



## UvA-DARE (Digital Academic Repository)

### Epistemic modelling and protocol dynamics

Wang, Y.

**Publication date**  
2010

[Link to publication](#)

#### **Citation for published version (APA):**

Wang, Y. (2010). *Epistemic modelling and protocol dynamics*. [Thesis, fully internal, Universiteit van Amsterdam]. Institute for Logic, Language and Computation.

#### **General rights**

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

#### **Disclaimer/Complaints regulations**

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

## Chapter 1

---

## Introduction

The ice cream company Häagen-Dazs has an advertising slogan: “Love her, love Häagen-Dazs”. In China, the company uses a slightly revised Chinese version of this slogan, which is literally translated in English as “Love her, take her to Häagen-Dazs!”. Compared to the original slogan, the Chinese version refines the conditional “protocol” with an action which is much more explicit than love itself. It seems the subtle revision really can make a difference: this supermarket ice cream in U.S. and Europe has become one of the most popular must-buys among Chinese young couples (ask a Shanghai girl about it!). It seems that although it may take more than one life to really understand what love is, you can simply show your love by taking your girlfriend to a nearby Häagen-Dazs shop (and of course buying something there). Actually, it does not matter what love is, what really matters is that what you do is *commonly known* to be a proof of your (undefined) love. How come an ice cream is associated with love? For that you must know that the company follows a super high-end marketing strategy in China. It is the huge price difference with the regular ice creams that contributes to the protocol “If you love her **then** take her to Häagen-Dazs!”, for this is what allows you to show (off) your love. The protocol is clearly not about truth, but it makes information flow.

Here by “protocol” we refer to the general notion of procedural rules that govern the actions of humans or machines. Besides giving meaning to actions as in the Häagen-Dazs story, protocols also let us know what to do or what not to do. In many cases they are the reasons for us to act in a certain way. When you are driving a car you are also driving with various traffic protocols. In case an accident happens legal protocols are called into play. While you are sending emails or sms to a friend to complain about the bad luck, communication protocols on computers are running to make sure the messages are delivered. Your friend may reply to you with a remark against the current local government who initiated a construction project which led to the traffic chaos in the city centre and claim he will vote for another party a few days later in the election according to the political protocol. Because of the existence of such protocols which restrict the potential behaviour of humans and machines, we save our civilization from a chaotic state. Without doubt, protocols rule the world.

Due to the importance of protocols, it is crucial to know the protocols. The

French greet each other by cheek kissing for (usually) two times while the Dutch generally do it thrice. The first cheek kissing between someone from France and someone Dutch may leave the proper termination of their greeting protocols in question. However, for someone Chinese used to the greeting protocol of shaking hands, the number of kisses is not the (only) question to execute the first such greeting successfully: from which side should I start? how much noise should I make? why alternating left and right? . . . A protocol announcement could solve this in advance. If no information is provided, people can always rely on a default protocol such as wait-and-see or copy-cat. The difference in protocols is the reason behind many conflicts and misunderstandings, so keeping your protocol knowledge updated is also important.

In many cases, protocols are used to reach certain goals, e.g., the exchange protocol *cash and carry* is to guarantee a fair exchange in a (hostile) open market. However, knowing the protocol may also prevent the protocol from achieving its goal. For example, if a girl *knows* that the guy who takes her out on a first date acts out the protocol “ask her about herself, to make her think you are really interested in her feelings”, she is maybe less impressed with how the date goes than if she ignores this. As a more intricate example, consider the following story in the historical novel *Romance of the Three Kingdoms*, one of the greatest classics in Chinese literature: After suffering his defeat at the battle of Red Cliffs, the warlord Cao Cao made his escape to a crossroads where the main path was wide and flat but longer than the other treacherous path which led to Huarong. The scouts reported to Cao Cao that smoke was seen rising from the Huarong trail suggesting an ambush. Cao Cao laughed: “I know Zhuge Kongming (the opponent strategist) so well. Everything he did was intended to deceive me. Thus the apparent truth must be a deception. The smoke seems to be signalling an ambush but it must be the enemy’s decoy to lure me to the main road.” He then ordered his men into the Huarong Trail, only to be trapped there by the ambush. In fact, knowing Cao Cao so well, Zhuge Kongming had guessed how Cao Cao would reason, and taking this into account, he still outsmarted him, and the smoke lured Cao Cao into the ambush. Ironically, just like what Cao Cao said, everything Kongming did was intended to deceive him, and what seemed to be the truth to Cao Cao was indeed a deception.

As we have seen from the above stories, protocol and knowledge have an intricate and dynamic relation with each other, which deserves careful study, and is the starting point of this dissertation:

### “Epistemic Modelling and Protocol Dynamics”

## 1.1 Background

**Epistemic Protocols** Protocols that involve reasoning about knowledge have been studied, under the name of *knowledge-based programs*, since the pioneering work of [HF89] and [FHMV97] in the setting of *Interpreted Systems (IS)* [FHMV95](or, equiv-

alently *Epistemic Temporal Logic* (ETL) [PR85]). Research within this framework has revealed that protocols with knowledge tests (e.g., **if**  $K_{ip}$  **then do**  $a$ ) are essentially more complex than standard programs [HF89, San91, Hal00]. Given an initial setting, a knowledge-based program may be represented by none or more than one interpreted system while a standard program induces a unique interpreted system. [MDH86, Hal87] showed that knowledge may help to develop efficient algorithms but the verification problem is quite involved. In the ETL framework, [PR03] gave a semantics of actions based on protocols which fleshes out the intuition that protocols let actions carry information as we remarked at the beginning of this chapter (see also [BS97] for a more general treatment).

Knowledge-based programs can be generalized to *epistemic protocols* which not only allow knowledge tests but also actions with epistemic effects, e.g. public announcements. Such actions are studied as objects in their own right in *Dynamic Epistemic Logic* (DEL) where actions and their epistemic effects are handled by epistemic event models and the built-in update mechanism [Pla89, GG97, BMS98]. During the last decade DEL has been successfully applied to a variety of scenarios from knowledge puzzles to social norm changes [vDvdHK07], due to its flexibility in modelling various epistemic interactions among agents.

Despite some informal protocols featured in the studies of epistemic puzzles (see, e.g., [vD03, AvDR09, VO07, vD08, DvEW10]), the epistemic protocols have not been formally studied as a central issue in the DEL framework until recently. Aiming at merging the temporal aspect of ETL and the dynamic epistemic aspect of DEL, a series of work has been done with extra protocol information provided to the epistemic models [HY09, vBGHP09, Hos09, HP10b] (see also [Hos10] for a survey). A *DEL-protocol* defined in this line of work is a set of sequences of DEL events (pointed event models [BMS98]) closed under finite prefix, similar to the definition of the protocol of [PR03] in an ETL setting. Moreover, a notion of *state-dependent protocols* is introduced in [vBGHP09], which allows different states in a given epistemic Kripke model to have different DEL protocols. In such a set up, the protocol at the real world may not be common knowledge. Given a DEL protocol and an initial model, we can generate a *unique* ETL-like model capturing both the epistemic dynamics and the *protocol information* as [Hos10] puts it.

However, as remarked in [PR03], an *explicit* set of sequences of events, as in the case of the DEL protocols mentioned above, is an *extensional* notion of the *common sense* protocols which are usually specified by a few rules governing the communications. To formally study epistemic protocols, in particular to address their verification problems, an epistemic protocol specification is preferably high-level, finitely representable and independent from the models. Note that the verification of an epistemic protocol can be tricky. Take the following classic example used in DEL literature: *the Russian Cards Problem* (RCP) (introduced to DEL by van Ditmarsch [vD03]):

**1.1.1. EXAMPLE. (Russian Cards Problem (RCP<sub>(n,n,k)</sub>))**  $2n + k$  cards are distributed randomly to three agents  $\{A, B, E\}$  such that agent  $A$  has  $n$  cards,  $B$  has  $n$  cards, and  $E$  has  $k$  cards. Now  $A$  and  $B$  want to inform each other their hands by public announcements, without

revealing his cards to  $E$ . Is it possible?

Ω

Now let us consider a simple case:  $RCP_{2,2,1}$  where the cards are denoted as numbers (0 – 4). A “promising protocol” for  $A$  to let  $B$  know  $A$ ’s cards without letting  $E$  know any card of  $A$  is that:  $A$  announces the disjunction of his actual hand (say 01) with all the different combinations of the remaining cards, so he would announce “I have 01 or 23 or 24 or 34.” Since  $B$  has one more card than  $E$  he can eliminate all of 23, 24 and 34, while  $E$  can only eliminate two of 23, 24 and 34. However, it does not work like this any more if  $E$  knows that the protocol is meant to reveal  $A$ ’s hand to  $B$ . Assume that  $E$  has 3. Then after the announcement by  $A$ ,  $E$  will know that  $A$  has either 01 or 24. Now  $E$  can perform the following reasoning: suppose that  $A$  has 24 and  $B$  has 01. Then  $B$  could not have learnt  $A$ ’s hand from  $A$ ’s announcement. So  $E$  can infer that  $A$  has 01. Another way to see that the would-be protocol is wrong is as follows. Suppose the protocol is commonly known, e.g., the procedure to generate the announcements is known to both  $A$  and  $E$ . Note that in the above case this procedure is a function from hands of two cards  $x, y$  to announcements  $f(xy) =$  “I have  $xy$  or  $z_1z_2$  or  $z_2z_3$  or  $z_1z_3$ .”, where  $z_1, z_2, z_3$  are the remaining 3 cards other than  $x, y$ . This function is injective, so the announcement reveals the hand immediately.

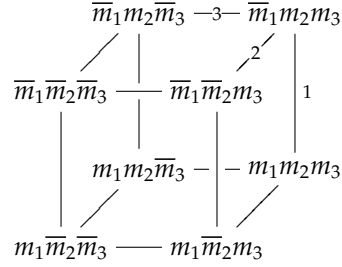
As demonstrated by the above example and many others mentioned in [vD03, vDvdHK07], a notable feature of epistemic protocols, compared to usual communication protocols, is that the correctness of the epistemic protocols heavily relies on the assumptions of the agents’ *meta-knowledge* about the protocol itself. It is reasonable to assume that the protocol and its goals are commonly known by all the agents including possible adversaries, if we want to apply the protocol repeatedly in real life cases. To check the correctness of protocols under the assumption that the protocol is commonly known, formalization of protocols is clearly imperative.

**Dynamic Epistemic Modelling** As in the formal verification of communication protocols, we would like to apply *model checking* to the verification of epistemic protocols, based on a logical language which can specify both the protocol and its goal. However, as observed in [FHMV97], a protocol involving knowledge preconditions should be verified w.r.t. the assumptions about the initial situation, e.g., to verify a protocol for  $RCP_{2,2,1}$  on a model with only two agents  $A, B$  does not make sense. However, two natural questions arise: how do we specify the assumptions and based on these assumptions, how do we generate a correct model to be checked? Let us now look at another classic puzzle in DEL and ETL (see e.g., [FHMV95, vDvdHK07]):

**1.1.2. EXAMPLE. ( $n$ -Muddy Children)** *Out of  $n$  children,  $k \geq 1$  got mud on their foreheads while playing. They can see whether other kids are dirty, but there is no mirror for them to discover whether they are dirty themselves. Then father walks in and says: “At least one of you is dirty!” Then he requests “If you know you are dirty, step forward now.” If nobody steps forward, he repeats his request: “If you now know you are dirty, step forward now.” After exactly  $k$  requests to step forward, the  $k$  dirty children suddenly do so.*

Ω

The changes of the children’s knowledge in this classic scenario are “perfectly” modelled by the update mechanism of public announcements on an initial Kripke model, usually in the following shape:



where  $\{1, 2, 3\}$  is the set of 3 children,  $m_i$  denotes the proposition that  $i$  is muddy and  $\bar{m}_i$  denotes its negation. The labelled equivalence relations model children’s epistemic accessibility relations ( $s \longleftrightarrow_i t$  means at state  $s$ ,  $i$  thinks  $t$  is possible).

As remarked in [vB09]: *There is no algorithm for producing it, but most people would agree that it fits the situation.* We assume that people, even non-logicians, would be able to “read off” the information from the graph representation e.g. “In any case, one agent does not know whether he is dirty or not, but he is sure about the other two.” In epistemic logic, it amounts to a conjunction  $\phi_1 \wedge \phi_2 \wedge \phi_3$  where  $\phi_1 = (K_1 m_2 \vee K_1 \bar{m}_2) \wedge (K_1 m_3 \vee K_1 \bar{m}_3) \wedge \neg(K_1 m_1 \vee K_1 \bar{m}_1)$  and similar for  $\phi_2$  and  $\phi_3$ . This suggests that we may translate an informal initial setting into a set of logical formulas and try to generate a correct model from this set of formulas.

In the context of dynamic epistemic modelling, [vDvdHK03a] demonstrates that there are intuitive epistemic formulas (*descriptions*) that characterize the initial models in the case of the card games. However, in general, a set of formulas translated from an informal description of the scenario may not have a unique model. In many cases, the informal assumptions in our mind can not be made fully precise. Even if the initial specification induces a model, we still need a method to generate it.

Here we may seek insights from computer science. A useful approach to represent models is the so-called *operational semantics* used for process algebra (e.g. CSP of [Hoa85]), where the model of a process term is generated by the operational rules on its subterms. A similar idea, *Tableau* [Pra80, SE89], appeared in logic as a method for solving the satisfiability problem of logics. Another inspiration is from the ETL framework where models are generated by composing local states of each agent. In DEL [vD02] made an early attempt to program epistemic actions while [vDvdHK03a] imported the idea of interpreted system in the specific context of card games.

**Model Checking** During the last three decades, (temporal logic) *model checking* has become a prominent application of logic in computer science (see [CGP99] for an extensive survey). We would like to apply model checking for epistemic protocols as attempted in [vDRV05, vE07, vD03]. However, when dynamic epistemic modelling is applied to complex situations, very large (even infinite) epistemic

models or event models are inevitable (see, for example [DW07]). The verification of certain properties may require the (exhaustive) exploration of such large models. In computer science, this problem is known as the *state space explosion* problem. Various methods have been proposed to handle this problem in temporal logic model checking. A very successful one in practice is the *symbolic model checking* technique initiated by McMillan et.al [BCM<sup>+</sup>92], which boosted the capability of model checking on large system enormously (see, for example [BCM<sup>+</sup>92], where more than  $10^{20}$  states are handled in some case studies). Despite the success of BDD-based symbolic model checking and the more recent development of bounded model checking using SAT-solvers (see, e.g., [CBRZ01, McM02]), the state explosion problem still remains a major hurdle to model checking real life complex systems. To reduce the state space, many approaches have been developed, for example, symmetry reduction [CEFJ96, ES96, ID96, SG04], partial order reduction [GPS96, Pel93], abstract interpretation [CC77], and abstraction-refinement methods [CGL94, CGJ<sup>+</sup>03, GHJ01, SG08]. Among such approaches, the abstraction-refinement method is considered to be the most general and flexible one; also it is fully automated [CGJ<sup>+</sup>03]. However, such techniques have not been introduced to the epistemic setting until recently [DOW08, CDLR09, CLDQ09].

## 1.2 Overview of the Dissertation

The general storyline of the dissertation is as follows: In Part I, we introduce logics to specify epistemic protocols including their goals and their dynamics. The verification problem can then be formalized as a model checking problem within a unified logical framework. To perform model checking we need to develop methods for finding/generating epistemic models, and this problem is addressed in Part II. Part III introduces abstraction techniques that are particularly useful on making the model checking more efficient in the epistemic setting. In Part IV we survey the application of epistemic analysis on protocols in a setting of security protocol verification.

The contributions of each chapter are briefly summarized as follows:

In Chapter 2: *Preliminaries*, we list the basic definitions used throughout this dissertation.

### Part I

Chapter 3: *Meta-knowledge Matters* departs from the existing discussions about protocols in DEL by introducing a logic which can specify both the epistemic protocols (by regular expressions) and their goals *inside the language*. By formally defining the epistemic protocol specification and their verification problems under the assumption of the meta-knowledge about the intended goal, we flesh out the remarks about the subtleties of epistemic protocol verification. Based on this framework, we discuss

how to find and verify deterministic epistemic protocols for the classic Russian Card Problem  $RCP_{3.3.1}$ .

In Chapter 4: *Logics of Knowledge and Protocol Change*, we address the question: “how people get to know a protocol?” by developing three logics which are convenient for reasoning about knowledge and protocol changes with different perspectives. With various *protocol announcement modalities*, we can handle the dynamics of protocols and formalize how the protocols let the actions carry new meanings. We show that all the three logics we introduced can be translated back to PDL on standard Kripke models, thus the techniques of modelling and model checking we developed in the other parts of the dissertation can be applied to these logics.

### Part II

We then turn to the issues of modelling in Chapter 5: *Composing Models*. We propose new composition operations on static and event models with arbitrary vocabularies, aiming at a compositional method for generating initial epistemic models. We prove some decomposition theorems w.r.t. our new operator and demonstrate the use of our methods by various examples. Algebraic properties linking the new operator to standard product update are also addressed.

In Chapter 6: *Counting Models*, we report some results on counting the number of different models given a finite set of initial assumptions. Restricted to image-finite models, we show that if a modal  $\mu$ -calculus formula has an infinite model modulo bisimulation then it has  $2^{\aleph_0}$  (cardinality of the continuum) different models modulo bisimulation. On the other hand, if it does not have any infinite models then all its models can be represented in a normal form.

### Part III

A 3-valued semantics for public announcement logic is defined and studied in Chapter 7: *Making Models Smaller* to facilitate abstractions of models for logic with dynamic modalities. We define a relation with vocabulary and agent mappings between concrete models and their abstractions, thus making it possible to also abstract the signatures of models. It is particularly applicable in an epistemic setting where agents are usually similar to each other. We then give a logical characterization of the abstraction relation thus showing it is safe to check properties on the abstract model instead of the original concrete model.

Chapter 8: *Accelerating the Transitions* studies the PDL on so-called *accelerated Kripke models* where the transitions in the models are labelled by regular expressions in order to obtain informative abstractions. By making use of a technique of regular expression rewriting, we analyse the complexity of the model checking and satisfiability problems of this logic and give a complete axiomatization.

### Part IV

Chapter 9: *Epistemic Approaches to Security Protocol Verification* surveys the epistemic approaches to security protocol analysis. We summarize the most important techniques in the ETL and DEL approaches to security protocol verification, and compare

these two approaches in term of convenience. We argue that some security properties can only be faithfully formalized by temporal logic with knowledge operators, but are not expressible by standard temporal logic. However, we need to pay some cost in model checking complexity, in exchange to the expressiveness we gain by using ETL.

### 1.3 Origins of the Material

The material that forms the main body of this dissertation is based on collaborations with various people: Chapter 3 extends a joint paper with Lakshmanan Kuppusamy and Jan van Eijck [WKvE09]; Chapter 4 is based on an unpublished manuscript; Chapter 5 is an elaborated version of joint work with Jan van Eijck and Floor Sietsma [vEWS10]; Chapter 6 is an extension of a discussion note with Floor Sietsma; Chapter 7 reports joint work with Francien Dechesne and Simona Orzan [DOW08]; Chapter 8 is an updated version of a paper with Taolue Chen and Jaco van de Pol [CvdPW08]; and Chapter 9 is based on a joint paper with Francien Dechesne [DW10].

Some papers related to the general topic of this dissertation are not included in the above chapters. I mention them here as pointers for further reading. With Francien Dechesne, I explored the possibility of using DEL for security protocol verification, as reported in [DW07]. This work also motivated the writing of the material constituting Chapter 9 where the essential ideas of [DW07] are summarized and compared to other approaches. Note that in this dissertation, we focus on knowledge but not belief while in joint work [vEW08] with Jan van Eijck we study a PDL-style DEL as a belief revision logic, which in the end leads to the use of PDL as a protocol logic in Chapter 3 and Chapter 4. Together with Floor Sietsma and Jan van Eijck, I designed a flexible logical framework for reasoning about communications over networks [WSvE10], which combines the dynamics of protocols as in Chapter 4 and the modelling advantages of ETL and DEL respectively. A game theoretical perspective of protocol execution is missing in the current dissertation. However, interested readers may have a look at joint work [TDW08] with Mohammad Dashti which presents a game theoretical analysis of exchange protocols with untrusted third parties. In the end, if the reader prefers a more entertaining introduction to (security) protocols than Chapter 9, she/ he may want to look at [DvETW09, DETW09] written by Francien Dechesne, Jan van Eijck, Wouter Teepe, and me.