Model Transformers for Dynamical Systems of Dynamic Epistemic Logic

Link to published version
DOI:10.1007/978-3-662-48561-3_26

Copy-Pastable BibTeX:

```
@inproceedings{Rendsvig-DELds-2015,
   author = {Rendsvig, Rasmus K.},
    title = {{Model Transformers for
            Dynamical Systems of Dynamic
            Epistemic Logic}},
   editor = {Hoek, Wiebe van der and
            Holliday, Wesley H. and Wang,
            Wen-fang},
booktitle = {Logic, Rationality, and
            Interaction (LORI 2015, Taipei)},
    pages = {316-327},
   series = {LNCS},
     year = {2015},
publisher = {Springer}
}
```

Rasmus K. Rendsvig
rendsvig@gmail.com

LU Information Quality Research Group, Lund University
Center for Information and Bubble Studies, University of Copenhagen

Abstract. This paper takes a dynamical systems perspective on the semantic structures of dynamic epistemic logic (DEL) and asks the question which orbits DEL-based dynamical systems may produce. The class of dynamical systems based directly on action models produce very limited orbits. Three types of more complex model transformers are equivalent and may produce a large class of orbits, suitable for most modeling purposes.

Keywords: dynamic epistemic logic, dynamical systems, model transformers, protocols, modeling

1 Introduction

When modeling socio-epistemic phenomena, working with the temporally local models of dynamic epistemic logic (DEL) is both a blessing and a bane. It is a blessing as both epistemic state models and their updates are small relative to a fully explicated epistemic temporal structure. This eases both model construction and comprehension. It is a bane as the small models are incomplete: each is an individual time-step while we seek to model temporally extended dynamics. To form a 'complete model', we must specify the 'temporal glue' that ties individual epistemic states together to dynamics.

This 'temporal glue' is often presented informally in the DEL literature by way of a natural language problem description, typically involving conditional tests to determine which update to apply. Methodologically, this leaves modelers with a small gap: when modeling information dynamics using the semantic tools of DEL, what *mathematical object* shall we identify as *the model* of our target phenomenon?

It is an advantage of the DEL approach that a full sequential model need not be specified from the outset, but a drawback that a complete formalization of the problem under investigation is missing. Ideally, such 'complete models' should be both

1. Computably tractable (for each step), and
2. Informative (model the problem, not just describe the solution).

The first *desideratum* is for implementation purposes. By the second, it is sought that eventual implementations are interesting: models that formalize problems without requiring they be solved first, allows one to draw informative conclusions about the modeled phenomena. The informal approach is typically informative.

This paper suggests a *dynamical systems* approach to specifying 'complete models' of information dynamics and provides some preliminary results.[1] As a (discrete time) dynamical system consists of only a state space X and a map $\tau : X \longrightarrow X$ iteratively applied, the future development of the dynamics depend only on the current state and the map τ. Dynamical systems thus provide a formal container for dynamical models in the local spirit of DEL. This stands in contrast to the only formal alternative, *DEL protocols* [3], which define dynamics globally. This approach is discussed in Section 3.

Dynamical systems are simple but may therefore also be limiting. E.g., if one's chosen model transformer class contains only action models, then the set of scenarios that can be modeled is very narrow: the same action model will be reapplied by the dynamical system, scenarios such as the well-known Muddy Children example [10] are among the unrepresentable phenomena. This provides a motivation for seeking broader classes of model transformers, the topic of Section 5. Three methods for defining complex model transformers are defined, being *multi-pointed action models*, *programs* and *problems*. The main technical results compare these approaches with respect to the orbits they can produce when used in dynamical systems.

2 DEL Preliminaries

Let be given a finite, non-empty set of propositional atoms Φ and a finite, non-empty set of agents, $\mathcal{A}$.

Definition 1 (Kripke Model). *A* Kripke model *is a tuple* $M = (\llbracket M \rrbracket, R, \llbracket\cdot\rrbracket)$ *where*

$\llbracket M \rrbracket$ *is a non-empty set of states;*

$R : \mathcal{A} \longrightarrow \mathcal{P}(S \times S)$ *is an accessibility function;*

$\llbracket\cdot\rrbracket : \Phi \longrightarrow \mathcal{P}(S)$ *is a valuation function.*

A pair (M, s) *with* $s \in \llbracket M \rrbracket$ *is called an* epistemic state.

Definition 2 (Language, Semantics). *Where* $p \in \Phi$ *and* $i \in \mathcal{A}$, *define a language* $\mathcal{L}$ *by*

$$\varphi := \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi$$

with non-propositional formulas evaluated over epistemic state (M, s) *by*

$$(M, s) \models K_i\varphi \textit{ iff } \forall t \in R_i(s), (M, t) \models \varphi.$$

[1] The approach to dynamical systems taken here thus differs from that [14], which mainly seeks modal logical descriptions of dynamical system concepts.

With a normal modal logical language like $\mathcal{L}$, the natural notion of equality of epistemic states is *bisimulation*:

Theorem 1 (Hennessy-Milner, [4], Thm.2.24). *Let M and M' be image-finite, i.e., $\forall s \in \llbracket M \rrbracket, \forall i \in \mathcal{A}$, the set $\{t : (s,t) \in R_i\}$ is finite. Then for all $s \in \llbracket M \rrbracket, s' \in \llbracket M' \rrbracket$, s and s' are modally equivalent iff (M,s) and (M',s') are bisimilar.*

When working with finite models, $\mathcal{L}$ is strong enough to distinguish any two non-bisimilar models:

Theorem 2 ([11], Thm.32). *Let (M,s) and (M',s') be finite epistemic states that are not n-bisimilar. Then there exists $\delta \in \mathcal{L}$ such that $(M,s) \models \delta$ and $(M',s') \not\models \delta$.*

Dynamics are introduced by transitioning from one epistemic state to the next:

Definition 3 (Model Transformer). *Let $\mathcal{M}$ be the set of epistemic states based on $\mathcal{A}$. A* model transformer *is a (possibly partial) function $\tau : \mathcal{M} \longrightarrow \mathcal{M}$.*

Several model transformers have been suggested in the literature, the most well-known being public announcement, $!\varphi$ [12]. Primary to this paper is the rich class of action models [2] with postconditions [8].

Definition 4 (Action Model). *An* action model *is a tuple $\Sigma = (\llbracket \Sigma \rrbracket, \mathsf{R}, \mathsf{pre}, \mathsf{post})$ where*

$\llbracket \Sigma \rrbracket$ *is a finite, non-empty set of* actions*;*

$\mathsf{R} : \mathcal{A} \longrightarrow \mathcal{P}(\llbracket \Sigma \rrbracket \times \llbracket \Sigma \rrbracket)$ *is an* accessibility *function;*

$\mathsf{pre} : \llbracket \Sigma \rrbracket \longrightarrow \mathcal{L}$ *is a* precondition *function;*

$\mathsf{post} : \llbracket \Sigma \rrbracket \longrightarrow \{\bigwedge_{i=0}^{n} \varphi_i \not\models \bot : \varphi_i \in \{\top, p, \neg p : p \in \Phi\}\}$ *is a* postcondition *function.*

A pair (Σ, σ) with $\sigma \in \llbracket \Sigma \rrbracket$ is called an epistemic action*.*

The precondition of an action σ specifies the conditions under which σ is executable; the postconditions specify how σ sets the values of select atoms. If $\mathsf{post}(\sigma) = \top$, then σ changes nothing.

An epistemic state is informationally updated with an epistemic action by taking their product:

Definition 5 (Product Update). *The* product update *of epistemic state $(M,s) = (\llbracket M \rrbracket, R, \llbracket \cdot \rrbracket, s)$ with epistemic action $(\Sigma, \sigma) = (\llbracket \Sigma \rrbracket, \mathsf{R}, \mathsf{pre}, \mathsf{post}, \sigma)$ is the epistemic state*

$$(M \otimes \Sigma, (s,\sigma)) = (\llbracket M \otimes \Sigma \rrbracket, R', \llbracket \cdot \rrbracket', (s,\sigma))$$

where

$\llbracket M \otimes \Sigma \rrbracket = \{(s,\sigma) \in \llbracket M \rrbracket \times \llbracket \Sigma \rrbracket : (M,s) \models \mathsf{pre}(\sigma)\}$

$R'_i = \{((s,\sigma),(t,\tau)) : (s,t) \in R_i \text{ and } (\sigma,\tau) \in \mathsf{R}_i\}$

$\llbracket p \rrbracket' = \{(s,\sigma) : s \in \llbracket p \rrbracket, \mathsf{post}(\sigma) \not\models \neg p\} \cup \{(s,\sigma) : \mathsf{post}(\sigma) \models p\}$.

In combination, an epistemic action (Σ, σ) and product update $\otimes$ thus define a model transformer. Denote the class of such transformers by $\mathbf{\Sigma}$. Each $\tau \in \mathbf{\Sigma}$ has the following pleasant property:

Fact (Bisimulation Preservation). $\forall \tau \in \mathbf{\Sigma}$, if (M, s) and (M', s') are bisimilar, then so are $\tau(M, s)$ and $\tau(M', s')$.

$\mathbf{\Sigma}$ is a very powerful class: for any finite epistemic state (M, s), it contains a transformer that will map (M, s) to any other finite epistemic state (M', s'), as long as no agents with empty access in M has non-empty access in M' and as long as M and M' differ only in the truth value of a finite number of atoms. The restrictions are due to the 'and'-condition used in defining R'_i in product update and the finite conjunction used in defining postcondition maps. If the directed relation given by these restrictions holds from (M, s) to (M', s'), then call the transition from the first to the second *reasonable*:

Definition 6 (Reasonable Transition). *Let* (M, s)= $(\llbracket M \rrbracket, R, V, s)$ *and* $(M', s') = (\llbracket M' \rrbracket, R', V', s')$ *be two epistemic states. Then the transition from* (M, s) *to* (M', s') *is* reasonable *iff*

1. *it* preserves insanity*: there exists a submodel* M^{s} *of* M *such that* $s \in \llbracket M^{\mathsf{s}} \rrbracket$ *and* $\forall i \in \mathcal{A}$*, if* $R'_i \neq \emptyset$*, then* R_i *is serial in* M^{s}*, and*
2. *it invokes* finite ontic change*:*

$$\begin{aligned} &\{p : \llbracket p \rrbracket \neq \emptyset \text{ and } \llbracket p \rrbracket \neq \llbracket M \rrbracket\} \\ \cup\, &\{p : \llbracket p \rrbracket = \emptyset\} \setminus \{p : \llbracket p \rrbracket' = \emptyset\} \\ \cup\, &\{p : \llbracket p \rrbracket = \llbracket M \rrbracket\} \setminus \{p : \llbracket p \rrbracket' = \llbracket M' \rrbracket\} \end{aligned}$$

is finite.

Theorem 3 (Arbitrary Change, [8], Prop.3.2). *Let the transition from finite* (M, s) *to finite* (M', s') *be reasonable. Then there exists a* $(\Sigma, \sigma) \in \mathbf{\Sigma}$ *such that* $(M, s) \otimes (\Sigma, \sigma)$ *and* (M', s') *are bisimilar.*

3 DEL Protocols

One framework which could be used to construct 'complete models' is *DEL protocols* [3,7,13,15].

Definition 7 (DEL Protocol). *Let* $\mathbf{\Sigma}^*$ *be the set of all finite sequences of transformers* $\tau \in \mathbf{\Sigma}$*. A set* $\mathsf{P} \subseteq \mathbf{\Sigma}^*$ *is a (uniform)* DEL protocol *iff* P *is closed under non-empty prefixes.*

A DEL protocol specifies which model transformers *may* be executed at a given time—whether they *can* be executed depends on the model transformers, e.g. their preconditions.

Where P is a DEL protocol and $\sigma = (\tau_1, ..., \tau_n) \in \mathsf{P}$, set $(M, s)^\sigma := \tau_n \circ \cdots \circ \tau_1(M)$. From an initial model (M, s) and time 0, a DEL protocol P produces a set of possible evolutions to each time n, namely

$\{(M, s)^\sigma : \mathsf{len}(\sigma) = n\}$. Notice that $\mathsf{len}(\sigma) = n$ does not imply that $(M, s)^\sigma$ exists: one of the transformers from σ may have been unexecutable at some earlier stage.

DEL protocols are dismissed as suitable for constructing 'complete models' as the results will be unexecutable, incorrect or uninformative. To see this, assume that some phenomenon that involves multiple model transformers $T = \{\tau_1, ..., \tau_n\}$, as e.g. Muddy Children does.

If the DEL protocol used is T^* (the set of all finite strings sequences of transformers from T) a very nice model is obtained: it is applicable to multiple initial states with varying mud distributions, and it may accordingly be used to obtain answers to questions about e.g. how the scenario unfolds as a function of the number of muddy children. Alas, T^* is infinite and as a model therefore unexecutable: given some initial state (M, s) it will not be possible to run T^* on (M, s) in finite time as the input to any function that is to determine the set $\{(M, s)^\sigma : \mathsf{len}(\sigma) = 1\}$ will be infinite.

To obtain an executable model, T^* could be pruned to obtain a finite DEL protocol $\mathsf{T} \subseteq T^*$, e.g. by setting some upper bound on the length of $\sigma \in \mathsf{T}$. The risk associated with this move (pruning) is that the model becomes useless or uninformative: if the upper bound is set too low, the model will terminate too soon and not provide a correct output; to ensure the upper bound high enough, the problem must have been solved beforehand, leading to an uninformative model. In the extreme case where the only included maximal σ is 'the correct one' given some natural language protocol and initial state, a descriptive model is produced, but such a 'gold in, gold out' model is of little interest from an investigative perspective.

4 DEL and Dynamical Systems

Given Theorem 3, one might expect that dynamical systems based on the class of action models $\mathbf{\Sigma}$ would allow modeling of a plethora of phenomena. Surprisingly, not even even simple and well-known epistemic puzzles such as Muddy Children can be modeled by this class. To see this, let us first clarify the notion of dynamical system.

As standardly defined [6], a dynamical system is a tuple $D = (X, T, \mathcal{E})$ where X is set, called the *state space*, $T \subseteq \mathbb{R}$ is a *time set* which forms an additive semi-group ($t_1, t_2 \in T \Rightarrow t_1 + t_2 \in T$) and $\mathcal{E} : X \times T \to X$ is an *evolution map* satisfying that $\mathcal{E}(x, 0) = 0$ and $\mathcal{E}(\mathcal{E}(x, t_1), t_2) = \mathcal{E}(x, t_1 + t_2)$.[2]

To obtain a state space for DEL-based dynamical systems, it is natural, given Theorem 1, to equate bisimilar epistemic states, and let the state space consist of each bisimulation type's smallest representative. For an epistemic state (M, s), this representative is given by (M, s)'s *generated submodel rooted at* s's *bisimulation quotient* $(M[s]/\rho^M, [s]^M_\rho)$, see [11], Sec. 3.6. Setting

$$\mathbf{M} := \{(M[s]/\rho^M, [s]^M_\rho) : (M, s) \text{ is an epistemic state}\},$$

[2] Erratum to published version: "$\mathcal{E}(x, 0) = 0$" should of course be "$\mathcal{E}(x, 0) = x$".

a class is obtained that contains a canonical representative of each epistemic state, each unique up to isomorphism.

As DEL updates are discrete and non-invertible, the suitable time set for a DEL-based dynamical system is $\mathbb{Z}_+$. The evolution function of any dynamical system $D = (X, \mathbb{Z}_+, \mathcal{E})$ with time set $\mathbb{Z}_+$ may be defined by the iterations of a function $e : X \to X$ by $\mathcal{E}(x, n) = e^n(x)$. Given the chosen state space, the suitable class of such functions e is the set of model transformers $\tau : \mathbf{M} \to \mathbf{M}$, denoted by $\mathbf{T}$.

Given these considerations, the following definition of DEL-based dynamical systems is obtained:

Definition 8 (DEL-based Dynamical System). *A* DEL-based dynamical system *is a pair* $D = (\mathbf{X}, \tau)$ *where* $\mathbf{X} \subseteq \mathbf{M}$ *and* $\tau : \mathbf{X} \to \mathbf{X}$.
The orbit *of* D *from* initial state $x_0 \in \mathbf{X}$ *is the sequence* $o(D, x_o) = (\tau^n(x_0))_{n \in \mathbb{Z}_+}$.

Remark. Given an epistemic action $\tau \in \mathbf{\Sigma}$, $x \in \mathbf{M}$ does not imply that $\tau(x) \in \mathbf{M}$. There will however be a $x' \in \mathbf{M}$ that is bisimilar to $\tau(x)$. Given Fact 1, each $\tau \in \mathbf{\Sigma}$ may be identified with a $\tau' \in \mathbf{T}$ by if $\tau(x) = (M, s)$, then $\tau'(x) = (M[s]/\rho^M, [s]^M_\rho)$. Henceforth, when executing an epistemic action (Σ, σ) in $x \in \mathbf{M}$, it is thus assumed that $x \otimes (\Sigma, \sigma) \in \mathbf{M}$.

It is immediately clear that any dynamical system $D = (\mathbf{X}, \tau)$ with $\tau \in \mathbf{\Sigma}$ will be limited in its orbits. In particular, where s_0 is the actual state in the initial epistemic state x_0 and σ_0 is the actual state of τ, then for any n, the actual state of $\tau^n(x_0)$ will be of the form $(...(s_0, \sigma_0), ..., \sigma_0)$. Consequently, any phenomenon that involves the occurrence of more than one actual action is unmodelable. As most phenomena do involve shift in the performed action, e.g. by a shift in the announcement made, there is a motivation for seeking out a more general class of model transformers.

5 Complex Model Transformers

The limitation of DEL-based dynamical systems does not stem from action models, but rather from the fact that their usage is not controlled. This problem is solved by DEL protocols or update streams; simply specify at which time which action model should be executed. However, this requires a description of the evolution before execution, leaving little of the local DEL spirit intact.

A natural way to specify which transformer should be applied next that still remains local in spirit is by using a map $\pi : \mathbf{M} \longrightarrow \mathbf{T}$. Composing such a π with the model transformers it picks at each epistemic state is then again a model transformer $\tau_\pi : \mathbf{M} \longrightarrow \mathbf{M}$ given by $\tau_\pi(x) = \pi(x)(x)$.

To be interesting from modeling and implementation perspectives, such π must be finitely representable. This puts constraints on the dynamical systems definable, but, as will be shown, the restriction is still to a vast class of such systems.

We focus on three ways of specifying maps π, each picking model transformers from $\mathbf{\Sigma}$. The choice to restrict attention to maps picking transformers from

$\mathbf{\Sigma}$ is warranted by Theorem 3: As basic transformers, this class has sufficient transformational power to construct a rich class of dynamical systems.

The first type is closely related to the (knowledge-based) programs known from interpreted systems [10], though defined to specify transformers based on the global, epistemic state rather than specifying sub-actions based on agents' local states:[3]

Definition 9 (Program). *A (finite, deterministic,$(\mathcal{L}, \mathbf{\Sigma})$)* program *is a finite set of formula-transformer pairs*

$$P = \{(\varphi_i, \tau_i) : \varphi_i \in \mathcal{L}, \tau_i \in \mathbf{\Sigma}\}$$

where $\forall i, j$ *if* $\varphi_i \neq \varphi_j$ *and* $(\varphi_i, \tau_j), (\varphi_j, \tau_j) \in P$, *then* $\mathbf{M} \models \varphi_i \wedge \varphi_j \rightarrow \bot$.[4]

Each program P *gives rise to a model transformer* τ_P *given by* $\tau_P(x) = \tau_i(x)$ *if* $x \models \varphi_i$ *and* $(\varphi_i, \tau_i) \in P$. *Denote this class by* $\mathbf{P}$.

Each program may be read as a set of conditional tests of the form **if** φ_i**, do** τ_i, in form similar to the informal specifications often used in DEL literature.

The explicit specification of programs stands in contrast with the implicit specification of the second transformer type, *problems*, where each instruction may be read **if** φ_i**, obtain** ψ_i. Problems as defined here are related to *epistemic planning problems*, also know from the DEL literature [5].

Definition 10 (Problem). *A (finite $(\mathcal{L}, \mathbf{\Sigma})$)* problem *is a pair*

$$\Pi = (Q, \mathbf{\Sigma}_\Pi)$$

where $Q = \{(\varphi_i, \psi_i) : \varphi_i, \psi_i \in \mathcal{L}\}$ *is a finite set of formula-formula pairs and* $\mathbf{\Sigma}_\Pi \subset \mathbf{\Sigma}$ *is a finite set of model transformers with an associated strict order* $<$.

A solution *to* $\Pi = (Q, T)$ *at epistemic state* x *is a model transformer* $\tau \in T$ *such that* $\forall(\varphi_i, \psi_i) \in Q$, *if* $x \models \varphi_i$, *then* $\tau(x) \models \psi_i$. *Denote the set of solution to* Π *at* x *by* $\Pi(x)$.

Each problem Π *gives rise to a model transformer* τ_Π *given by* $\tau_\Pi(x) = \min_< \Pi(x)$. *Denote this class by* $\mathbf{\Pi}$.

The model transformer τ_Π is defined using the strict order $<$ on $\mathbf{\Sigma}_\Pi$ to ensure that τ_Π is a function: nothing in the definition ensures that $|\Pi(x)| \leq 1$.

The last model transformer type to be considered is a slight generalization of action models [1], where each such may have multiple actual states. In the definition it is required, non-standardly, that the preconditions of the actual states must be mutually exclusive. This is to ensure that executing a multi-pointed action model using product update remains a single-pointed epistemic state.

[3] Programs based on agents' local states is also at least to some degree feasible in a DEL setting, using *parallel action model composition* [9].

[4] ERRATUM: This line should read "if $(\varphi_i, \tau_i), (\varphi_j, \tau_j) \in P$ and $\tau_i \neq \tau_j$, then $\mathbf{M} \models \varphi_i \wedge \varphi_j \rightarrow \bot$"

Definition 11 (Multi-Pointed Epistemic Actions). *A (finite, deterministic) multi-pointed epistemic action is an epistemic action* (Σ, σ) *with* σ *replaced by a finite, non-empty set* $S \subseteq [\![\Sigma]\!]$*, where for each* $\sigma, \sigma' \in S$*, if* $\sigma \neq \sigma'$*, then* $\mathbf{M} \models pre(\sigma) \wedge pre(\sigma') \rightarrow \bot$*.*

Applied using product update, each (Σ, S) *is a model transformer* $\tau : (M \otimes \Sigma, (s, S)) \mapsto ([\![M \otimes \Sigma]\!], R', [\![\cdot]\!]', (s, \sigma_i))$ *where* $(M, s) \models pre(\sigma_i)$*. Denote this class by* $\mathbf{\Sigma}^{+}$*.*

With mutually exclusive preconditions, a multi-pointed action model (Σ, S) encodes a map $\pi : \mathbf{M} \longrightarrow \mathbf{T}$ with image $\{(\Sigma, \sigma) : \sigma \in S\}$ by $\pi(x) = (\Sigma, \sigma)$, $x \models pre(\sigma)$.

6 Results

Note initially that DEL-based dynamical systems fair better than DEL protocols in regard to executability and informativity. DEL-based dynamical systems resting on either a program or a multi-pointed action model are step-wise computable, as both transformer types are finite and therefore require only check of a finite set of formulas at each (M, s). The case for problems must be checked against [5]. Moreover, DEL-based dynamical systems will provide informative models: once a system is defined, one may start investigating how its orbits behave as a function of initial state without having pre-solved the encoded problem.

The first main result shows that dynamical systems based on the class $\mathbf{\Pi}$ of problem-based model transformers can model any reasonable, deterministic, finite or cyclic sequence of finite epistemic states. Problem-based dynamical systems can thus model a large class of phenomena.

The proof of Proposition 1 is by brute force. The construction results in a large, cumbersome problem fully pre-encoding the target orbit. For many modeling purposes, far more economical complex model transformers will do.

Definition 12 (Finite Variation, Deterministic). *Let* $\overline{x} = (x_0, x_1, ...)$ *be a sequence of epistemic states from* $\mathbf{M}$*.* $\overline{x}$ *has* finite variation *iff*

1. $\overline{x}$ *is finite, or*
2. $\exists n, m, k \in \mathbb{Z}_{+} \backslash \{0\} : x_k = x_{k+m}$ *for all* $k \geq n$*.*

$\overline{x}$ *is* deterministic *iff if* $x_k, x_{k+1}, x_m \in \overline{x}$ *and* $x_k = x_m$*, then* $x_{m+1} \in \overline{x}$ *and* $x_{k+1} = x_{m+1}$*.*

Proposition 1 (Arbitrary Orbits). *Let the sequence* $\overline{x} = (x_0, x_1, ...)$ *of finite epistemic states be deterministic, with finite variation and where the transition between each* x_i *and* x_{i+1} *is reasonable. Then there exists a dynamical system* $D = (\mathbf{M}, \tau_{\Pi})$ *with* $\tau_{\Pi} \in \mathbf{\Pi}$ *such that* $o(D, x_0) = \overline{x}$*.*

Proof. By constructing a problem $\Pi = (Q, \Sigma_{\Pi})$ that gives rise to the sought τ_{Π}.

For each $x_i, x_j \in \overline{x}$, $x_i \neq x_j$, let $\delta_{i,j}$ be a formula that distinguishes x_i from x_j such that $x_i \models \delta_{i,j}$ and $x_j \not\models \delta_{i,j}$; this $\delta_{i,j}$ exists by Theorem 2. As $\overline{x}$ has finite variation, $\delta_i := \bigwedge_{j:x_j \in \overline{x} \setminus \{x_i\}} \delta_{i,j}$ is a formula that distinguishes x_i from all other $x_j \in \overline{x}$. For each $x_i, x_{i+1} \in \overline{x}$, let $\tau_i \in \mathbf{\Sigma}$ be a model transformer such that $\tau_i(x_i) = x_{i+1}$; this exists by Theorem 3.

Let Q be the smallest set that for each $x_i, x_{i+1} \in \overline{x}$ contains (δ_i, δ_{i+1}). Let $\mathbf{\Sigma}_\Pi$ be the smallest set that for each $x_i, x_{i+1} \in \overline{x}$ contains τ_i. Both Q and $\mathbf{\Sigma}_\Pi$ are finite by the assumption of finite variation, so $\Pi = (Q, \Sigma_\Pi)$ is a finite program, so τ_Π is a model transformer.

That $o(D, x_0) = \overline{x}$ when $D = (\mathbf{M}, \tau_\Pi)$ is shown by induction on x_n: *Base:* ${\tau_\Pi}^0(x_0) = x_0$. *Step:* Assume ${\tau_\Pi}^n(x_0) = x_n$. If $\overline{x} = (x_0, ..., x_n)$, then $o(D, x_0) = \overline{x}$ as $(\delta_n, \varphi) \notin Q$ for any φ, by determinism of $\overline{x}$, so $\tau_\Pi(x_n)$ is undefined. If $x_{n+1} \in \overline{x}$, then $(\delta_n, \delta_{n+1}) \in Q$ and $\tau_n \in \mathbf{\Sigma}_\Pi$. By construction, $\Pi(x_n) = \tau_n$, so $\tau_\Pi(x) = x_{n+1}$. □

Proposition 2 (Problem Orbit Properties). *Let $o(D, x_0) = \overline{x}$ with $D = (\mathbf{M}, \tau_\Pi)$, $\tau_\Pi \in \mathbf{\Pi}$. Then $\overline{x}$ is deterministic and for each $x_i, x_{i+1} \in \overline{x}$, the transition from x_i to x_{i+1} is reasonable.*

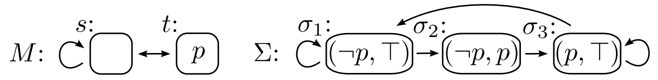
Proof. $\overline{x}$ is deterministic as τ_Π is a function; each transition is reasonable as $x_{i+1} = \tau(x_i)$ for some $\tau \in \mathbf{\Sigma}$.

Propositions 1 and 2 cannot be strengthened to a characterization result as not all problem-based dynamical system have finite variation:

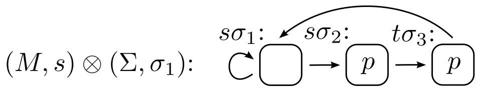
Proposition 3 (Infinite Variation). *There exists a dynamical system $D = (\mathbf{M}, \tau_\Pi)$ with $\tau_\Pi \in \mathbf{\Pi}$ such that $o(D, x_0)$ does not have finite variation.*

Proof. Let $D = (\mathbf{M}, \tau_\Pi)$ with problem $\Pi = (\{(\top, \top)\}, \{(\Sigma, \sigma_1)\})$. This trivial problem has unique solution (Σ, σ_1) for all $(M, s) \in \mathbf{M}$. Hence, for all $x \in \mathbf{M}$, $\tau_\Pi(x) = (M, s) \otimes (\Sigma, \sigma_1)$.

Let M and Σ given by



Then $o(D, (M, s))$ does not have finite variation: for each iteration of τ_Π, the state not satisfying p will split, inserting a new p state as it's child with σ_2:



All other states have only one child, with σ_3.

In all further applications of (Σ, σ_1), the circular structure seen in $(M, s) \otimes (\Sigma, \sigma_1)$ is preserved, only with an additional p state. No two such models are bisimilar, and hence the orbit does not have finite variation. □

The second main result shows that also program-based dynamical systems and dynamical systems based on multi-pointed action models can produce a vast class of orbits.

Proposition 4 (Equivalence). *Let $\overline{x} = (x_0, x_1, ...)$ be a sequence of epistemic states. Then*

1. $\exists \tau_{\Pi} \in \mathbf{\Pi}$ *such that for* $D = (\mathbf{M}, \tau_{\Pi})$, $o(D, x_0) = \overline{x}$.

$\Uparrow$

2. $\exists \tau_P \in \mathbf{P}$ *such that for* $D = (\mathbf{M}, \tau_P)$, $o(D, x_0) = \overline{x}$.

$\Updownarrow$

3. $\exists \tau_{\Sigma^+} \in \mathbf{\Sigma}^+$ *such that for* $D = (\mathbf{M}, \tau_{\Sigma^+})$, $o(D, x_0) = \overline{x}$.

If $\overline{x} = (x_0, x_1, ...)$ has finite variation and x_0 is finite, then the three statements are equivalent.

Proof.
Case: 2. $\Rightarrow$ 1. Let $D = (\mathbf{M}, \tau_P)$, $\tau_P \in \mathbf{P}$ with $o(D, x_0) = \overline{x} = (x_0, x_1, ...)$ be given.

Construct a problem $\Pi = (Q, \Sigma_{\Pi})$ as follows: Let Q be the smallest set that for each $(\varphi_i, \tau_i) \in P$ contains $(\varphi_i, \top)$. Let $\mathbf{\Sigma}_{\Pi}$ be the smallest set that for each $(\Sigma, \sigma) \in \mathbf{\Sigma}_P$ contains (Σ, σ^*) identical to (Σ, σ) in all respects except that $pre(\sigma^*) = pre(\sigma) \wedge \varphi_i$. As P is finite, $\Pi = (Q, \Sigma_{\Pi})$ is a finite problem; τ_{Π} is a model transformer as the φ_i's of P are mutually exclusive.

Then $o((\mathbf{M}, \tau_{\Pi}), x_0) = o((\mathbf{M}, \tau_P), x_0)$: Assume $x_i, x_{i+1} \in \overline{x}$. Then $x_{i+1} = \tau(x_i)$ for some $\tau = (\Sigma, \sigma)$ such that for some φ, $(\tau, \varphi) \in P$. Hence for some $\varphi, (\tau, \varphi) \in P$, it holds that $x_i \models \varphi$. Given the preconditions and that $(\varphi, \top) \in Q$, $\tau^* = (\Sigma, \sigma^*) \in \mathbf{\Sigma}_{\Pi}$ will be the only solution to Π at x_i. As $x_i \models \varphi$, $\tau^*(x_i) = \tau(x_i)$.

Assume $\overline{x} = (x_0, ..., x_n)$ is finite. Then either $x_n \not\models \varphi_i$ for all $(\varphi_i, \tau_i) \in P$ or if $x_n \models \varphi_i$ for $(\varphi_i, (\Sigma, \sigma)) \in P$, then $x_n \not\models pre(\sigma)$. In the first case, $x_n \not\models \varphi_i$ for all $(\varphi_i, \top) \in Q$; in the second, $x_n \not\models pre(\sigma^*)$. In either case, $\tau_{\Pi}(x_n)$ is undefined.

Case: 2. $\Rightarrow$ 3. Let $D = (\mathbf{M}, \tau_P)$, $\tau_P \in \mathbf{P}$ with $o(D, x_0) = \overline{x} = (x_0, x_1, ...)$ be given. Let $\mathbf{\Sigma}_{\Pi}$ be as in the case 2. $\Rightarrow$ 1. Define a multi-pointed action model (Σ^+, S) by $\Sigma^+ = \biguplus \{\Sigma : (\Sigma, \sigma^*) \in \mathbf{\Sigma}_{\Pi}\}$ and $S = \{\sigma^* : (\Sigma, \sigma^*) \in \Sigma_{\Pi}\}$. Let τ_{Σ^+} be the associated model transformer.

Then $o((\mathbf{M}, \tau_{\Sigma^+}), x_0) = o((\mathbf{M}, \tau_P), x_0)$: Assume $x_i, x_{i+1} \in \overline{x}$. Then $x_{i+1} = \tau(x_i)$ for some $\tau = (\Sigma, \sigma)$ such that for some φ, $(\tau, \varphi) \in P$. Hence for some $\varphi, (\tau, \varphi) \in P$, it holds that $x_i \models \varphi \wedge pre(\sigma)$, so by construction, $x_i \models pre(\sigma^*)$. Hence only the submodel (Σ, σ^*) of Σ^+ is executable at x_i, so $\tau_{\Sigma^+}(x_i) = \tau_P(x_i)$.

If $\overline{x} = (x_0, ..., x_n)$ is finite, then either $x_n \not\models \varphi_i$ for all $(\varphi_i, \tau_i) \in P$ or if $x_n \models \varphi_i$ for $(\varphi_i, (\Sigma, \sigma)) \in P$, then $x_n \not\models pre(\sigma)$. In the first case, $x_n \not\models pre(\sigma^*)$ for all $(\Sigma, \sigma^*) \in \Sigma^+$; in the second, $x_n \not\models pre(\sigma^*)$. In either case, $\tau_{\Sigma^+}(x_n)$ is undefined.

Case: 3. $\Rightarrow$ 2. Let $D = (\mathbf{M}, \tau_{\Sigma^+})$, $\tau_{\Sigma^+} \in \mathbf{\Sigma}^+$ with $o(D, x_0) = \overline{x} = (x_0, x_1, ...)$ be given. Let the Σ^+ of τ_{Σ^+} be $\Sigma^+ = (\Sigma, S)$ and create from it a set of $|S|$ single-pointed action models $A = \{(\Sigma, \sigma) : \sigma \in S\}$. Create a program $P = \{(pre(\sigma), (\Sigma, \sigma)) : (\Sigma, \sigma) \in A\}$. P is both finite and deterministic.

Then $o((\mathbf{M}, \tau_P), x_0) = o((\mathbf{M}, \tau_{\Sigma^+}), x_0)$: Assume $x_i, x_{i+1} \in \overline{x}$. Then $x_i \models pre(\sigma)$ for exactly one $\sigma \in S$. As $(pre(\sigma), (\Sigma, \sigma)) \in P$, $\tau_P(x_i) = \tau_{\Sigma^+}(x_i)$.

If If $\overline{x} = (x_0, ..., x_n)$ is finite, then $x_n \not\models pre(\sigma)$ for all $\sigma \in S$. Hence for all $(\varphi, \tau) \in P$, $x_n \not\models \varphi$, so $\tau_P(x_n)$ is undefined.

Case: 1. $\Rightarrow$ 2., *if* $\overline{x} = (x_0, x_1, ...)$ *has finite variation and* x_0 *is finite*: Let $D = (\mathbf{M}, \tau_\Pi)$, $\tau_\Pi \in \mathbf{\Pi} = (Q, \Sigma_\Pi)$ with $o(D, x_0) = \overline{x} = (x_0, x_1, ...)$ having finite variation. Brute force construct a program using characteristic formulas: let δ_i be the characteristic formula of $x_i \in \overline{x}$. For each pair $x_i, x_{i+1} \in \overline{x}$, there is a unique $\tau_i \in \Sigma_\Pi$ such that $\tau_i(x_i) = x_{i+1}$. Let $P = \{(\delta_i, \tau_i) : x_i \in \overline{x}\}$. As $\overline{x}$ has finite variation, P is finite and gives rise to a model transformer τ_P.

Then $o((\mathbf{M}, \tau_P), x_0) = o((\mathbf{M}, \tau_\Pi), x_0)$: Assume $x_i, x_{i+1} \in \overline{x}$. Then $(\delta_i, \tau_i) \in P$, so $\tau_P(x_i) = x_{i+1}$. If $\overline{x} = (x_0, ..., x_n)$ is finite, then by Proposition 2, for no $x_i, i < n$ is $x_i = x_n$. Hence $(\delta_n, \tau) \notin P$, for any τ. Hence $\tau_P(x_n)$ is undefined. □

Corollary 1 (Orbit Properties). *For any dynamical system* $D = (\mathbf{M}, \tau)$ *with* $\tau \in \mathbf{P} \cup \mathbf{\Sigma}^+$ *and any* $x_0 \in \mathbf{M}$, $o(D, x_0)$ *is deterministic and for each* $x_i, x_{i+1} \in \overline{x}$, *the transition from* x_i *to* x_{i+1} *is reasonable.*

Proof. Let D be as described. By Proposition 4 there exists a $D' = (\mathbf{M}, \tau_\Pi)$, $\tau_\Pi \in \mathbf{\Pi}$, that recreates $o(D, x_0)$. The corollary then follows from Proposition 2.

7 Conclusion

The main contributions are

▷ that although dynamical systems defined using epistemic action models can produce only very limited orbits, dynamical systems that control when particular action models are used may produce orbits sufficient for most modeling purposes, and

▷ that the three methods for controlling which action models are applied are equivalent under the presented conditions.

The first result shows that DEL-based dynamical systems provide a rich framework for producing mathematically specified models of information dynamics. The latter shows that there are multiple ways of extending the DEL toolbox compatible with modeling using dynamical systems.

It would be interesting to make an in-depth comparison between DEL protocols and DEL-based dynamical systems, comparing the orbits they may produce and under which conditions such might be equivalent. Two considerations here involve the finite nature of DEL protocols, guaranteeing finite variation not guaranteed by DEL-based dynamical systems, and the 'bisimulation respecting' behavior of DEL-based dynamical systems, which is not necessarily followed by DEL protocols. Obtaining such results could be used to link DEL-based dynamical systems with Epistemic Temporal Logic via the results in [3].

Moreover, it would be interesting to investigate any deeper relationship between dynamic epistemic logic and dynamical systems; the latter field is well-developed, and one could envision that methods and results may be transferable.

References

1. A. Baltag and L. S. Moss. Logics for Epistemic Programs. *Synthese*, 139(2):165–224, 2004.
2. A. Baltag, L. S. Moss, and S. Solecki. The Logic of Public Announcements, Common Knowledge, and Private Suspicions (extended abstract). In *Proc. of the Intl. Conf. TARK 1998*, pages 43–56. Morgan Kaufmann Publishers, 1998.
3. J. van Benthem, J. Gerbrandy, T. Hoshi, and E. Pacuit. Merging Frameworks for Interaction. *Journal of Philosophical Logic*, 38(5):491–526, 2009.
4. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.
5. T. Bolander and M. Birkegaard. Epistemic planning for single- and multi-agent systems. *Journal of Applied Non-Classical Logics*, 21(1):9–34, 2011.
6. H. W. Broer and F. Takens. Preliminaries of Dynamical Systems Theory. In B. Hasselblatt, H. W. Broer, and F. Takens, editors, *Handbook of Dynamical Systems, Vol. 3*. North-Holland, 2010.
7. C. Dégremont. *The Temporal Mind: Observations on the logic of belief change in interactive systems*. PhD thesis, University of Amsterdam, 2010.
8. 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)*, Texts in Logic and Games, Vol. 3, pages 87–117. Amsterdam University Press, 2008.
9. J. van Eijck, F. Sietsma, and Y. Wang. Composing models. *Journal of Applied Non-Classical Logics*, 21(3-4):397–425, 2011.
10. R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. The MIT Press, 1995.
11. V. Goranko and M. Otto. Model Theory of Modal Logic. In P. Blackburn, J. van Benthem, and F. Wolter, editors, *Handbook of Modal Logic*. Elsevier, 2008.
12. J. A. Plaza. Logics of public communications. In M. L. Emrich, M. S. Pfeifer, M. Hadzikadic, and Z. W. Ras, editors, *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, pages 201–216, 1989.
13. B. Rodenhäuser. A logic for extensional protocols. *Journal of Applied Non-Classical Logics*, 21(3-4):477–502, 2011.
14. D. Sarenac. Modal Logic for Qualitative Dynamics. In Olivier Roy, Patrick Girard, and Mathieu Marion, editors, *Dynamic Formal Epistemology*, volume 351 of *Synthese Library*, pages 75–101. Springer, 2011.
15. Y. Wang. *Epistemic Modelling and Protocol Dynamics*. Doctoral thesis, Universiteit van Amsterdam, 2010.