Logic of Information Flow on Communication Channels

Tracking Number:377

ABSTRACT

In this paper, we develop an epistemic logic to specify and reason about the information flow on the underlying communication channels. By combining ideas from Dynamic Epistemic Logic (DEL) and Interpreted Systems (IS), our semantics offers a natural and neat way of modelling multi-agent communication scenarios with different assumptions about the observational power of agents. We relate our logic to the standard DEL and IS approaches and demonstrate its use by studying a telephone call communication scenario.

Categories and Subject Descriptors

I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*modal logic*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems*

General Terms

Theory

Keywords

modal logic, dynamic epistemic logic, interaction structures, protocol, channel

1. INTRODUCTION

The 1999 'National Science Quiz' of *The Netherlands Organisation for Scientific Research (NWO)*[1] had the following question [10, 16]

> *Six friends each have one piece of gossip. They start making phone calls. In every call they exchange all pieces of gossip that they know at that point. How many calls at least are needed to ensure that everyone knows all six pieces of gossip?*

To reason about the information flow in such scenario, we need to take into account the following issues: the messages that agents possess (e.g. secrets), the knowledge of agents, the dynamics of the system in terms of the information passing (e.g. telephone calls) and the underlying communication channels (e.g. network of landlines). To incorporate specific designs for such issues, we first need to make a choice between two mainstream logical frameworks to multi-agent systems: *Interpreted Systems* and *Dynamic Epistemic Logic*.

Interpreted Systems (ISs), introduced by [7] and [12] independently, are mathematical structures that combine history-based temporal components of a system with epistemic ones (defined in terms of *local states* of the agents). ISs are convenient to model knowledge development based on the given temporal development of a system. In ISs the epistemic structure is generated from the temporal structure in a uniform way. However, the generation of temporal structures is not specified in the framework.

A different perspective on the dynamics of multi-agent systems is provided by Dynamic Epistemic Logic (DEL) [8, 3]. The main focus of DEL is not on the temporal structure of the system but on the epistemic impact of events as the agents perceive them. The development of a system through time is essentially generated by executing so-called *action models* on a static initial model, to generate an updated static model. The epistemic relations in the initial static model and in the action models are not generated uniformly as in IS. It is customary to start out from a static situation of universal ignorance, where the ignorance is supposed to be common knowledge[2].

Much has been said already about the comparison of the two frameworks (see e.g. [17, 9]), but at a purely theoretical level. In this paper, we will demonstrate the benefits of combining the two approaches by presenting a framework where the temporal development of the system is generated by executing DEL-style actions and where epistemic relations are generated by matching local states and history of observations as in ISs.

Related Work and Contributions.

An early proposal to extend DEL with explicit communication channels is in [15]. Communication channels in an IS framework made their appearance in [13]. Recent work [11, 2] addresses the information passing on so-called *communication graphs* or *interaction structures*, where *"messages"* are either atomic propositions or Boolean combinations of atomic propositions. In [22] a PDL-style DEL language is developed that allows explicit specification of protocols. The present paper attempts to blend the DEL and IS approaches to communication along channels. More specifically, the contributions of this paper are:

- Combining insights from Dynamic Epistemic Logics and Interpreted Systems, we propose a logic $\mathcal{L}^I_{mpc}$ to specify and

[1] It is the 10th question from the 1999 edition. For a list of references about the problem c.f. [10].

[2] In a situation with n atomic propositions, this gives an initial model consisting of 2^n worlds, with universal accessibility relations for all agents.

Cite as: Logic of Information Flow on Communication Channels, Author(s), *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. XXX-XXX.

reason about the information flow over underlying communication channels. Unlike in previous work [11, 2, 15], we can *specify* the communication protocols in our language and deal with information flow in terms of both the *messages* and propositions.

- The semantics of $\mathcal{L}^I_{mpc}$ is given on single-state models with respect to different observational equivalence relations generated in IS-style, which are also studied and compared in this paper.
- The DEL-style actions in $\mathcal{L}^I_{mpc}$ allow us to model various communication actions such as message passing and group announcements. In particular we define an external informing action, which essentially announces the protocol that agents are supposed to follow, thus making it common knowledge that the future behavior of agents is constrained. It turns to make a crucial difference whether epistemic protocols such as those discussed in [19, 20, 21] are assumed to be common knowledge among the agents carrying out the protocol or not (see also [22]).
- Taking advantage of the concise nature of our semantics, we also propose a generic method of epistemic modeling where the initial model is simply the *real world* and all the initial assumptions are specified explicitly by means of formulas of $\mathcal{L}^I_{mpc}$. This significantly simplifies the modeling procedure. According to the semantics, the relevant possible states can be automatically constructed *on-the-fly* while evaluating the formulas. In particular, there is no need to specify the whole state space at the beginning.
- When the exact values of the messages are irrelevant, we can specify the protocol and the initial requirements in an intuitive and neat way, as demonstrated by the study of telephone communications among agents. We show that it is impossible to obtain new common knowledge by telephone calls or voice mails but that we can get arbitrarily close to common knowledge if we can not only send messages but also make statements like "I just called a and I know he got m".

The paper is organized as follows. We introduce our logic $\mathcal{L}^I_{mpc}$ in Section 2. Section 3 relates our logic to the standard DEL and IS approaches. Section 4 introduces a modeling method and illustrates this method by a study of variations on the puzzle that was mentioned at the start of the paper. The final section concludes and lists future work.

2. LOGIC $\mathcal{L}^I_{mpc}$

2.1 Language

Let I be a finite set of agents, M be a finite set of message terms and A be a finite set of basic actions. A network *net* is a hypergraph of agents in I, namely a set of subsets of I as in [2]. For example if $net = \{\{1,2\},\{1,2,3\}\}$ then there is a private channel $\{1,2\}$ of agents $1, 2$ and there is a public channel of all the three agents.

The set $Prop_{I,A,M}$ of basic propositions is defined by

$$p ::= has_i m \mid com(G) \mid past(\bar{\alpha}) \mid future(\alpha)$$

with $i \in I$, $m \in M$, $G \subseteq I$, $\bar{\alpha} = \alpha_0;\alpha_1;\ldots;\alpha_k \in A^*$ and $\alpha \in A$.

$has_i m$ is intended to mean that i possesses the message m[3]; while $com(G)$ expresses that group G forms a channel in the network; $past(\bar{\alpha})$ says that the sequence of actions $\bar{\alpha}$ just happened and $future(\alpha)$ means that α can be executed as a next step in the current protocol. The formulas of $\mathcal{L}^I_{mpc}$ are built from the set $Prop_{I,A,M}$ as follows:

$$\begin{array}{rcl} \phi & ::= & \top \mid p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \langle\pi\rangle\phi \mid C_G\phi \\ \pi & ::= & \alpha \mid \pi_1;\pi_2 \mid \pi_1 \cup \pi_2 \mid \pi^* \end{array}$$

with $p \in Prop_{I,A,M}$, $G \subseteq I$ and $\alpha \in A$. Let $\Pi_{A'}$ be the set of all *protocols* π based on basic actions in $A' \subseteq A$. Let $\Pi = \Pi_A$ and $Form(\mathcal{L}^I_{mpc})$ be the set of all the $\mathcal{L}^I_{mpc}$ formulas. Each $\alpha \in A$ is a tuple:

$$\langle G, \phi, M_0 \ldots M_{|I|}, x\rangle \in \mathcal{P}(I) \times Form(\mathcal{L}^I_{mpc}) \times (\mathcal{P}(M))^{|I|} \times (\Pi \cup \{\#\})$$

The intended meaning of the formulas is mostly as usual as in dynamic epistemic logics: $C_G\phi$ expresses "the agents in G commonly know ϕ", $\langle\pi\rangle\phi$ expresses "the protocol π can be executed, and at least one execution of π yields a state where ϕ holds". Here π is a regular expression built from the basic actions.

For every atomic action $\alpha = \langle G, \phi, M_0 \ldots M_{|I|}, x\rangle$, let $obs(\alpha) = G$ be the set of agents that can observe α; let $Pre(\alpha) = \phi$ be the precondition for α to be executed and let $Pos(\alpha) = \langle M_0 \ldots M_{|I|}, \rho\rangle$ (with $\rho \in \Pi \cup \{\#\}$) be the postcondition of execution of α. The postcondition $\langle M_0 \ldots M_{|I|}, \rho\rangle$ lists for each agent i the set of messages M_i that get delivered to i by action α and the protocol ρ that the agents are going to follow from now on. If ρ equals $\#$, this expresses that the agents keep following the current protocol, if ρ equals $\pi \in \Pi$ this expresses that they change their protocol to π. In this paper we assume that the agents can always observe the actions which deliver some messages to him, namely if β has $M_j \neq \emptyset$ then $j \in obs(\beta)$.

As usual, we define $\bot$, $\phi \vee \psi$, $\phi \to \psi$, $\langle C_G\rangle\phi$ and $[\pi]\phi$ as the abbreviations of $\neg\top$, $\neg(\neg\phi \wedge \neg\psi)$, $\neg\phi \vee \psi$, $\neg C_G\neg\phi$ and $\neg\langle\pi\rangle\neg\phi$ respectively. Moreover, we use the following abbreviations:

$$\begin{array}{rl} K_j\phi := & C_{\{j\}}\phi \\ has_i M' := & \bigwedge_{m\in M'} has_i m \\ dhas_G M' := & \bigwedge_{m\in M'} \bigvee_{j\in G} has_j m \\ com(net) := & \bigwedge_{G\in net} com(G) \wedge \bigwedge_{G\notin net} \neg com(G) \\ \pi^n := & \underbrace{\pi;\pi;\ldots;\pi}_{n} \\ \Sigma\Pi' := & \bigcup_{\pi\in\Pi'} \pi \text{ where } \Pi' \subset \Pi \text{ is finite.} \\ \langle\rangle^{\leq n}\phi := & \langle \bigcup_{k\leq n} (\Sigma A)^k\rangle\phi \\ \langle\rangle^{min(n)}\phi := & \langle\rangle^{\leq n}\phi \wedge \neg\langle\rangle^{\leq n-1}\phi \\ \mathtt{Know}_i\exists\mathtt{has}_j m := & K_i has_j m \wedge \neg has_i m \\ \exists\mathtt{Know}_i\mathtt{has}_j m := & K_i has_j m \wedge has_i m \end{array}$$

where: $K_j\phi$ means that agent j knows ϕ; $dhas_G M'$ says M' are distributed among agents in G; $com(net)$ specifies the communication channels in the network; $\langle\rangle^{\leq n}\phi$ should be read as "ϕ can be realized with a sequential protocol within n steps" and $\langle\rangle^{min(n)}\phi$ says "ϕ can be realized in n steps and needs at least n steps". Note that the usual temporal operator $\Diamond$ (sometimes called F) of IS approaches (e.g. [11]) can be defined by $\langle(\Sigma A)^*\rangle$ while $\langle\rangle^{\leq n}$ serves as a generalization of the *arbitrary announcement* that is added to DEL in [1].

By having both has and K in the language we can make a distinction between knowing a message and knowing its content. $\mathtt{Know}_i\exists\mathtt{has}_j m$ and $\exists\mathtt{Know}_i\mathtt{has}_j m$ express the *de dicto* and *de re* reading of *knowing a message*: $\mathtt{Know}_i\exists\mathtt{has}_j m$ says that agent i knows that agent j has *certain* message m, but he doesn't know the content of m himself. $\exists\mathtt{Know}_i\mathtt{has}_j m$ expresses that agent i knows that agent j has certain message m and he also possesses the message m himself thus knows the content. For example, let m be the

[3] *has* is a commonly used predicate in the logic of security protocols to model declarative knowledge about messages c.f., e.g., [14].

hidding place of Bin Laden and suppose it is commonly known that Al-Qaeda knows the place secretly, then $\texttt{Know}_i\exists\texttt{has}_{\texttt{Al-Qaeda}}m$ should intuitively hold but not $\exists\texttt{Know}_i\texttt{has}_{\texttt{Al-Qaeda}}m$ for $i \neq$ Al-Qaeda.

2.2 Semantics

In order to interpret basic propositions $Prop_{I,A,M}$, we let the finer structure of the basic propositions correspond with a finer structure in the states (replacing the traditional valuation in Kripke structures):

DEFINITION 1. *A state for $\mathcal{L}^I_{mpc}$ s is a tuple:*

$$\langle net, M_0, \ldots, M_{|I|}, \bar{\alpha}, M'_0, \ldots, M'_{|I|}, \pi\rangle \in Net \times F \times (A)^* \times F \times \Pi.$$

where $Net = \mathcal{P}(\mathcal{P}(I))$ and $F = (\mathcal{P}(M))^{|I|}$. Let $IS(s,i) = M'_i$ be i's current set of messages (information set), $AM(s) = \bar{\alpha}$ be the action history, $CC(s) = net$ be the available communication channels and $Prot(s) = \pi$ be the protocol the agents need to follow from this state. Let $AM_k(s) = \alpha_k$ in $\bar{\alpha}$. The initialization of s is another state:

$$Init(s) = \langle net, M_0, \ldots, M_{|I|}, \epsilon, M_0, \ldots, M_{|I|}, (\Sigma A)^*\rangle.$$

The length of s is $l(s) = |AM(s)|$.

Intuitively, each state represents a possible development of the system with the constraint for the future. Note that past is linear (it consists of a single sequence), but the future can be branching (it may consist of several sequences). As for $Init(s)$, we do not record any actions thus $AM(Init(s)) = \epsilon$ and $Prot(Init(s)) = (\Sigma A)^*$ simply says every protocol is possible in the future.

$has_i m$, $com(G)$ and $past(\bar{\alpha})$ can be interpreted in a straightforward way at state s according to $IS(s,i)$, $AM(s)$ and $CC(s)$ respectively. To give the semantics for $future(\alpha)$ at a state s, we need to check whether α *complys* with the current protocol $Prot(s)$ and compute the remaining protocol after the execution of α when we define the postcondition for α later on. For this, we define a *"division"* operation $\backslash\alpha$ on regular expressions with the auxiliary constants ε (empty sequence) and δ (deadlock) as follows:

$$\begin{array}{ll}
\varepsilon\backslash\alpha = \delta & \delta\backslash\alpha = \delta \\
\alpha\backslash\alpha = \varepsilon & \beta\backslash\alpha = \delta \\
(\pi;\pi')\backslash\alpha = (\pi\backslash\alpha);\pi' & (\pi \cup \pi')\backslash\alpha = \pi\backslash\alpha \cup \pi'\backslash\alpha \\
(\pi)^*\backslash\alpha = \pi\backslash\alpha;(\pi)^* &
\end{array}$$

where $\pi;\pi'$, $\pi + \pi'$ and π^* above have to be in a *normal form* by applying the absorbing rules below first:

$$\begin{array}{ll}
\varepsilon;\pi = \pi & \delta;\pi = \delta \\
\delta \cup \pi = \pi &
\end{array}$$

For example: $(\alpha \cup (\beta;\gamma))^*\backslash\beta = (\alpha\backslash\beta \cup (\beta;\gamma)\backslash\beta);(\alpha \cup \beta;\gamma)^* = (\delta \cup (\varepsilon;\gamma));(\alpha \cup \beta;\gamma)^* = \gamma;(\alpha \cup (\beta;\gamma))^*$. Note that in general we do not have $\beta;(\pi\backslash\beta) = \pi$.

Let $L(\pi)$ be the language of the regular expressions defined by the following:

$$\begin{array}{l}
L(\delta) = \emptyset \qquad L(\varepsilon) = \{\epsilon\} \qquad L(\alpha) = \{\alpha\} \\
L(\pi;\pi') = \{\bar{\alpha};\bar{\beta} \mid \bar{\alpha} \in L(\pi), \bar{\beta} \in L(\pi')\} \\
L(\pi \cup \pi') = L(\pi) \cup L(\pi') \\
L(\pi^*) = \{\bar{\alpha}_1;\ldots;\bar{\alpha}_n \mid \bar{\alpha}_1, \ldots, \bar{\alpha}_n \in L(\pi)\}
\end{array}$$

It is easy to see that the operation we defined can compute the remaining of the protocol after executing basic action α:

PROPOSITION 1. $L(\pi\backslash\alpha) = \{\bar{\beta} \mid \alpha;\bar{\beta} \in L(\pi)\}$.

Similar to [5, 2], we give the truth value of complex $\mathcal{L}^I_{mpc}$ formula on *single* states but not *pointed Kripke models*, while the possible states to interpret epistemic formulas are generated in a uniform way by $\sim^x_i$ defined later.

Let $s = \langle net, M_0, \ldots, M_{|I|}, \bar{\beta}, M'_0, \ldots, M'_{|I|}, \pi\rangle$, we have:

$$\begin{array}{rcl}
s \vDash has_i(m) & \Leftrightarrow & m \in IS(s,i) \\
s \vDash com(G) & \Leftrightarrow & G \in CC(s) \\
s \vDash past(\bar{\alpha}) & \Leftrightarrow & \bar{\alpha} \text{ is a suffix of } AM(s) \\
s \vDash future(\alpha) & \Leftrightarrow & Prot(s)\backslash\alpha \neq \delta \\
s \vDash \neg\phi & \Leftrightarrow & s \nvDash \phi \\
s \vDash \phi \wedge \psi & \Leftrightarrow & s \vDash \phi \text{ and } s \vDash \psi \\
s \vDash C_G\phi & \Leftrightarrow & \text{for all } v, \text{ if } s \sim^x_G t \text{ then } t \vDash \phi \\
s \vDash \langle\pi\rangle\phi & \Leftrightarrow & \exists s' : s[\![\pi]\!]s' \text{ and } s' \vDash \phi
\end{array}$$

where $\sim^x_G$ is the reflexive transitive closure of $\{\sim^x_i \mid i \in G\}$ and π are protocols functioning as *state changers*:

$$\begin{array}{rcl}
s[\![\alpha]\!]s' & \Leftrightarrow & s \vDash Pre(\alpha) \text{ and } s' = s|_{Pos(\alpha)} \\
s[\![\pi_1;\pi_2]\!]s' & \Leftrightarrow & s[\![\pi_1]\!] \circ [\![\pi_2]\!]s' \\
s[\![\pi_1 \cup \pi_2]\!]s' & \Leftrightarrow & s[\![\pi_1]\!] \cup [\![\pi_2]\!]s' \\
s[\![(\pi_1)^*]\!]s' & \Leftrightarrow & s[\![\pi_1]\!]^* s'
\end{array}$$

where $\circ, \cup$ and * at right-hand side express the usual composition, union and reflexive transitive closure on relations respectively. If $Pos(\alpha) = \langle N_0, \ldots, N_{|I|}, \rho\rangle$ then

$$s|_{Pos(\alpha)} = \langle net, M_0, \ldots, M_{|I|}, \bar{\beta};\alpha, M'_0 \cup N_0, \ldots, M'_{|I|} \cup N_{|I|}, f(\rho)\rangle$$

where $f(\rho) = \begin{cases} \pi\backslash\alpha \text{ if } \rho = \# \\ \pi' \quad \text{if } \rho = \pi' \end{cases}$.

Now we define $\sim^x_i$ among states. A state s is said to be *consistent* if $Init(s)[\![AM(s)]\!]s$. For the special case that $AM(s) = \epsilon$ we let $s[\![\epsilon]\!]s' \Leftrightarrow s = s'$. It is then easy to see that for any s, $Init(s)$ is always consistent[4].

We say $t \sim^x_i t'$ iff the following conditions are met:

consistency t and t' are consistent.

local initialization $IS(Init(t), i) = IS(Init(t'), i)$

local history $AM(t) \approx^x_i AM(t')$ where x is the type of the observational power defined below.

Let $AM(t) \approx^x_i AM(t') \Leftrightarrow AM(t)|^x_i = AM(t')|^x_i$. Then we can have several reasonable definitions of $AM(t)|^x_i$ to capture different observation powers of agents:

1. $AM(t)|^{set}_i = \{\alpha \mid i \in obs(\alpha)\}$ as in [2].
2. $AM(t)|^{1st}_i$ is the subsequence that only keeps the first occurrence of each $\alpha \in AM(t)|^{set}_i$ as in [4].
3. $AM(t)|^{asyn}_i$ is the subsequence that only keeps $\alpha \in AM(t)|^{set}_i$.
4. $AM(t)|^{\tau}_i$ is the sequence that replaces each occurrence of $\alpha \notin AM(t)|^{set}_i$ by τ.

where $x \in Sem = \{set, asyn, 1st, \tau\}$, we then have:

PROPOSITION 2. $\approx^{\tau}_i \subseteq \approx^{asyn}_i \subseteq \approx^{1st}_i \subseteq \approx^{set}_i$.

We then call the semantics defined by $\sim^x_i$ the *x-semantics*, and denote the corresponding satisfaction relation as $\vDash^x$.

Recall that we require that the agents can always observe the actions that change his information set. We then have:

[4]Note that we can actually omit the current information sets $IS(s,i)$ in the definition of a state, but compute it by applying the actions in $AM(s)$, thus only generate consistent states. We keep the current information sets there to simplify notations and make it more efficient to evaluate basic propositions according to the semantics.

PROPOSITION 3. *For any consistent state t: $t \sim_i^x t'$ implies $IS(t,i) = IS(t',i)$ where $x \in$ Sem.*

PROOF. Note that our actions can only add messages to the information sets of agents but never delete any messages. According to this monotonicity, we only need to check the above claim for $\sim_i^{set}$ and it is straightfoward since agent can always observe the action that changes his information set. □

2.3 Communication Actions

In the following we define a few very useful basic actions. Let $s = \langle net, M_0, \ldots, M_{|I|}, \bar{\alpha}, M'_0, \ldots, M'_{|I|}, \pi\rangle$. Postconditions of basic action β are in the form $Pos(\beta) = \langle N_0, \ldots, N_{|I|}, \rho\rangle$ where $N_j = \emptyset$ for $j \notin obs(\beta)$ and $\rho \in \Pi \cup \{\#\}$. We list the basic actions in the table below (where $j \in obs(\beta)$):

β (communication resp. channels)	obs	Pre common part is: $com(obs(\beta)) \wedge future(\beta)$	Pos common part. is: $\rho = \#$
$send^i_G(M')$	$G \cup \{i\}$	$has_i M$	$N_j = M'$
$share_G(M')$	G	$dhas_G M'$	$N_j = M'$
$sendall^i_G(M')$	$G \cup \{i\}$	$has_i M' \wedge \bigwedge_{m \notin M'} \neg has_i m$	$N_j = M'$
$shareall_G(M')$	G	$dhas_G M' \wedge \bigwedge_{m \notin M'} \neg dhas_i m$	$N_j = M'$
$inform^i_G(\phi)$	$G \cup \{i\}$	$K_i \phi$	$N_j = \emptyset$
β (external info)	obs	Pre common part is: $future(\beta)$	Pos no common part
$exinfo(\phi)$	I	ϕ	$\rho = \#$
$exprot(\pi')$	I	$\langle \pi' \rangle \top$	$\rho = \pi'$

The first group of actions are communication actions that respect the channels. $send^i_G(M')$ is the action that i sends the set of messages M' to the group G with precondition $com(obs(send^i_G(M'))) \wedge future(send^i_G(M')) \wedge has_i M$ meaning that there is a channel to perform this action and it is allowed by the current protocol and i should possess all the messages in M. $Pos(send^i_G(M')) = \langle N_0, \ldots, N_{|I|}, \#\rangle$ where $N_j = M'$ for $j \in obs(send^i_G(M'))$. $share_G(M')$ shares the messages distributed among the members of group G. $sendall^i_G(M')$ differs from $send^i_G(M')$ in the extra precondition that M' should be *all* the messages that i has. Similar for $shareall_G(M')$. $inform^i_G(\phi)$ is the group announcement of ϕ within $G \cup \{i\}$.

The second group of actions are public announcements that do not respect the channels. They model the external information which is given to the agents. For example, after executing $exinfo(\phi)$, the states agents consider possible will all satisfy ϕ due to the definition of $\sim^x_G$ and the fact that all the agents can observe this action. $exprot(\pi')$ announces the protocol π' that the agents are supposed to follow in the future. Note that it is different from the action $exinfo([\pi']\top)$. Actually $exprot(\pi')$ can never be defined by $exinfo(\phi)$ since $exprot(\pi')$ shapes the future by changing $Prot(s)$.

We can define more complex actions based on the above basic actions. For example:

$$mail^i_G(M') = \bigcup_{M'' \subseteq M'} sendall^i_G(M'')$$

models the voice mail from i for the group G, in which i shares all the messages that he possesses within M'[5]. Similarly $call^i_G(M') = \bigcup_{M'' \subseteq M'} shareall_i(G)M''$ models the conference call which shares all the messages that the group have in M'.

Similarly, new operator $\langle\rangle^{\leq n}_{A'} := \langle exprot((\Sigma_{\alpha \in A'}\alpha)^*)\rangle; \langle\rangle^{\leq n}$ can be defined to obtain a restricted version of bounded future operator such that $\langle\rangle^{\leq n}_{A'}\phi$ expresses that "there is a sequential protocol using only actions in A' to achieve ϕ in less or equal than n steps".

[5]Here M' encodes the *relevant context* e.g. messages that are *"about work"*.

3. COMPARISON WITH IS AND DEL

The results in this section relate our logic to IS and DEL approaches. Theorem 1 shows that by the semantics of $\mathcal{L}^I_{mpc}$, an interpreted system is generated implicitly from a single state. Proposition 4 and Theorem 1 demonstrate that our approach is powerful and concise in modelling actions, comparing to DEL.

Let us compare our approach to IS first. Note that in the following we only consider consistent states.

Let the history of s be a sequence: $hist(s) = s_0 s_1 \ldots s_{l(s)}$ where $s_0 = Init(s)$, $s_{l(s)} = s$ and $s_k[\![\alpha_k]\!]s_{k+1}$ for any k such that $\alpha_k = AM_k(s)$. It is easy to see that if $hist(s) = s_0 s_1 \ldots s_{l(s)}$ then $s_0 s_1 \ldots s_k = hist(s_k)$ for any $k \leq l(s)$. Let $ExpT^x$ be the Interpreted System with actions labels with respect to $x-$semantics: $\{H, \rightarrow_\alpha, \{R_i \mid i \in I\}, V\}$ where:

- $H = \{hist(s') \mid s' \text{ is consistent.}\}$
- $\langle s_0 \ldots s_n\rangle \rightarrow_\alpha \langle s_0 \ldots s_n s_{n+1}\rangle \Leftrightarrow s_n[\![\alpha]\!]s_{n+1}$.
- $\langle s_0 \ldots s_n\rangle R_i \langle s'_0 \ldots s'_m\rangle$ iff $s_n \sim^x_i s'_m$.
- $V(\langle s_0 \ldots s_n\rangle)(p) = \top \Leftrightarrow s_n \vDash^x p$ where $p \in Prop_{I,A,M}$.

It is clear that the language of $\mathcal{L}^I_{mpc}$ can be seen as a fragment of the *Propositional Dynamic Logic* (PDL): $\mathcal{L}^I_{pdl}$ with basic action set $A \cup I$ such that C_G can be seen as $(\Sigma G)^*$. Let $\Vdash_{PDL}$ denote the usual semantics of $\mathcal{L}^I_{pdl}$ then it is not hard to see:

THEOREM 1. *For any formula $\phi \in \mathcal{L}^I_{mpc}$ and for each consistent $\mathcal{L}^I_{mpc}$-state s:*

$$s \vDash^x \phi \Leftrightarrow ExpT^x, hist(s) \Vdash_{PDL} \phi.$$

This result shows that if we abstract away the inner structure of basic propositions and actions, then our logic can be looked as a PDL language interpreted on ISs that are generated in a particular way w.r.t the some constraints. Note that this result does not implies the decidability of $\mathcal{L}^I_{mpc}$ since although PDL language is decidable on general Kripke structures, we do not know yet whether it is decidable on the restricted class of the generated models $ExpT^x$.

Now consider the DEL language $\mathcal{L}^I_{del}$:

$$\phi \quad ::= \quad \top \mid p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \langle \mathbb{A}, e\rangle\phi \mid C_G\phi$$

where p is in a set of basic propositions $Prop$, $G \subseteq I$ and $\mathbb{A}$ is an *action model* with e as a designated action. Action models are tuples in the form of $(E, \{\asymp_i\}_{i\in I}, Pre, Pos)$ where $\asymp_i$ models agents i's observational power on events in E (e.g. $e_1 \asymp_i e_2$ means i is not sure which one of e_1 and e_2 happened); the precondition function $Pre : E \rightarrow \mathcal{L}^I_{del}$ describes when an event can happen and the postcondition $Pos : E \rightarrow (Prop \rightarrow \mathcal{L}^I_{del})$ makes (finitely many) basic propositions p change their truth values, after executing the events, to the truth values of $Pos(e)(p)$ in the static Kripke model, thus model the factual changes caused by the event [18].

The semantics for epistemic formulas is usual and

$$\mathbb{M}, s \Vdash_{DEL} \langle \mathbb{A}, e\rangle\phi \Leftrightarrow \mathbb{M} \otimes \mathbb{A}, (s, e) \vDash \phi$$

where the operation $\otimes$ is defined below:

Given a static Kripke model $\mathbb{M} = (W, \{R_i\}_{i\in I}, V)$ and an action model $\mathbb{A} = (E, \{\asymp_i\}_{i\in I}, Pre, Pos)$, the updated model $\mathbb{M} \otimes \mathbb{A} = (W', \{R'_i\}_{i\in I}, V')$ is defined:

$$\begin{aligned} W' &= \{\langle w, e\rangle \mid \mathbb{M}, w \vDash Pre(e)\} \\ R'_i &= \{(\langle w, e\rangle, \langle v, e'\rangle) \mid wR_i v \text{ and } e \asymp_i e'\} \\ V'(\langle w, e\rangle) &= V(w)(Pos(e)(p)) \end{aligned}$$

To facilitate the comparison, let us consider $\mathcal{L}^I_{mpc-*}$, the star-free fragment of $\mathcal{L}^I_{mpc}$[6]. Let $ExpK^x(s)$ be the Kripke model $\{W, \{R_i \mid$

[6]$*$ should not appear in the preconditions of actions.

$i \in I\}, V\}$ obtained by the *expansion* of the state s according to $x-$semantics:

- $W = \{s' \mid s \sim_I^x s'\}$ where $\sim_I^x$ is the reflexive transitive closure of $\{\sim_i^x \mid i \in I\}$.
- $R_i = \sim_i^x |_{W \times W}$.
- $V(s)(p) = \top \Leftrightarrow s \vDash^x p$ where $p \in Prop_{I,A,M}$.

Note that although I, A, M are assumed to be finite, W in $ExpK^x(s)$ can still be infinite due to the fact that we record the past explicitly in the states. For $x \in \{set, 1st, asyn\}$ which correspond to asynchronized semantics and an sequence of actions α, $\{\bar{\beta} \mid \bar{\alpha} \approx_i^x \bar{\beta}\}$ is infinite thus W can be infinite in $ExpK^x(s)$.

Based on $ExpK^x(s)$, it seems plausible to obtain a similar correspondence result as Theorem 1 for $\mathcal{L}^I_{mpc-*}$and $\mathcal{L}^I_{del}$, since the basic actions in $\mathcal{L}^I_{mpc-*}$ look like special cases of pointed action models in DEL. However, it is not the case in general. To see this, we first recall a fact from [17]: If we look $\langle \mathbb{A}, e\rangle$ as a basic action modality when considering PDL semantics, then for any formula $\phi \in \mathcal{L}^I_{del}$:

$$\mathbb{M}, s \Vdash_{DEL} \phi \Leftrightarrow Forest(\mathbb{M}, \mathcal{A}), (s) \Vdash_{PDL} \phi \quad (\star)$$

where $Forest(\mathbb{M}, \mathcal{A})$ is the IS generated by executing all the possible sequences of action models in $\mathcal{A}$ on $\mathbb{M}, s$[7]. We now show the effects of actions in $\mathcal{L}^I_{mpc}$ can not be simulated by action models.

PROPOSITION 4. *There is no translation $T : A \to \mathcal{A}$ such that for all consistent $\mathcal{L}^I_{mpc}$-state s:*

$$T(ExpT^x), hist(s) \leftrightarrows Forest(ExpK^x(s), \mathcal{A}), s$$

where $x \in \{set, 1st, asyn\}$, $T(ExpT^x)$ is the IS obtained from $ExpT^x$ by replacing each label of $\alpha \in A$ by $T(\alpha) \in \mathcal{A}$ and $\leftrightarrows$ is the bisimulation for transitions labled by $I \cup \mathcal{A}$.

PROOF. [17] shows that $Forest(ExpK^x(s))$ must satisfy the property of *Perfect Recall* meaning that if the agents can not distinguish two sequences of action $\bar{\alpha}; \alpha$ and $\bar{\beta}; \beta$ then they can not distinguish $\bar{\alpha}$ and $\bar{\beta}$. However, $ExpT^x$ clearly does not satisfy this property for $x \in \{set, 1st, asyn\}$ in general. For example, $send^i_j(M); \gamma \approx_j^x \gamma; send^i_j(M)$ where $x \in \{set, 1st, asyn\}$ and γ is some action j can not observe, however $send^i_j(M) \not\approx_j^x \gamma$. □

If we consider $\tau-$semantics, then a correspondence result can be obtained. First let $T_{DEL} : \mathcal{L}^I_{mpc-*} \to \mathcal{L}^I_{del}$ be defined as follows:

$$\begin{array}{rcl} T_{DEL}(\top) & = & \top \\ T_{DEL}(p) & = & p \\ T_{DEL}(\neg\phi) & = & \neg T_{DEL}(\phi) \\ T_{DEL}(\phi_1 \wedge \phi_2) & = & T_{DEL}(\phi_1) \wedge T_{DEL}(\phi_2) \\ T_{DEL}([\alpha]\phi) & = & [ExpA^x(\alpha)]T_{DEL}(\phi) \\ T_{DEL}([\pi_1 \cup \pi_2]\phi) & = & T_{DEL}([\pi_1]\phi) \wedge T_{DEL}([\pi_2]\phi) \\ T_{DEL}([\pi_1; \pi_2]\phi) & = & T_{DEL}([\pi_1][\pi_2]\phi) \end{array}$$

where $ExpA^\tau(\alpha)$ is the pointed action model $\{E, \{R_i \mid i \in I\}, V, e_\alpha\}$ obtained by the *saturation* of the action α according to $x-$semantics:

- $E = \{e_\beta \mid \beta \in A\}$
- $e_\beta R_i e_{\beta'} \Leftrightarrow \beta = \beta'$ or $i \notin obs(\beta) \cup obs(\beta')$.
- $Pre(e_\beta) = T_{DEL}(Pre(\beta))$.
- If $Pos(\beta) = \langle M_0, \ldots, M_I, x\rangle$ then:
  $Pos(e_\beta)(has_i m) = \begin{cases} \top & \text{if } m \in M_i \\ has_i m & \text{if otherwise} \end{cases}$
  $Pos(e_\beta)(com(G)) = com(G)$
  $Pos(e_\beta)(past(\bar{\gamma}; \gamma)) = \begin{cases} past(\bar{\gamma}) & \text{if } \gamma = \beta \\ \bot & \text{if otherwise} \end{cases}$

[7]Due to the limit of space, readers are refered to [17] for details.

Note that we have not defined $Pos(e_\beta)(future(\gamma))$ yet. Unfortunately it is undefinable by postcondition in DEL framework, namely, by a function assigning each $future(\gamma)$ a DEL formula. To see this, first note that the truth value of $future(\gamma)$ depends on the *protocol* that agents are going to follow and it is not expressible so far in our language. Moreover, even if we introduce $protocol(\pi)$ in the langauge to denote it, we still need infinite disjunctions: $Pos(e_\beta)(future(\gamma)) = \bigvee\{protocol(\pi) \mid \pi\backslash(\beta; \gamma) \neq \delta\}$.
To go around this, we can restrict ourselves to the actions that do not change the protocol, namely those α such that $Pos(\alpha) = \langle M_0, \ldots, M_I, \#\rangle$. Clearly this will exclude $exprot(\pi)$ defined earlier. And then we can set $Pos(e_\beta)(future(\gamma)) = \top$ and obtain the following result:

THEOREM 2. *If A does not contain any "protocol changer", then for any $\phi \in \mathcal{L}^I_{mpc-*}$ for any consistent $\mathcal{L}^I_{mpc}$-state s:*

$$s \vDash^\tau \phi \Leftrightarrow ExpK^\tau(s), s \Vdash_{DEL} T_{DEL}(\phi).$$

4. APPLICATIONS

4.1 Common Knowledge

Before the case study of the telephone communication scenario mentioned in the introduction, we first prove some general results concerning common knowledge. As an appetizer, we show that common knowledge cannot be reached if there is no channel containing all agents (i.e. agents cannot communicate publicly).

We first prove that if the agents can perform any non-public action and have not agreed on a special protocol, then there is always some *"dummy"* action that does not change anything about the knowledge of the agents:

LEMMA 1. *For any group of agents G, there is an action α^G_d such that $obs(\alpha^G_d) = G$ and for any set of basic actions A containing α^G_d, any state s such that $G \in CC(s)$, and any formula φ that does not contain any instances of $past(\bar{\alpha})$:*

$$s \vDash^\tau \langle exinfo(com(CC(s))); exprot((\bigcup_{\alpha \in A} \alpha)^*)\rangle(\varphi \leftrightarrow \langle \alpha^G_d\rangle\varphi)$$

PROOF. Let $\pi = exinfo(com(CC(s))); exprot((\bigcup_{\alpha\in A} \alpha)^*)$ and $\alpha^G_d = inform^i_G(\top)$ for some $i \in G$. Let $s[\![\pi]\!]s'$. We need to show $s' \vDash^\tau \varphi \leftrightarrow \langle \alpha^G_d\rangle\varphi$. The proof goes with induction on φ. The nontrivial cases are:

- $\varphi = future(\alpha)$. Since the protocol is unchanged and clearly $(\bigcup_{\alpha\in A} \alpha)^*\backslash\alpha^G_d = (\bigcup_{\alpha\in A} \alpha)^*$, the valuation of φ is unchanged.
- $\varphi = C_{G'}\psi$. Clearly, $C_{G'}\psi \to \langle \alpha^G_d\rangle C_{G'}\psi$ holds. For the other direction: suppose $s' \vDash^\tau \neg C_{G'}\psi$ we need to show $s' \vDash^\tau \neg\langle \alpha^G_d\rangle C_G\psi$. Clearly there is some G'^*-path to a state t where $\neg\psi$ holds. Then since this connection exists and the protocol and communication channel are common knowledge, $Prot(t') = Prot(s')$ and $CC(t') = CC(s')$ for any t' on the path and also for $t' = t$. Then by induction hypothesis, $t \vDash^\tau \langle \alpha^G_d\rangle\neg\psi$. For every t' on the path there is a unique state $u_{t'}$ such that $t'[\![\alpha^G_d]\!]u_{t'}$ because the preconditions of α^G_d hold. The links on the path from s' and t are preserved in a path from $u_{s'}$ to u_t, because we performed the same action at all worlds in the path. Since $t \vDash^\tau \langle \alpha^G_d\rangle\neg\psi$, $u_t \vDash^\tau \neg\psi$. So then since there is a path from $u_{s'}$ to u_t, $u_{s'} \vDash^\tau \neg C_G\psi$ and $s' \vDash^\tau \neg\langle \alpha^G_d\rangle C_G\psi$. □

THEOREM 3. *For any $n \in \mathbb{N}$, any formula φ containing no instances of $past(\bar{\alpha})$, any state s such that $I \notin CC(s)$ and any set*

of basic actions A containing only communications respecting the channels such that for every group $G \in CC(s)$ there is a dummy action $\alpha_d^G \in A$,

$$s \vDash^{\tau} \langle \mathit{exinfo}(\mathit{com}(CC(s))); \mathit{exprot}((\bigcup_{\alpha \in A} \alpha)^*)\rangle(\neg C_I\varphi \to \neg\langle\rangle^{\leq n} C_I\varphi)$$

PROOF. Suppose there was a minimal n such that the property would hold. Then there is a sequence of actions $\bar{\alpha}$ of length n such that after executing $\bar{\alpha}$, φ is common knowledge. Let $\bar{\alpha} = \bar{\beta}; \alpha$ and $G = obs(\alpha)$ and $j \notin G$ some outsider. Such j always exists because $I \notin CC(s)$. Since n was minimal, after execution of $\bar{\beta}$ there is no common knowledge of φ. Suppose we would after execution of $\bar{\beta}$, instead of α, execute α_d^G. By the previous lemma after this there would be no common knowledge of φ. Agent j observes a τ when α is executed, but he would also observe a τ if α_d^G was executed. So agent j can never know whether α or α_d^G was executed and he can never know whether common knowledge was established. So there can never be common knowledge of φ. □

Note that although we may have dummy actions, we cannot reduce τ-semantics to $asyn$-semantics, with $past(\bar{\alpha})$ in the language e.g $s \vDash^{\tau} \langle\alpha_d^G\rangle K_i \neg past(\alpha; \beta)$ but $s \vDash^{asyn} \langle\alpha_d^G\rangle \neg K_i \neg past(\alpha; \beta)$ where $i \notin G \cup obs(\alpha) \cup obs(\beta)$. In fact, if we use $asyn$-semantics, we can prove a more general result:

THEOREM 4. *For any $n \in \mathbb{N}$, for any state s with $I \notin CC(s)$, any protocol π containing only communications that respect the channel and any $\varphi \in \mathcal{L}^I_{mpc}$:*

$$s \vDash^{asyn} \langle \mathit{exinfo}(\mathit{com}(CC(s))); \mathit{exprot}(\pi)\rangle(\neg C_I\varphi \to \neg\langle\rangle^{\leq n} C_I\varphi)$$

PROOF. Suppose the opposite. Let n be the minimal such number. Then there is some sequence $\bar{\alpha}$ of length n such that after the execution of $\bar{\alpha}$, φ is common knowledge. Define α and $\bar{\beta}$ as $\bar{\alpha} = \bar{\beta}; \alpha$. Let $G = obs(\alpha)$ and $j \notin G$ some outsider. Since n was minimal, common knowledge of φ does not hold before α was executed. But j gets no information on the moment α was executed, so j can never know whether α was executed and common knowledge was reached. So common knowledge can never be reached. □

So if the communication is *asynchronous* in the sense that agents who do not participate in the communication are unaware that any communication is going on, then even if the agents can publicly agree on a protocol beforehand they still cannot reach common knowledge. On the other hand, in the *synchronous* case (τ-semantics), if we publicly announce the protocol then agents may know what is going on and moreover whether a star-free protocol is finished by counting the steps. In that case common knowledge may be achieved.

4.2 Telephone Calls

Let us recall the scenario: a group of people each know a secret and they can make telephone calls between every two people in order to communicate all their secrets. We want to know the minimal number of telephone calls needed to make sure everyone knows all secrets. Before modeling this particular situation, we first propose a general modeling method based on our semantics and actions we defined in Section 2.3:

1. Select a set of suitable actions A to model the communications in the scenario.
2. Build a single state as the *real world* to model the initial setting. i.e. $s = \langle net, \bar{M}_i, \epsilon, \bar{M}_i, (\Sigma A)^*\rangle$ where net is the network, $\bar{M}_i$ models *"who has what"* and $(\Sigma A)^*$ restricts the actions agents can use.
3. *Translate* the informal assumptions of the scenario into formulas or protocols in $\mathcal{L}^I_{mpc}$.
4. Use $exinfo(\phi)$ $exprot(\pi)$ to make the above assumptions common knowledge.

Now we are ready to model the telephone call scenario. We already defined $call^i_G(M_I)$ and $mail^i_G(M_I)$ in Section 2.3 as a conference call or mail to a group G, sharing all messages the group has. Here the call between two people is just a special case, thus we complete the first step. Let $M_I = \{m_0, \ldots m_{|I|}\}$, network $net^{tel}_I = \{\{i, j\} \mid i \neq j \in I\}$. Here m_i is the secret of agent i. Then the initial state is:

$$s^{tel}_I = \langle net^{tel}_I, \{m_0\} \ldots \{m_{|I|}\}, \epsilon, \{m_0\} \ldots \{m_{|I|}\}, \pi\rangle$$

where $\pi := (\bigcup_{G \in net^{tel}} A)^*$ is the protocol the agents follow and expresses that the agents can only make one on one telephone calls, sharing all their messages. As we can see, in the initial situation each agent only knows his own secret. We use some abbreviations for facts we need to express:

$$\begin{aligned} \texttt{OneSecEach}_I &:= \textstyle\bigwedge_{i \in I}(has_i m_i \wedge \bigwedge_{j \neq i} \neg has_j m_i) \\ \texttt{HasAll}_I &:= \textstyle\bigwedge_{i \in I} has_i M_I \\ \texttt{TP} &:= \mathit{exinfo}(\mathit{com}(net^{tel}_I) \wedge \texttt{OneSecEach}_I) \\ \texttt{TP}_A &:= \texttt{TP}; \mathit{exprot}((\textstyle\bigcup_{\alpha \in A} \alpha)^*) \end{aligned}$$

$\texttt{OneSecEach}_I$ translates the assumption that "all agents know one secret not known to the other agents". $\texttt{HasAll}_I$ expresses that all agents know all secrets, as the goal we want to achieve. It is easy to see that:

$$s^{tel}_I \vDash^{x} \langle \mathit{exinfo}(\texttt{OneSecEach}_I)\rangle C_I \bigwedge_{i \neq j \in I} \texttt{Know}_i \exists \texttt{has}_j m_j$$

so after a public announcement that each agent has one secret, it is common knowledge that every agent knows that every other agent j has a secret m_j, and also that the agents except j do not have this secret. However,

$$s^{tel}_I \nvDash^{x} \langle \mathit{exinfo}(\texttt{OneSecEach}_I)\rangle \bigvee_{i \neq j \in I} \exists \texttt{Know}_i \texttt{has}_j m_j$$

so after this same public announcement there is not one agent who knows the secret of another agent. These results hold for any $x \in$ *Sem*. In our framework, we use public announcements to set the communication channel and protocol. $\texttt{TP}$ summarizes the announcements needed for the starting situation of the telephone puzzle without the protocol and $\texttt{TP}_A$ adds the information that the agents can use the actions from A. We use *call,mail,inform* to denote the sets of actions with the corresponding types.

Then the following result states that we need exactly $2|I| - 4$ calls to make sure every agent knows all secrets:

PROPOSITION 5. *For any $x \in$ Sem:*

$$s^{tel}_I \vDash^{x} \langle TP_{call}\rangle\langle\rangle^{min(2|I|-4)} \texttt{HasAll}_I$$

A proof of this proposition is given in [10]. The protocol given there is the following: pick a group of four agents 1 ... 4 and let 4 be their informant. Let all other agents call agent 4, then let the four agents communicate all their secrets within their group and let all other agents call agent 4 again. In our framework we can express this as follows: $call^4_5(M_I)$; ...; $call^4_{|I|}(M_I)$; $call^1_2(M_I)$; $call^3_4(M_I)$; $call^1_3(M_I)$; $call^2_4(M_I)$; $call^4_5(M_I)$; ...; $call^4_{|I|}(M_I)$

Now assume the agents cannot make direct telephone calls, but they can only leave voicemail messages. This means that any agent can tell the secrets he knows to another agent, but he cannot in the

same call also learn the secrets the other agent knows. How many voicemail messages would we need in this case?

Intuitively we can use $mail^i_j(M_I)$; $mail^j_i(M_I)$ to mimic each $call^i_j(M_I)$, thus we have:

$$s^{tel}_I \models^x \langle \mathtt{TP}_{mail} \rangle \langle\rangle^{\leq 4|I|-4} \mathtt{HasAll}_I.$$

However, we can do much better:

PROPOSITION 6. *For any $x \in Sem$:*

$$s^{tel}_I \models^x \langle TP_{mail} \rangle \langle\rangle^{min(2|I|-2)} HasAll_I$$

PROOF. Consider the following protocol: $mail^1_2(M_I)$; $mail^2_3(M_I)$; ...; $mail^{|I|-1}_{|I|}(M_I)$; $mail^{|I|}_1(M_I)$; $mail^{|I|}_2(M_I)$; ...; $mail^{|I|}_{|I|-1}(M_I)$. Clearly, this results in all agents knowing all secrets. The length of this protocol is $2|I|-2$. This protocol is minimal. To see why this holds, first observe that there has to be one agent who is the first to learn all secrets. For this agent to exist all other agents will first have to make at least one call to reveal their secret to someone else. This is already $|I|-1$ calls. The moment that agent learns all secrets, since he is the first, all other agents do not know all secrets. So each of them has to receive at least one more call in order to learn all secrets. This also takes $|I|-1$ calls which brings the total number of calls to $2|I|-2$. □

Note that to obtain the above results, we did not use the full power of our framework, since the agents can only communicate the content of their messages and not about higher-order knowledge. In the following, we will study whether we can reach common knowledge of $\mathtt{HasAll}_I$ under $\tau-$semantics. We give the agents more power by allowing them to communicate not only messages but arbitrary formulas of the language in one-on-one calls by doing an *inform* action. Even in this case, we can never reach common knowledge of all messages:

PROPOSITION 7. *For any $n \in \mathbb{N}$, if $|I| > 2$ then:*

$$s_I \not\models^\tau \langle TP_{call,inform} \rangle \langle\rangle^{\leq n} C_I HasAll_I$$

PROOF. Follows from Theorem 3. □

However, we can approach common knowledge arbitrarily close. For any finite sequence of agents $w = ij...k$ define:

$$K_w\varphi := K_iK_j...K_k\varphi$$

PROPOSITION 8. *For any finite sequence w of agents from I, there exists some $n \in \mathbb{N}$ such that:*

$$s_I \models^\tau \langle TP_{call,inform} \rangle \langle\rangle^{\leq n} K_w HasAll_I$$

PROOF. We will give a protocol that results in the desired property. First we execute the protocol given in the proof of Proposition 6. Note that after executing this protocol, agent $|I|$ knows that everyone knows all secrets. Let $w = a_1...a_n$. We execute $inform^{|I|}_{a_n}(\mathtt{HasAll}_I)$; $inform^{|I|}_{a_{n-1}}(K_{a_n}\mathtt{HasAll}_I)$; ...; $inform^{|I|}_{a_1}(K_2...K_{a_n}\mathtt{HasAll}_I)$ and clearly, after these actions the desired property will hold. □

Surprisingly, if we do not give the agents the extra power of communicating arbitrary formulas then in the case that $|I| = 3$ we can reach common knowledge. In our τ-semantics, when two agents call each other the third one will know something happened because he observes a τ action. This is a bit like trying to use a telephone line and getting a busy tone: you know some communication is going on, but you don't know between which agents it is. If there are only three agents and it is common knowledge that the only possible communicative action is calling, then the third agent knows the other two are calling each other. This gives the following result:

PROPOSITION 9. *If $|I| \leq 3$ then for some $n \in \mathbb{N}$:*

$$s_I \models^\tau \langle TP_{call} \rangle \langle\rangle^{\leq n} C_I HasAll_I$$

PROOF. For $|I| < 3$ the proof is trivial. Suppose $|I| = 3$, say $I = \{1,2,3\}$. A protocol that results in the desired property is as follows. First, execute $call^1_2(M_I)$, $call^2_3(M_I)$ and $call^2_1(M_I)$. Now all agents know all secrets, and agent 2 knows this. Also, since agent 1 learned the secret of agent 3 from agent 2, he knows that agent 2 and 3 must have communicated after the last time he spoke to agent 2, so agent 3 must know the secret of agent 1. Regarding agent 3, he knows agent 2 has all secrets the moment he communicated with agent 2, and he observed a τ when agent 2 called agent 1 after that. Since there are only three agents agent 3 can deduce that agent 1 and 2 communicated so he knows agent 1 knows all secrets. Since all agents can reason about each others knowledge it is common knowledge that all agents have all secrets. □

Now imagine a situation where the agents are allowed to publicly announce a protocol they are going to follow, which is more complex than just the set of actions they can choose from. Then, in our τ-semantics, it is possible to reach common knowledge:

PROPOSITION 10. *There is a protocol π of call actions such that*

$$s_I \models^\tau \langle TP; exprot(\pi) \rangle \langle\rangle^{\leq n} C_I HasAll_I$$

PROOF. Let π be the protocol given in the proof of proposition 5. Let the agents agree to execute π with an $exprot(\pi)$ action and then execute π. Since each agent receives a τ at every communicative actions, they can all count the number of communicative actions that have been executed and they all know when the protocol has been executed. So at that moment, it will be common knowledge that everyone has all secrets. □

This shows the use of the ability to communicate about the future protocol and not only about the past and present. There are many more situations where announcing the protocol is very important, for example the puzzle of 100 prisoners and a light bulb [6] or many situations in distributed computing.

However, when we use $asyn$-semantics, the agents cannot count the number of communicative actions happening and so they can never know when the protocol has been executed. Because of this they can never reach common knowledge:

PROPOSITION 11. *There is no protocol π of call and $inform$ actions such that*

$$s_I \models^{asyn} \langle TP; exprot(\pi) \rangle \langle\rangle^{\leq n}_{call,inform} C_I HasAll_I$$

PROOF. Follows from Theorem 4. □

These results show the way we can use our framework to model a lot of different situations, often with surprising outcomes.

5. CONCLUSIONS AND FUTURE WORK

We developed an expressive dynamic epistemic logic tailored to specify and reason about the information flow over communication channels, and we proposed an intuitive lightweight modeling method for multi-agent communications scenarios. The logic and the modeling method were put to use in the telephone call example.

Our framework is very flexible in modeling different observational power of agents and various communication actions. For example, we can define the communication action in [11] : "i gets j's information without j noticing that" as $\alpha = download^i_j(M)$ with $obs(\alpha) = i$, $Pre(\alpha) = com(\{i, j\}) \wedge has_j M$ and a suitable postcondition adding messages to i's information set[8]. Therefore our framework can facilitate the comparison among different approaches with different assumptions. The table below summarizes the setting of our framework comparing to others:

Ref	Actions	Information flow	Obs. Power
[15]	inform	propositions	$\equiv^{\tau}$
[11]	download	Boolean atomic propositions	$\equiv^{\tau}$
[2]	inform	positive atomic propositions	$\equiv^{set}$
Our work	by design	messages or propositions	by design

We end with a list of further issues to be explored:

Theoretical Issues Many theoretical issues are left for future work e.g. the model checking and satisfiability problem of (the fragments of) $\mathcal{L}^I_{mpc}$ w.r.t different x-semantics; the expressivity of $\mathcal{L}^I_{mpc}$ comparing to various fixed point logics. Another interesting issue is the logical characterization of the observational equivalences defined in our work.

Network In this work, we take as networks the hyper graphs of [2], thus assuming the communication channels to be symmetric. More constrained network definitions with asymmetric channels are also possible. Moreover, different social networks/organizations may have different properties, e.g. the network of a group of gossiping girls is usually connected and transtive[9] while the network for a secret society is usually not transitive due to the hierarchy and secrecy. Thus leaking a secret to your closest girl friend may cause it to be a shared knowledge among all the girls on the next day, but gossiping about your boss with the juniors under your supervision might be safe in a secret society.

Actions There are other useful actions that we did not cover in the paper. For example, we have assumed that message passing actions are always monotonic but there are cases when deleting messages from memory or buffer is natrual. Another assumption is that the agents either clearly observe an action or observe nothing at all. This excludes the modeling of actions which may give some agents partial observations e.g. BCC in email. [15] also mentioned the possibility of changing the channels, e.g. deleting people from your Christmas card sending list if they did not a reply card last year. Such actions could be handled within our framework with little adaption.

Protocol We use regular expressions without tests to specify sequential protocols. We leave out tests since the observation of a test is not clear, unless grouped with follow-up actions. It seems that this is expressive enough for many useful applications. In the more general setting, we would like to have parallel composition in the protocol language and model the protocol by composing local protocols for each agent. We may also link our work to [17, 9], where the *"protocol"* is considered as a set of sequences explicitly.

Knowledge Transfer Our framework paves a way to discuss message passing and knowledge transfer over communication channels at the same time, thus maybe applicable to a security setting where information flows should be controlled strictly complying certain knowledge requirements. The distinction of de dicto and de re reading of knowledge may help us to formalize zero knowledge proofs.

[8][11] phrases such download action with propositions but not messages.

[9]In the sense that if girl A can call girl B and girl B can call girl C then A is in touch with C.

6. REFERENCES

[1] T. Agotnes, P. Balbiani, H. van Ditmarsch, and P. Seban. Group announcement logic. *To appear in Journal of Applied Logic*.

[2] K. R. Apt, A. Witzel, and J. A. Zvesper. Common knowledge in interaction structures. In A. Heifetz, editor, *TARK*, pages 4–13, 2009.

[3] A. Baltag, L. S. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. Technical Report SEN-R9922, CWI, Amsterdam, 1999.

[4] A. Baskar, R. Ramanujam, and S. P. Suresh. Knowledge-based modelling of voting protocols. In *TARK '07: Proceedings of the 11th conference on Theoretical aspects of rationality and knowledge*, pages 62–71, New York, NY, USA, 2007. ACM.

[5] M. Cohen and M. Dam. A complete axiomatization of knowledge and cryptography. In *LICS*, pages 77–88. IEEE Computer Society, 2007.

[6] P. O. Dehaye, D. Ford, and H. Segerman. One hundred prisoners and a light bulb. *Mathematical Intelligencer*, 24(4):53–61, 2003.

[7] R. Fagin, J. Y. Halpern, M. Y. Vardi, and Y. Moses. *Reasoning about knowledge*. MIT Press, Cambridge, MA, USA, 1995.

[8] J. Gerbrandy and W. Groeneveld. Reasoning about information change. *Journal of Logic, Language and Information*, 6(2):147–169, April 1997.

[9] T. Hoshi and A. Yap. Dynamic epistemic logic with branching temporal structures. *Synthese*, 169(2):259–281, July 2009.

[10] C. A. J. Hurkens. Spreading gossip efficiently. *Nieuw Archief voor Wiskunde*, 5/1(2):208–210, 2000.

[11] E. Pacuit and R. Parikh. Reasoning about communication graphs. In J. van Benthem, D. Gabbay, and B. Löwe, editors, *Interactive Logic — Proceedings of the 7th Augustus de Morgan Workshop*, Texts in Logic and Games, pages 135–157, Amsterdam, 2007.

[12] R. Parikh and R. Ramanujam. Distributed processes and the logic of knowledge. In *Proceedings of the Conference on Logic of Programs*, pages 256–268, London, UK, 1985. Springer-Verlag.

[13] R. Parikh and R. Ramanujam. A knowledge based semantics of messages. *Journal of Logic, Language and Information*, 12(4), 2003.

[14] R. Ramanujam and S. P. Suresh. Deciding knowledge properties of security protocols. In *Proc. Theoretical Aspects of Rationality and Knowledge*, pages 219–235. Morgan Kaufmann, 2005.

[15] F. Roelofsen. Exploring logical perspectives on dis- tributed information and its dynamics. Master's thesis, University of Amsterdam, 2005.

[16] J. van Benthem. 'one is a lonely number': on the logic of communication. In Z. Chatzidakis, P. Koepke, and W. Pohlers, editors, *Logic Colloquium '02*, pages 96–129, Wellesley MA, 2002. ASL & A.K. Peters.

[17] J. van Benthem, J. Gerbrandy, and E. Pacuit. Merging frameworks for interaction: Del and etl. In *TARK '07: Proceedings of the 11th conference on Theoretical aspects of rationality and knowledge*, pages 72–81, New York, NY, USA, 2007. ACM.

[18] J. van Benthem, J. van Eijck, and B. Kooi. Logics of communication and change. *Information and Computation*, 204(11):1620–1662, November 2006.

[19] H. van Ditmarsch. The russian cards problem. *Studia Logica*, pages 31–62, October 2003.

[20] H. van Ditmarsch and B. Kooi. Semantic results for ontic and epistemic change. In G. Bonanno, W. van der Hoek, and M. Wooldridge, editors, *Logic and the Foundations of Game and Decision Theory (LOFT 7)*, pages 87–117, October 2008.

[21] H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. (Synthese Library). Springer, 1st edition, November 2007.

[22] Y. Wang, L. Kuppusamy, and J. van Eijck. Verifying epistemic protocols under common knowledge. In *TARK '09: Proceedings of the 12th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 257–266, New York, NY, USA, 2009. ACM.