n} L_i$

Run $r$: a sequence of global states

Protocol $R$: set of runs

Valuation function $\nu: L \rightarrow \Phi$

Indistinguishability $\mathcal{I} \approx \mathcal{I}'$ iff $l_i = l'_i$

Interpreted system: $(R, \nu)$

Our focus: protocol component (not valuation)
Agents $\mathcal{I} = \{1, \ldots, n\}$

Local states $L_i$, global states: $L = \prod_{i=1}^{n} L_i$

Run $r$: a sequence of global states

Protocol $R$: set of runs (note: given, not generated)

Valuation function $\nu: L \rightarrow \Phi$

Indistinguishability $\overrightarrow{I_i} \approx \overrightarrow{I'_i}$ iff $I_i = I'_i$

Interpreted system: $(R, \nu)$

Our focus: protocol component (not valuation)
Work in progress!
Work in progress!

We want to

- Link the process algebra specs to ISs
- Allow to transform results from one to the other:
  - PA as a syntax for generating ISs
  - exploited analysis tools available for IS
  - characterize semantic properties of classes of PA specs
Trace: sequence of decorated actions.
\([\_\_\_]_{aux}: CCSi \text{ process } \mapsto \text{ set of traces.}\)

\([0]_{aux} \overset{\cdot}{=} \langle \rangle\)
Trace: sequence of decorated actions.
$\llbracket \rightarrow \rrbracket_{aux}$: $CCSi$ process $\mapsto$ set of traces.

$\llbracket 0 \rrbracket_{aux} \equiv \langle \rangle$

$\llbracket d; p \rrbracket_{aux} \equiv d \circ \llbracket p \rrbracket_{aux}$
Trace: sequence of decorated actions.
\([ \_ ]_{aux} \): CCSi process \( \mapsto \) set of traces.

\[
\begin{align*}
[0]_{aux} & \equiv \langle \rangle \\
[d; p]_{aux} & \equiv d \bowtie [p]_{aux} \\
[p + q]_{aux} & \equiv [p]_{aux} \cup [q]_{aux}
\end{align*}
\]
Trace: sequence of decorated actions.
\[−\]_aux: CCSi process \(\mapsto\) set of traces.

\[
\begin{align*}
[0]_aux & \triangleq \langle \rangle \\
[d; p]_aux & \triangleq d \circ [p]_aux \\
[p + q]_aux & \triangleq [p]_aux \cup [q]_aux \\
[p \parallel q]_aux & \triangleq [p]_aux \parallel_{tr} [q]_aux
\end{align*}
\]

where \(\parallel_{tr}\) auxiliary function
Trace: sequence of decorated actions. 
\([\cdot]\)\_aux: CCS\_i process \leftrightarrow \text{set of traces}. 

\[
\begin{align*}
[0]_{aux} & \equiv \langle \rangle \\
[d; p]_{aux} & \equiv d \circ [p]_{aux} \\
[p + q]_{aux} & \equiv [p]_{aux} \cup [q]_{aux} \\
[p \parallel q]_{aux} & \equiv [p]_{aux} \mid tr [q]_{aux}
\end{align*}
\]

where \(\mid tr\) auxiliary function

\[
P \mid tr \emptyset \equiv \emptyset \mid tr P \equiv P
\]
Trace: sequence of decorated actions.  
\(\llbracket - \rrbracket_{aux} \mapsto \text{set of traces.}\)  

\[
\begin{align*}
\llbracket 0 \rrbracket_{aux} & \overset{\cdot}{=} \langle \rangle \\
\llbracket d; p \rrbracket_{aux} & \overset{\cdot}{=} d \uplus \llbracket p \rrbracket_{aux} \\
\llbracket p + q \rrbracket_{aux} & \overset{\cdot}{=} \llbracket p \rrbracket_{aux} \cup \llbracket q \rrbracket_{aux} \\
\llbracket p \parallel q \rrbracket_{aux} & \overset{\cdot}{=} \llbracket p \rrbracket_{aux} \parallel_{tr} \llbracket q \rrbracket_{aux}
\end{align*}
\]

where \(\parallel_{tr}\) auxiliary function 

\[
\{\langle \rangle\} \cup P \parallel_{tr} Q \overset{\cdot}{=} P \parallel_{tr} \{\langle \rangle\} \cup Q \overset{\cdot}{=} (P \parallel_{tr} Q)
\]
Trace: sequence of decorated actions. 
$\llbracket - \rrbracket_{aux} : CCSi$ process $\mapsto$ set of traces.

\[
\begin{align*}
\llbracket 0 \rrbracket_{aux} & \doteq \langle \rangle \\
\llbracket d; p \rrbracket_{aux} & \doteq d \lhd \llbracket p \rrbracket_{aux} \\
\llbracket p + q \rrbracket_{aux} & \doteq \llbracket p \rrbracket_{aux} \cup \llbracket q \rrbracket_{aux} \\
\llbracket p \parallel q \rrbracket_{aux} & \doteq \llbracket p \rrbracket_{aux} \parallel_{tr} \llbracket q \rrbracket_{aux}
\end{align*}
\]

where $\parallel_{tr}$ auxiliary function

\[
\begin{align*}
\{(J)\alpha \lhd tr\} \cup P \parallel_{tr} \{(J')\alpha' \lhd tr'\} \cup Q & \doteq \\
(J)\alpha \lhd \{(tr) \cup P \parallel_{tr} \{(J')\alpha' \lhd tr'\} \cup Q\} \cup \\
(J')\alpha' \lhd \{(J)\alpha \lhd tr'\} \cup \{tr\} \cup P \parallel_{tr} Q \cup \\
\cup\{(J \cup J')a \lhd \{tr\} \cup P \parallel_{tr} \{tr'\} \cup Q\} \cup \{(J')a \lhd tr' \in Q\} & \doteq \\
\{\alpha = a? \land \alpha' = a!\} \lor (\alpha = a! \lor \alpha' = a?)
\end{align*}
\]
\[ [p]_{tr} = \{ tr \mid tr \in [p]_{aux} \land closed(tr) \} \]

('closed(tr): tr contains no send or receive actions)
Comparing operational and IS-semantics:

- There is a one-one correspondence between local states of the protocol in IS semantics and local trace projections in operational semantics.
Comparing operational and IS-semantics:

- There is a one-one correspondence between local states of the protocol in IS semantics and local trace projections in operational semantics.

Consider finite initialized and prefix-closed (fipc) ISs. Characterizing the class of ISs generated:

- If $|A| = 1$: for each fipc interpreted system $R$, there is a process algebraic description $p$ such that $[\llbracket p \rrbracket]_{tr} = R$.
- For $|A| \geq 2$ and at least 2 agents: there exist fipc ISs that cannot be generated by any process algebraic specification.
Towards Characterization

Cf. embedding of DEL in ISs (van Benthem et al):

- Perfect Recall: by construction
- Synchronicity: depends on properties $\rho$
- Uniform No Miracles: ??

Future work: relate different $\rho$-types to structural properties of epistemic relations in ISs.
(E.g. with additional parameters distinguishing more groups of agents.)
We propose to include epistemic elements in operational specification.
We propose to include epistemic elements in operational specification.

A connection to interpreted systems helps to open the tools developed for multi-agent systems.
We propose to include epistemic elements in operational specification.

A connection to interpreted systems helps to open the tools developed for multi-agent systems.

On the theoretical level, it will be interesting to characterize the class of ISs generated by our framework. (Like van Benthem et al. 2010 did for DEL.)
We propose to include epistemic elements in operational specification.

A connection to interpreted systems helps to open the tools developed for multi-agent systems.

On the theoretical level, it will be interesting to characterize the class of ISs generated by our framework. (Like van Benthem et al. 2010 did for DEL.)

Any questions?
We propose to include epistemic elements in operational specification.

A connection to interpreted systems helps to open the tools developed for multi-agent systems.

On the theoretical level, it will be interesting to characterize the class of ISs generated by our framework. (Like van Benthem et al. 2010 did for DEL.)

Thank You!!!