Chapter 4 MULTI-AGENT DYNAMIC-EPISTEMIC LOGIC

The public announcements or observations *!P* studied in Chapter 3 are an easily described way of conveying information. The only difficulty may be finding good static epistemic models for the initial situation where updates start. But information flow gets much more challenging once we give up the uniformity in this scenario. In conversation, in games, or at work, agents need not have the same access to the events currently taking place. When I take a new card from the stack in our current game, I see which one I get, but you do not. When I overhear what you are telling your friend, you may not know that I am getting that information. At the website of your bank, you have an encrypted private conversation which as few people as possible should learn about. The most enjoyable games are all about different information for different players, and so on. In all these cases, the dynamics itself poses a challenge, and elimination of worlds is definitely not the right mechanism. This chapter is about a significant extension of *PAL* that can deal with partially private information, reflecting different *observational access* of agents to the event taking place. We will present motivating examples, develop the system *DEL*, and explore some of its technical properties. As usual, the chapter ends with some further directions and open problems, from practical and philosophical to mathematical. *DEL* is the true paradigm of dynamic-epistemic logic, and its ideas will return over and over again in this book.

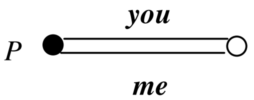
4.1 Multi-agent information flow

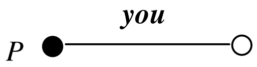
Here is what may be the simplest scenario that goes beyond Chapter 3, from a brochure for my Spinoza project in 1998, written to explain dynamic logic to university officials:

Example Two envelopes.

> We have both drawn a closed envelope. It is common knowledge between us that one holds an invitation to a lecture on logic, the other to a wild night out in Amsterdam. Clearly, we are both ignorant of the fate in store for us. Now I open my envelope, and read the contents, without showing them to you. Yours remains closed. Which information has passed exactly through my action? I know now which fate is in store for me. But you have also learnt something, viz. that I know – though not what I know. Likewise, I did not just learn what is in my envelope. I also learnt something about you, viz. that you know that I know. The latter fact has even become common knowledge between us. And this may not even be all.

What is a general principle behind this? Intuitively, we need to *eliminate links* here instead of worlds. The initial information state is one of collective ignorance, and the update removes my uncertainty link – with both worlds still needed to model your ignorance:



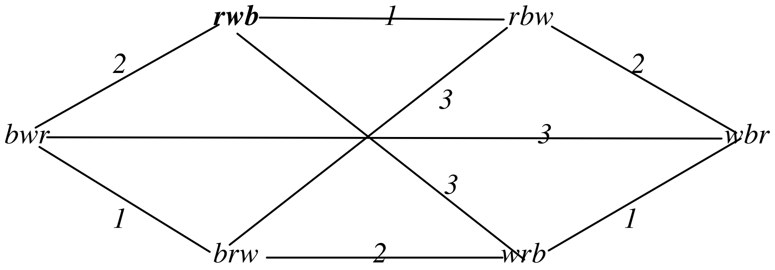


In the resulting model, all observations from our story come out right. ■

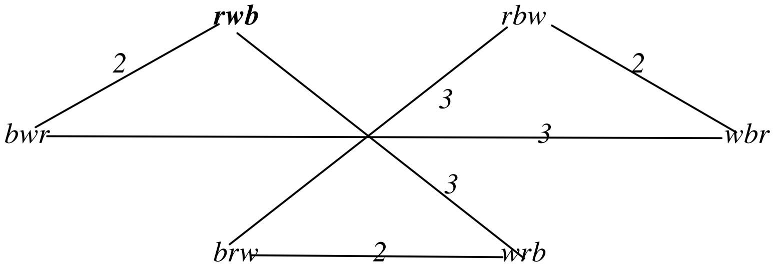
This kind of update occurs in many games. Van Ditmarsch 2000 has many examples.

Example Partial showing in the Three-Cards game.

The three players have drawn their single card each, as in the earlier diagram:



Now player *2* shows his card in public, but *only to player 1*. The resulting update publicly removes the uncertainty lines for player *1*, but not for players *2* or *3*:



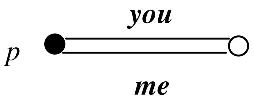
■

Games like this may not look serious, but they provide a sort of *normal form* for many informational scenarios. Here is another phenomenon all around us:

Example *cc* and *bcc* in email.

Sending an email message to someone with a list of people under the *cc* button is like making a public announcement in the total group, assuming the computers work perfectly. But what does it mean to send a message with a *bcc* list? Now these people receive the message, but the others do not realize that, or, if they understand the system, they may think that there might be undisclosed recipients, but they cannot be sure. Scenarios like this do not work with link cutting, as it may be necessary to *add new worlds*. ■

Here is a simplest situation where a similar phenomenon happens. Initially, we are both ignorant whether p is true. A simplest model for this will again look as follows:



Now you hear a public announcement that p, but you are not sure whether I heard it, or I just thought it was a meaningless noise. In this case, intuitively, we need to keep two things around: one copy of the model where you think that nothing has happened, and one updated copy for the information that I received. This requires at least *3* worlds, and hence we need a new update mechanism that can even increase the number of worlds.

Most communication has private aspects, inadvertently or with deliberate hiding. Social life is full of procedures where information flows in restricted ways. Such diversity is highlighted in games, that manipulate information flow and are an enjoyable training ground for social skills. [72] These phenomena pose real modeling challenges. It is not easy to write epistemic models with the right features by hand, and we need a systematic style of thinking. The usual practical art of modeling has to become a bit of a science.

It is deep! Before proceeding, I want to address a concern. Some colleagues see games and communication on a par with small-talk and gossip, [73] forced upon us by living in a crowd on this planet, but devoid of the lonely depth and beauty of truth and proof. In contrast, I think that multi-agent differences in information and observation are so crucial to human life, and the emerging social structures so essential, that it is *this* we want to understand in greater depth. It is easy to be rational when alone, but the greatest feats of the mind unfold in interaction. Social structure is not a curse, but constitutive of who we are as humans.

We will now analyze differential observational access, making epistemic logic much livelier. Other key logical features of multi-agency will unfold in subsequent chapters.

4.2 Event models and product update

The general idea in what follows is that agents learn from observing events to which they

---

[72] The system of this chapter has been applied to fully describe parlour games like *Clue*.

[73] But see Dunbar 1998 on the beauties and positive power of gossip.

may have different access. An important step toward such a calculus was taken in Gerbrandy 1999A, using non-well-founded sets to model private announcement in subgroups: a subgroup gets informed while the rest learns nothing. Baltag, Moss & Solecki 1998 added the crucial idea of observational access via 'event models' to create the much more general system in use today. [74] A textbook is van Ditmarsch, van der Hoek & Kooi, 2007, but our presentation will be self-contained, with some deviant aspects. The resulting system of *dynamic-epistemic logic* (*DEL*) provides a general account of multi-agent update of epistemic models, with some new features that are of interest in their own right.

For a start, the information models provided by epistemic logic have a natural companion, when we look at the events involved in scenarios of communication or interaction. Epistemic events can be treated much like epistemic states:

Definition Event models.

An *epistemic event model* is a structure $\boldsymbol{E} = (E, \{\sim_i\}_{i\in G}, \{Pre_e\}_{e\in E}, e)$ with a set of *events* E, epistemic uncertainty relations $\sim_i$ for each agent, [75] a map assigning *preconditions* Pre_e to events e, stating just when these are executable, and finally, an *actual event* e. [76] ■

Event models are like static epistemic ones in endowing the events affecting our current model *themselves* with epistemic structure. Agents' uncertainty relations encode which events they cannot distinguish between in the scenario, because of observational limitations. When I open my envelope, you know it must be either Lecture or Night-Out, but you cannot distinguish the two events of 'my reading *L*' and 'my reading *N*'. Thus, both are in the event model, though not events we all consider irrelevant, like Mount Etna erupting. Next, the event model carries no propositional valuation, but instead, events come with *preconditions* for their occurrence. A public announcement *!P* of a fact *P* presupposes truth of *P*, reading Night-Out has a precondition that this is the real card in my envelope, my asking a genuine question meant I did not know the answer. Most events carry information about when and where they occur. That is why they are informative!

---

[74] For an up-to-date survey of themes and contributions, cf. Baltag, van Ditmarsch & Moss 2008.

[75] Accessibilities will often be equivalence relations, but we emphatically include event models with directed minimal accessibility for later scenarios of 'misleading' where knowledge gets lost.

[76] As with actual worlds in epistemic models, we do assume that one event actually takes place.

This feature is at the heart of the following update mechanism, that describes how new worlds, after update, become pairs of old worlds with an event that has taken place:

Definition Product update.
For any epistemic model ***M**, s* and event model ***E**, e,* the *product model* ***M x E**, (s, e)* is an epistemic model with the following main components. Its domain is the set $\{(s, e) \mid s$ a world in ***M***, e an event in ***E*** and ***M***, $s \models Pre_e\}$, and its accessibility relations satisfy

$$(s, e) \sim_i (t, f) \text{ iff } both\ \ s \sim_i t\ \ and\ \ e \sim_i f,$$

Finally, the valuation for atoms p at (s, e) is the same as that at s in ***M***. ■

Explanation The new model uses the Cartesian product of the old model ***M*** and the event model ***E***, and this explains the possible growth in size. But some pairs are filtered out by the preconditions, and this elimination makes information flow. The *product rule* for epistemic accessibility reflects a simple idea: we cannot distinguish two new worlds if we could not distinguish them before and the new events cannot distinguish them either.

The above definition is incomplete: we have not specified *what language the preconditions come from*! We will assume here that they are in our epistemic language, either as atomic propositions ('I have the red card'), or as more complex epistemic ones ('I think you may know whether *P*'). Later on, we will allow the precondition language to contain dynamic update modalities, [77] but this feature is not required in the examples to follow. Finally, the intuitive understanding is that the preconditions are common knowledge in the group. Many of these assumptions can be lifted eventually to make the framework more general, but such variations will be obvious to readers who have grasped the simpler base case.

This mechanism can model a wide range of scenarios, keeping track of surprising changes in truth value for propositions generalizing the 'self-refuting' announcements of Chapter 3. We only give some simple examples here.

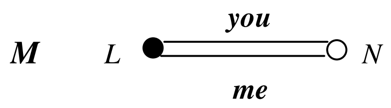
Example Public announcement.
The event model for a public announcement $!P$ has just one event with precondition P, and reflexive accessibility arrows for all agents. The product model ***M x E*** then just retains the pairs $(s, !P)$ where ***M***, $s \models P$, thus producing an isomorphic copy of our earlier ***M***$|P$. ■

---

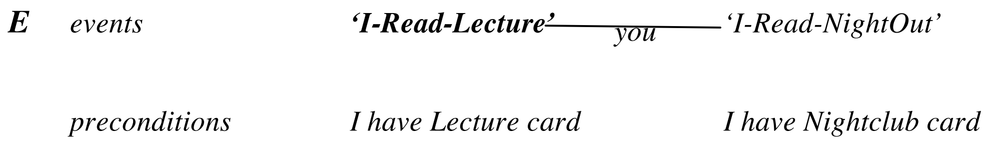
[77] Technically, this requires a simultaneous recursion with some of the definitions to follow.

Example The two envelopes.

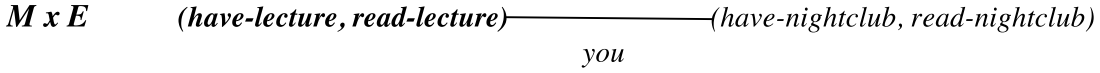
Our initial epistemic model was like this (‘*L*’ is my having the lecture, ‘*N*’ the Nightclub):



You could not distinguish between the two events of my reading ‘Lecture’ and my reading ‘NightOut’ – and this shows in the following event model, where the black event is actual:



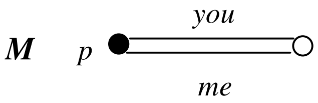
We also assume all reflexive arrows for both agents. In the resulting product model ***M x E***, of the *4* possible pairs, *2* drop out by the preconditions. But this time, we do not just copy the original model: the product rule performs link-cutting, yielding the right outcome:


M x E
(have-lecture, read-lecture)
(have-nightclub, read-nightclub)
you

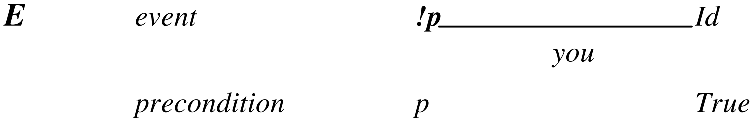
■

Example The doubtful signal.

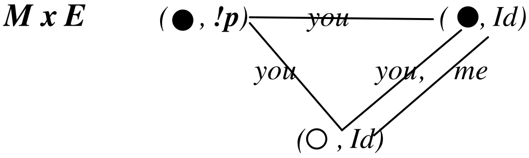
Here is how model size can increase. Consider our earlier model



Now take an event model for the earlier scenario where I hear *!P*, but think that you may have witnessed just the trivial identity event *Id* that can happen anywhere:



This time, the product model ***M x E*** has *3* worlds, arranged as follows:



[78] The reader can see how this satisfies our intuitive expectations described earlier. ■

Product update in realistic scenarios can explode size. Ji 2004 analyzes this for *bcc* in emails, whose complexity we feel when trying to track who knows what with this device. Van Ditmarsch 2000 analyzes model changes in the parlour game of *Clue,* that are much more delicate than our simple examples might suggest. [79] All this points to applications in the real-world setting of *security*: cf. van Eijck, Sietsma & Wang 2009.

4.3 Exploring the product update mechanism further

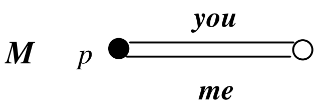
Product update provides something that is rare in logical semantics: a systematic mechanism for *constructing* and maintaining the right models. We will look at its theory later on. But descriptively, it is of interest in that it can be used to classify types of scenarios in communication. For instance, here is one important boundary:

Under-informed versus misinformed When you buy a board game, the rules printed on the cardboard cover may make things complicated, but they will not *necessarily* mislead you. Likewise, *bcc* in email will not mislead you if you are aware that it is available. There are degrees of public ignorance, and product update helps classify them. For instance, public resolution of questions does not increase domains, but cuts links (cf. Chapter 6).

Arrows of belief In contrast with this, an illicit secret observation of a card will mislead you, inducing a false belief that nothing happened: it crosses a threshold toward *cheating*. Product update can also model this, using *pointed accessibility arrows* indicating what the current world could be like according to you. In particular, that world itself is not among these when you are mistaken. This issue will return in Chapter 7 on belief models and belief change, but for now, we just give one concrete illustration:

Example A secret peek.

We both do not know if P, but I secretly look and find out.

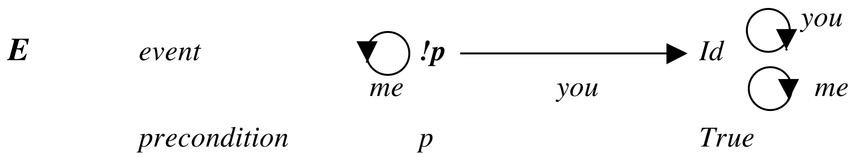


Here is the event model, with the pointed arrows as just indicated:

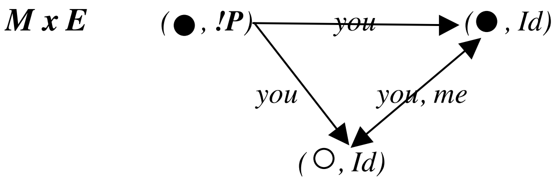
---

[78] The actual world in ***M* x *E*** Is on the top left: it has the old actual world plus the actual event.

[79] In this practical setting, computer programs that perform update and related symbolic tasks are a useful tool, such as the *DEMO* system of Jan van Eijck: http://homepages.cwi.nl/~jve/demo/.



We get a product model ***M x E*** with *3* worlds, as in the following picture. For convenience, we leave out reflexive arrows (one for me in the real world, two for us both in the others):



In the actual world, I know exactly what has happened – while you still think, mistakenly, that we are in one of the other two worlds, where everything is just as before. ■

Thick events and agent-types So far, we had simple physical events, but the framework allows any type of event. This extends its power to much more complex cases, where we need to distinguish *agent types*, such as the beloved ones from logic puzzles, where one must separate Liars (who always lie) from Truth-Tellers (who always speak the truth).

Example Liars versus truth tellers.

You meet someone who tells you that *P*, but you do not know if she is a Liar, or a Truth-Teller. You hear her say that *P* is the case, which you know to be true. Here is how product update tells you that the person is a Truth-Teller. One forms new *pair events* of the form (agent type, proposition announced), with the following preconditions: $Pre_{(Truth\text{-}Teller,\ !P)} = P$, $Pre_{(Liar,\ !P)} = \neg P$. Only the first can happen in the actual world, and so you know that you are meeting a Truth-Teller. Of course, in general, more events will qualify, but the point is that we can encode precisely what we know about the agent types in events like this. ■

Varying kinds of agents Which type of agent is hard-wired in product update? *DEL* says that uncertainties only arise from earlier uncertainties: this is perfect *memory*. It also says that old uncertainty can be removed by observation, giving agents powers of *observation* encoded in the event model. We will return to these features in Chapter 11. This does raise the interesting issue how to modify *DEL* product update for other agents, say, with defective memory. Here is an illustration from van Benthem & Liu 2004: [80]

---

[80] Van Ditmarsch & French 2009 have a more sophisticated account of *acts* of forgetting.

Example Update with memory-free agents.

Memory-free agents go by the last observation made. Thus, we can change their product update rule to read simply $(s, e) \sim (t, f)$ iff $e \sim f$. Liu 2008 has more discussion, including modeling agents with k-cell bounded memories for any number k (cf. Snyder 2004). ■

Real world change *DEL* so far assumed that atomic facts do not change in the process of getting information. This fits with its emphasis on *preconditions*. But actions can also *change* the world, as described by their *postconditions*. When the latter are simple, it is easy to incorporate change. In event models, now let each event e have a precondition Pre_e for its occurrence as before, but also an *atomic post-condition* $Post_e$ that assigns to each event e the set of atoms true after an occurrence of e. Equivalently, one can record this same information as a *substitution* $\sigma^E{}_e$ from proposition letters p to either p, or $\neg p$.

Definition Product update with change in ground facts.

The product model of an epistemic model $\boldsymbol{M}$ and an event model $\boldsymbol{E}$ is formed exactly as before, with just this change in the propositional valuation: $(s, e)\ V_{\boldsymbol{MxE}}(p)$ iff $p \in Post_e$.[81] ■

Why must we exercise care with the form of postconditions? The reason is that there is no canonical update method for making complex epistemic postconditions φ true. There may even be paradoxes in specifications, where the very act of bringing about φ blocks it.

4.4 Language, logic, and completeness

Given our extensive study of *PAL*, the set-up for the dynamic logic *DEL* turns out familiar. The only novelty is the use of event models inside modalities, which takes some guts:[82]

Definition *DEL* language and semantics.

The *dynamic epistemic language* is defined by the following inductive syntax rules:

$$p \mid \neg\varphi \mid \varphi\wedge\psi \mid K_i\varphi \mid C_G\varphi \mid [\boldsymbol{E}, e]\,\varphi$$

where $(\boldsymbol{E}, e)$ is any event model with actual event e. There is a recursion here, since preconditions in event models come from the same language.[83]

---

[81] Van Benthem, van Eijck & Kooi 2006 give a complete logic for this extended system.

[82] In practice, one often uses finite event models, though this is not strictly needed. We might then use abstract representations of event models to make things look more linguistic.

The semantic interpretation is standard for all clauses, except for the dynamic modality:

$$\boldsymbol{M}, s \models [\boldsymbol{E}, e]\varphi \quad \text{iff} \quad \text{if } \boldsymbol{M}, s \models Pre_e, \text{ then } \boldsymbol{M} \mathbf{x} \boldsymbol{E}, (s, e) \models \varphi. \qquad \blacksquare$$

This is a watershed. If you are down-to-earth, you will find a syntax putting models in a language weird, and your life will be full of fears of inconsistency. If you are born to be wild, you will see *DEL* as a welcome flight of the imagination.

Next, axiomatizing this language goes by the *PAL* methodology of recursion axioms:

*Theorem (*Baltag, Moss & Solecki 1998*)* *DEL* is effectively axiomatizable and decidable.

Proof The recursion axioms are like those for *PAL*, with the following adjustments: [84] [85]

$$[\boldsymbol{E}, e]q \quad \leftrightarrow \quad (Pre_e \rightarrow q)$$

$$[\boldsymbol{E}, e]K_i\varphi \quad \leftrightarrow \quad (Pre_e \rightarrow \bigwedge{}_{e \sim i\, f\, in\, \boldsymbol{E}}\, K_i[\boldsymbol{E}, f]\varphi.$$

The latter axiom follows if we recall that the accessible worlds in ***M x E*** are those whose world component is accessible in ***M***, and whose event-component is accessible in ***E***. [86] The over-all completeness argument about reducing arbitrary dynamic-epistemic formulas to equivalent ones in the epistemic base language runs as before for *PAL*. ■

One can easily adapt these axioms to deal with the two earlier-mentioned extensions. With substitutions for world change, one amends the atomic case to

$$[\boldsymbol{E}, e]q \quad \leftrightarrow \quad (Pre_e \rightarrow \sigma^{\boldsymbol{E}}{}_e(\,q))$$

And for memory-free agents, one amends the recursion axiom for knowledge to

$$[\boldsymbol{E}, e]K_i\varphi \quad \leftrightarrow \quad (Pre_e \rightarrow \bigwedge{}_{f \sim i\, e\, in\, \boldsymbol{E}}\, U[\boldsymbol{E}, f]\varphi)$$

where *U* is the universal modality over all worlds in the model ***M***.

---

[83] *Caveat*. Preconditions with dynamic modalities referring to other event models might introduce circularities in the system. This is avoided by viewing the preconditions as literal *parts of* event models, so that the only event models referred to must be 'smaller' in some sense.

[84] It is a useful exercise to check that the *PAL* axioms are special case of the following ones.

[85] We omit common knowledge here, for which we give a treatment later.

[86] *Caveat*. If the event model is infinite, this axiom may have an infinitary conjunction over all $f \sim_i e$. The axioms remain finite if the event models are only *finitely branching*.

4.5 Conclusion

Event models are a new style of modeling, drastically increasing the analytical powers of dynamic epistemic logic. One can now state systematically what agents know at successive stages of realistic scenarios like games, or any systems with restricted information flow, through a calculus of systematic model construction.

In principle, *DEL* still behaves as with public announcements, but interested readers will see how, in the rest of this chapter, it raises new issues of its own. These include iterated update over longer-term temporal universes and extensions to group knowledge. We now continue with this, and at the end, we formulate a number of open problems linking *DEL* to other computational and mathematical frameworks in perhaps surprising ways.

4.6 Model theory of product update: bisimulation and update evolution

We study the semantic behaviour of product update a bit more, to get a better sense of what this mechanism of model change does. For a start, consider a traditional issue:

Preserving special logics Suppose that ***M*** belongs to a special model class for an epistemic logic defined by some frame condition on accessibility, and ***E*** satisfies the same relational constraint. Will the product model ***M x E*** also fall in this class? This is definitely the case for the extremes of *K* (no constraint) and *S5* (equivalence relations), but things are much less clear in between. In fact, things get worse than with public announcements. There, update always goes from models to sub-models, and so we observed that all logics whose characteristic frame conditions are *universally definable* are preserved. But with product update, we only have the following observation from general model theory:

Fact All *universal Horn formulas* are preserved under product update,
i.e., all first-order formulas of the form $\forall x_1 \dots \forall x_k (A \rightarrow B)$,
where *A, B* are conjunctions of atoms.

This is easy to prove given the conjunctive nature of product update. There is a general result in first-order logic that a formula is preserved under sub-models and direct products iff it definable by a conjunction of universal Horn formulas. It is an open problem which precise syntactic class of first-order formulas is preserved under product update.

Example Properties non-preserved under product update.

The preceding fact excludes existential properties such as *succession*: $\forall x\, \exists y\, Rxy$. Here is how this can fail. Take a two-world model ***M*** with $x \rightarrow y$, and a two-event model ***E*** with *e*

$\rightarrow f$, where x satisfies Pre_f and y satisfies Pre_e. The pair (x, f) has no successor in $\boldsymbol{M} \times \boldsymbol{E}$, as world and event arrows run in opposite directions. But universal frame conditions with disjunction can also fail, such as *connectedness* of the ordering. In fact, the preceding counter-example also turns two connected orders into a non-connected one. [87] ■

Bisimulation invariance However this may be, at least, with product update, we have not left the world of basic epistemic logic, since the same semantic invariance still holds:

Fact All formulas of *DEL* are invariant for epistemic bisimulation.

Proof One proof is by the above reduction to basic *EL* formulas. More informative is a direct induction, where the case of the dynamic modalities uses an auxiliary

Fact For any two bisimilar epistemic models $\boldsymbol{M}, s$ and $\boldsymbol{N}, t$, the product model $(\boldsymbol{M}, s) \times (\boldsymbol{E}, e)$ is bisimilar with the product model $(\boldsymbol{N}, t) \times (\boldsymbol{E}, e)$. ■

This observation leads to a new notion and one more question. In Chapter 2, bisimulation was our structural answer to the question when two epistemic models represent the same information state. The present setting raises the issue when two event models are the same. One might say that two event models $(\boldsymbol{E}_1, e_1)$, $(\boldsymbol{E}_2, e_2)$ *emulate each other* when, for all epistemic models $(\boldsymbol{M}, s)$, the product models $(\boldsymbol{M}, s) \times (\boldsymbol{E}_1, e_1)$ and $(\boldsymbol{M}, s) \times (\boldsymbol{E}_2, e_2)$ are bisimilar. [88] Van Eijck, Ruan & Sadzik 2007 provide a complete characterization.

Update evolution Like public announcement, product update becomes more spectacular when *repeated* to model successive assertions, observations, or other informational events. We discussed *PAL* with iterations already in Chapter 3, and it will return in Chapter 15. *DEL* scenarios with iteration generate *trees* or *forests* whose nodes are finite sequences of announcement events, starting from worlds in the initial epistemic model. We will study *DEL*-over-time in Chapter 11, but for the moment, we look at one concrete illustration.

Product update may blow up epistemic models starting from some initial $\boldsymbol{M}$, creating ever larger models. But will it always do so? We discuss one scenario with finite extensive

---

[87] One might use such cases against product update. But one can also bite the bullet: the examples show how cherished properties for static logics no longer look plausible when we perform quite plausible dynamic model constructions. For instance, the fragility of *connectedness* has often been remarked upon in the theory of ordering – and in Chapter 9, losing it is just what we want.

[88] One can show that this holds iff $\boldsymbol{E}_1, \boldsymbol{E}_2$ produce bisimilar results on all bisimilar inputs $\boldsymbol{M}, \boldsymbol{N}$.

games of imperfect information (cf. van Benthem 2001, and Chapter 10 below). First collect all moves with their preconditions (these restrict what is playable at each node of the game tree) into one event model ***E***, that also records which players can observe which move. Now, rounds of the game correspond to updates with this game-event model:

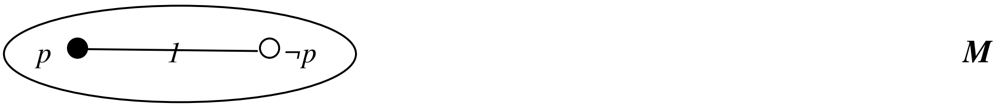
Definition Update evolution models.

Let an initial epistemic model ***M*** be given, plus an event model ***E***. The *update evolution model Tree(**M**, **E**)* is the possibly infinite epistemic tree model whose successive layers are disjoint copies of all successive product update layers ***M x E***, ***(M x E) x E***, ... [89] ■

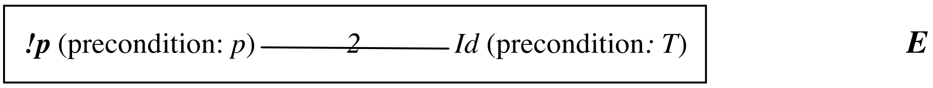
The potential infinity in this sequence can be spurious – as in those games where the complexity of information first grows, but then decreases again toward the end game:

Example Stabilization under bisimulation.

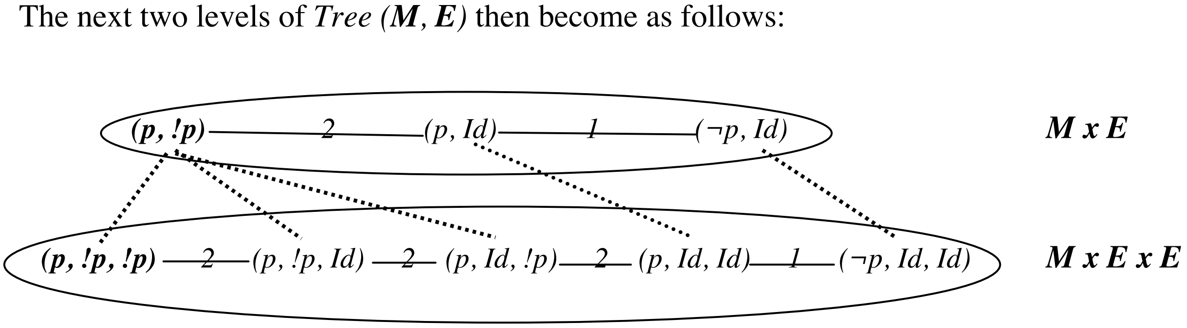
Consider an epistemic model ***M*** with two worlds satisfying *p, ¬p,* between which agent *1* is uncertain, though *2* is not. The actual world on the left has *p*.



For convenience, we read the literals *p, ¬p* as naming the worlds. Now an announcement of *p* takes place, and agent *1* hears this. But agent *2* thinks that the announcement might also just be a statement "True" that could hold anywhere. The event model ***E*** for this scenario is one we had earlier, and we now draw it as follows:



The next two levels of *Tree (**M**, **E**)* then become as follows:



---

[89] This definition will be made more precise in the epistemic-temporal logics of Chapter 11.

Now note that there is an epistemic *bisimulation* between these two levels, connecting the lower three worlds to the left with the single world ***(p, !p)*** in ***M x E***. Thus, ***M x E x E*** is bisimilar with ***M x E***, and the subsequent tree iteration is finite modulo bisimulation. ■

Update evolution also suggests that *DEL* event models may yield compact representations of complex processes. In Chapter 11, we look at the general background in branching *temporal logic*, and find precise conditions under which an extensive game with imperfect information can be represented as a tree induced by iterated product update.[90]

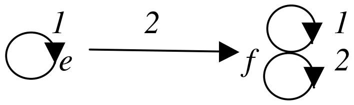
4.7 Language extensions: common knowledge

Is the dynamic-epistemic language strong enough for our models? Several new modalities make sense, as in Chapters 2 and 3. In particular, in communication scenarios, we need to deal with common knowledge – perhaps also implicit knowledge. Already the extension with common knowledge suggests further strengthening to a propositional dynamic logic*:* van Benthem, van Eijck & Kooi 2006 supplies details for what follows.

Finding a *DEL*-style recursion axiom for common knowledge is not at all routine. Indeed, it is not clear intuitively what the right common knowledge axiom should be, even for simple scenarios. We will work with arbitrary models, not necessarily with equivalence relations. Here is about the simplest illustration of what we want to analyze.

Example Secret observation, or private subgroup announcement.

Consider the following two-event model involving two agents *1, 2* – where *1* learns that the precondition of *e* holds, say *p*, while *1* thinks that nothing has happened (*T*):



---

[90] Van Benthem 2001 stated a *Finite Evolution Conjecture*: starting with finite ***M*** and ***E***, the model *Tree (**M, E**)* remains finite modulo bisimulation. Hence some horizontal levels ***M x E***^{k} and ***M x E***^{l} in the tree must be bisimilar, with $k<l$. This was refuted in Sadzik 2005. Finite evolution up to bisimulation holds only in single-agent *S5*-models, but can fail with two *S5*-agents. (Sadzik uses *pebble games* over ***M*** and ***E**,* where *Tree (**M, E**)* is finite modulo bisimulation iff the response player has a winning strategy.) The Conjecture holds for special cases, like when the epistemic relations for all agents in the ***E*** are linearly ordered by inclusion. This applies to some games.

First, we want *2* to learn nothing significant here. This can be seen as follows. The second event with its accessibilities acts as a public announcement of its precondition T = 'true'. Now here is a law of public announcement (announcing T does not change a model):

Fact $<!T>\varphi \leftrightarrow \varphi$ is schematically valid in *PAL*.

Of course, *1* does learn that p is the case (note that only e is accessible to e for *1*):

$$[\boldsymbol{E}, e]K_1p \;\leftrightarrow\; (p \to K_1(\boldsymbol{E}, e]p) \;\leftrightarrow\; (p \to K_1(p \to p)) \;\leftrightarrow\; (p \to T) \;\leftrightarrow\; T$$

Now, try to write a recursion axiom for common knowledge formulas $[\boldsymbol{E}, e]C_{\{1, 2\}}\varphi$ in this setting: it is not obvious! [91] Here is a solution, written for convenience with existential modalities, that may be computed using the techniques that prove our next theorem:

$$<\boldsymbol{E}, f><C_G>\varphi \leftrightarrow <(1 \cup 2)^*>\varphi$$
$$<\boldsymbol{E}, e><C_G>\varphi \leftrightarrow (p \wedge <(?p; 1)^*(p \wedge \varphi)) \vee (p \wedge <(?p; 1)^*(p \wedge <2><(1 \cup 2)^*>\varphi)) \quad \blacksquare$$

One solution here is to just add an ordinary common knowledge modality, give up on a recursion axiom, and prove completeness via special-purpose inference rules (as in Baltag, Moss & Solecki 1998). But if we want to stick with recursion axioms, the driving force in our principled information dynamics, we need another approach.

The only known solution right now is a drastic extension of our static language to an epistemic variant of *propositional dynamic logic*, a system we have encountered before as providing program structure on top of basic announcement actions:

Definition *EDL* language and semantics.
The *epistemic dynamic language EDL* is defined by the following inductive syntax rules:

formulas $p \mid \neg\varphi \mid \varphi\wedge\psi \mid [\pi]\varphi$
programs $i \mid \pi \cup \pi \mid \pi \,;\, \pi \mid \pi^* \mid ?\varphi$

The semantics is over the standard epistemic models of Chapter 2, in a mutual recursion. [92]

---

[91] Draw a picture of how $C_G\varphi$ is true in a product model $\boldsymbol{M \times E}$, with finite accessibility sequences $(s_1, e_1), \ldots, (s_k, e_k)$, and now try to write the same information at the level of $\boldsymbol{M}$ itself, with only **s**-sequences. You need to keep track of different events attached, and it is unclear how to do this.

[92] Cf. Blackburn, de Rijke, Venema 2000, Harel, Kozen, Tiuryn 2000, van Benthem to appearB.

Formulas are interpreted as usual, where $\boldsymbol{M}, s \models [\pi]\varphi$ means that φ is true at all successors of s in $\boldsymbol{M}$ according to the relation denoted by the program π. Simultaneously, program expressions π denote binary accessibility relations between worlds defined inductively along with the given operations: atomic i are agents' epistemic uncertainty relations, $\cup$ stands for *union* ('choice'), $;$ for *sequential composition*, and * for *reflexive-transitive closure* (Kleene star). Finally, $?\varphi$ is the *test program* $\{(s, s) \mid \boldsymbol{M}, s \models \varphi\}$. ■

Caveat: EDL versus DEL This language is a dynamic logic in the sense of logics of computation, but it should not be confused with the dynamic-epistemic logic *DEL* of this chapter that changes models. The preceding semantics stays inside one fixed model.[93]

Complex program expressions now define epistemic accessibilities for complex agents. An example is common knowledge: $C_G\varphi$ can be defined as $[(\bigcup_{i\in G} i)^*]\varphi$, with a matching agent $(\bigcup_{i\in G} i)^*$. Conditional common knowledge $C_G^{\psi}\varphi$ as in Chapter 3 involves a complex agent defined using tests: $(?\psi\ ;\ \bigcup_{i\in G} i\ ;\ ?\psi)^*$. Of course, not every program corresponds to a natural epistemic agent: we leave the study of natural fragments open here. Still, *EDL* makes an interesting general point remains about group agency (cf. Chapter 12): one can use program logics to give an explicit account of epistemic group agents.

Back to our original topic, the language *EDL* does provide recursion axioms for common knowledge, and much more. The following result states this abstractly. Consider a version of *DEL* defined as earlier in this Chapter, but where the static language is now *EDL*. The crucial recursion axiom will now be of the form $\langle \boldsymbol{E}, e\rangle\langle\pi\rangle\varphi$, for any program expression π of *EDL*, where we use existential versions for convenience in what follows.

Definition Product closure.

We call a language *L product closed* if for every formula φ in *L*, the expression $\langle \boldsymbol{E}, e\rangle\varphi$, interpreted using product update as before, is equivalent to a formula already inside *L*. We speak of *effective* product closure if there is an algorithm producing the equivalents. ■

Our earlier completeness result showed that standard epistemic logic *EL* is product-closed for *finite* event models – and indeed the recursion axioms provide an effective algorithm.

---

[93] Even so, a standard propositional dynamic perspective makes sense for *DEL*. In Chapter 3, we briefly viewed the universe of all epistemic models as a big state space (the 'Supermodel'), with transition relations matching model updates. See Chapter 11 for more on this line.

We will restrict ourselves to finite event models henceforth. Now product closure fails for the epistemic language *EL-C* that just adds plain common knowledge (Baltag, Moss, Solecki 1998; van Benthem, van Eijck & Kooi 2006). But here is a case of harmony:

Theorem The logic *E-PDL* is effectively product-closed.[94]

Proof The argument involves two simultaneous inductions. Fix some finite pointed event model $(\mathbf{E}, e)$, where in what follows, we will assume that the domain of $\mathbf{E}$ is enumerated as a finite set of numbers $\{1, \ldots, n\}$. Now, we show by induction on *EDL*-formulas that

Lemma A For φ in *EDL*, $<\mathbf{E}, e>\varphi$ is in *EDL*.

The inductive steps for atomic formulas and Booleans are as for completeness of *DEL*. The essential step is the case $<\mathbf{E}, e><\pi>\varphi$ with program π, that we analyze as follows:

Definition Path expressions.

Given any *EDL* program π, and any two events e, f in the given event model $\mathbf{E}$, we define the *EDL* program $T(e,f,\pi)$ by the following induction:

$$\begin{array}{lcll}
T(e,f,i) & = & ?Pre_e \,;\, i \,;\, ?Pre_f & \text{if } e \sim f \text{ in } \mathbf{E} \\
 & & ?\bot & \text{otherwise [95]} \\
T(e,f,\pi_1 \cup \pi_2) & = & T(e,f,\pi_1) \cup T(e,f,\pi_2) & \\
T(e,f,\pi_1 \,;\, \pi_2) & = & \bigcup_{g \in \mathbf{E}} (T(e,g,\pi_1) \,;\, T(g,f,\pi_2)) & \\
T(e,f,?\varphi) & = & ?<\mathbf{E}, e>\varphi & \text{if } e = f \\
 & & ?\bot & \text{otherwise} \\
T(e,f,\pi^*) & = & P(e,f,n,\pi) & n \text{ is largest in } \mathbf{E}
\end{array}$$

Here the auxiliary program $P(e,f,i,\pi)$ is defined by induction on the natural number i:

$$\begin{array}{lcll}
P(e,f,o,\pi) & = & T(e,f,\pi) \cup \; ?T & \text{if } e = f \\
 & & T(e,f,\pi) & \text{if } e \neq f \\
P(e,f,i+1,\pi) & = & P(e,f,i,\pi) \cup & \\
 & & P(e,i,i,\pi); P(i,i,i,\pi)^*; P(i,f,i,\pi) & \text{if } i \neq e, i \neq f \\
 & & P(e,i,i,\pi); P(i,i,i,\pi)^* & \text{if } i \neq e, i = f \\
 & & P(i,i,i,\pi)^*; P(i,f,i,\pi) & \text{if } i = e
\end{array}$$

■

---

[94] The first proof of this result was in van Benthem & Kooi 2004 using finite automata.

[95] Here, the 'falsum' $\bot$ is a formula that is always false.

All these programs are in the language *EDL*. Their meanings are clarified in the following assertion that is proven simultaneously with all other assertions in the main proof:

Claim (a) $(w, v) \in [[T(e, f, \pi)]]^{M}$ iff $((w, e), (v, f)) \in [[\pi]]^{M\,x\,E}$
(b) $(w, v) \in [[P(e, f, i, \pi)]]^{M}$ iff there is a finite sequence of transitions
in $\boldsymbol{M\ x\ E}$ of the form $(w, e)\ [[\pi]]\ x_1 \ldots x_k\ [[\pi]]\ (v, e)$ such that none of
the stages $x_j = (w_j, e_j)$ has an event e_j with index $j \geq i$.

We will not do the inductive steps in the proof of the Claim. The most complex case in the clauses for $P(e, f, i+1, \pi)$ expresses that a finite-path from e to f can at worst pass through the event with index i some finite number of times – and marking these, it can then be decomposed as an initial part from e with i of lower index i for the intermediate stages, an iteration of cases from i to i with lower index, and a final part from i to f with lower index:

$$e \quad \ldots <i \ldots \quad \boldsymbol{i} \quad \ldots.. \quad \boldsymbol{i} \quad \ldots.. \quad \boldsymbol{i} \quad \ldots <i \ldots \quad f$$

With this explanation, here is our desired equivalence:

Lemma $<\boldsymbol{E}, e><\pi>\varphi \leftrightarrow \bigvee_{i \in E} <T(e, i, \pi)><\boldsymbol{E}, i>\varphi$

The proof follows easily from the explanation for the relations $T(e, i, \pi)$. ■

The preceding proof is constructive, and it yields an effective algorithm for writing correct recursion axioms for common knowledge and in fact, all other *EDL*-definable notions. [96] Van Benthem, van Eijck & Kooi 2006 use this to provide axioms for common knowledge in subgroups after complex scenarios such as uses of *bcc* in email. [97]

---

[96] *Open Problem*: Does *EDL* have natural smaller fragments satisfying product closure?

[97] Van Benthem & Ikegami 2008 prove product closure by analyzing the recursion in an *epistemic μ–calculus* (cf. Chapter 15) extending *EDL* with fixed-point operators over positive formulas. The key is the equivalence $<\boldsymbol{E}, e><\pi^*>\psi \leftrightarrow <\boldsymbol{E}, e>\psi \vee <\boldsymbol{E}, e><\pi><\pi^*>\psi$. One views formulas $<\boldsymbol{E}, e><\pi^*>\psi$ as propositional variables p_e for all events e, that can be solved in one simultaneous smallest fixed-point equation, with clauses matching the above transition predicates $T(i, j, \pi)$ for the program π. To show that the solution is in *EDL*, they use the fact that propositional dynamic logic is a μ–calculus fragment whose recursions involve only modal diamonds and disjunctions. The same analysis shows that the epistemic μ–calculus is effectively product-closed.

4.8 Further issues and open problems

We conclude with a list of further developments in the *DEL* framework, actual or potential.

Agents and channels *PAL* and *DEL* have no explicit agency, as they just analyze events. How can agents be added? And how to model their communication channels, that restrict the sort of knowledge achievable in a group? Van Benthem 2006C discusses agents with one-to-one telephone lines. Roelofsen 2006 translates communication channels in terms of event models. A general *DEL* theory of channels (cf. Apt, Witzel & Zvesper 2009, van Eijck, Sietsma & Wang 2009 for recent progress) meets with the analysis of security and cryptography in computer science (Dechesne & Wang 2007, Kramer 2007).

Computational complexity The complexity of *DEL* core tasks is as for *PAL*. Model checking is in ***P***, model comparison is bisimulation checking in ***P***, and satisfiability is probably *Pspace*-complete, though this has not yet been proved. But other questions seem of a different nature, having to do with difficulty for agents, not complexity of our logic. There are thresholds in model size in going from public to private communication, or from telling the truth to lying and cheating. Likewise, there is complexity in update evolution, when predicting maximal size of intermediate models between the opening of a game and its end game. What is the right notion of agent-oriented complexity here? Chapter 11 discusses this same issue in somewhat more detail – but we offer no solution.

Abstract update postulates The update rule of *DEL* seems just one option. Chapter 3 had a *postulational* correspondence analysis of *PAL*, showing how world elimination is the only operation satisfying intuitive postulates on update. Can one extend this, showing how, in a space of model constructions, the *DEL* axiom $[\mathbf{E}, e]K_i\varphi \leftrightarrow (PRE_e \rightarrow \wedge \{ K_i[\mathbf{E}, f]\ \varphi))\ \mid f \sim_i e \text{ in } \mathbf{A}\})$ fixes product update? But there are also other ways of capturing update rules. Given a space of models and an ordering or distance function, one can look for models closest to the current one satisfying some condition. Thus, a hard information update *!P* on a current model ***M*** might be a model *closest to* ***M*** that has undergone some minimal change to make *P* common knowledge. [98] Sorting out connections between these approaches seems another desideratum to better understand the mechanics of *DEL* information update.

---

[98] More abstractly, updates have been studied as minimal jumps along inclusion in models that contain all relevant information stages: cf. van Benthem 1989, Jaspars 1994.

Backward or forward-looking? In line with the preceding point, *DEL* is *past-oriented*: its preconditions tell us what was the case when the informative event occurred. In contrast with this, many temporal logics of agency (cf. Chapter 11) are *future-oriented*, talking about what will be the case with postconditions of 'coming to know' or 'seeing to it that φ' (Belnap, Perloff & Xu 2001). What is the connection between these approaches?

Generic event languages *DEL* has a striking asymmetry. Epistemic models ***M*** interpret a language, but event models ***E*** do not. In particular, preconditions Pre_e are not *true* at event *e* in ***E***: they state what must be true *in **M*** for *e* to happen. But one might define a second epistemic language for event models, describing generic properties of events, with atoms, Booleans, and modalities over indistinguishable events. Van Benthem 1999A has an 'epistemic arrow logic' for this purpose, that describes product update in a joint modal language for epistemic and event models. This fits the remark in Gerbrandy 1999A that event models are often too specific. We want to make *generic assertions* like "for every epistemic event of type *X*, result *Y* will obtain". Sadrzadeh & Cristea 2006 show how algebraic formalisms can abstract away from specifics of *DEL*. What would be an optimal framework? See Aucher 2009 for a recent proposal that stays closer to logic.

Links with other disciplines: events, information, and computation Events are important in philosophy, linguistics, and artificial intelligence (McCarthy 1963, Sergot 2008). It would be of interest to connect our logics with these other traditions where events are first-class citizens. [99] *DEL* also meets computer science in the study of specific computational and informational events. We saw how it relates to fixed-point logics of computation, and to logics of agents and security – and these initial links should be strengthened.

Logic of model constructions From a technical point of view, perhaps the most striking feature of *DEL* is its explicit logic of *model constructions,* such as *PAL*-style definable submodels or *DEL*-style products. Van Benthem 1999B links the latter to a basic notion in logic: *relative interpretations* between theories using pair formation, predicate substitution, and domain relativization. Formalizing properties of such constructions seems a natural continuation of standard model theory. In this line, van Benthem & Ikegami 2008 use product closure as a criterion for expressiveness of logical languages, extending the usual

---

[99] Cf. the 2010 issue of the *Journal of Logic, Language and Information* on relating different paradigms in the temporal logic of agency, edited by van Benthem & Pacuit.

closure under translation or relativization. But there are also other technical perspectives on the product update mechanism of *DEL*. For instance, the discussion in Chapter 3 of iterated announcements $[!P][!Q]\varphi$ suggests a more general axiom

$$[\mathbf{E}_1, e_1][\mathbf{E}_2, e_2]\varphi \leftrightarrow [\mathbf{E}_1 \circ \mathbf{E}_2, (e_1, e_2)]\varphi,$$

where o is some easily definable operation of *composition* for event models. Another such operation would be *sum* in the form of disjoint union. What is a natural *complete algebra of event models*? Here, we meet computer science once more. Like *DEL*, Process Algebra (cf. Bergstra, Ponse & Smolka, eds., 2001) is all about definable bisimulation-invariant model constructions. [100] Can we connect the two approaches in a deeper manner?

From discrete dynamic logics to continuous dynamical systems We conclude with what we see as a major challenge. Van Benthem 2001, 2006C pointed out how update evolution suggests a long-term perspective that is like the evolutionary dynamics found in *dynamical systems*. Sarenac 2009 makes a strong plea for carrying Logical Dynamics into the latter realm of emergent system behaviour, pointing at the continuous nature of real events in time – and also in space. Technically, this would turn the *DEL* recursion axioms into differential equations, linking the discrete mathematics of *DEL* and epistemic temporal logic with classical continuous mathematics. Interfacing current dynamic and temporal logics with the continuous realm is a major issue, also for logic in general.

4.9 Literature

A significant first step extending *PAL* to private announcements in subgroups was made in Gerbrandy & Groeneveld 1997. Gerbrandy 1999A is a trail-blazer covering many more examples, raising many conceptual issues concerning epistemic dynamics for the first time, and technically defining a broad class of 'epistemic updates' over non-wellfounded sets. Van Ditmarsch 2000 studies the logic of a family of partly private learning events that suffice for modeling the moves in some significant parlour games. Extending Gerbrandy's work decisively, it was Baltag, Moss & Solecki 1998 who added the key idea of product update via event models, giving *DEL* its characteristic flavour and modeling power, and clarifying the mathematical theory. Baltag & Moss 2004 is a mature later version of what

---

[100] In fact, speaking simultaneously and public announcements $!(P \wedge Q)$ are already a rudimentary form of concurrency, obeying laws very different from those for sequential composition. Also relevant is van Eijck, Ruan & Sadzik 2006 on bisimulation-invariant operations on event models.

is now known as the '*BMS* approach'. Van Benthem 2001 is a systematic description of update evolution in *DEL* linked with temporal logic and games of imperfect information. Van Benthem, van Eijck & Kooi 2006 is a version of *DEL* with factual change and recursion axioms for common knowledge. These are just a few basic sources. Further contributions will be referenced in the chapters where they make their appearance.