The Hintikka's world project

Sophie Pinchinat and François Schwarzentruber

No Institute Given

Abstract. We present an online software for getting started with epistemic reasoning. Playgrounds (muddy children, Sally and Anne, etc.) are proposed, whose specification is described in Dynamic Epistemic Logic. Those various playgrounds feature a pedagogical visual description of the current knowledge of each agent and the knowledge she has about the knowledge of the others, and so on, which is commonly called *higher-order knowledge*. Additionally, an end user, whether expert or not, can create its own playground by specifying the initial epistemic situation and run a finite sequence of events to observe the information change of each agent along this course of events. Beyond the aim of offering a concrete setting for experimenting on dynamic epistemic logic, the software displays the epistemic temporal tree structure resulting from the execution of the sequence of events, with precise information such as the truth value of propositions in a node and the event labels along its branch from the root.

To make the description of the software as friendly as it is, we propose a dialog-based paper where two protagonists, **Q** and **A**, have a conversation about the aims of this tool and the functionalities it offers. The paper is organized as follows. Section 1 is meant to motivate the Hintikka's world project. Sections 2 and 3 address the usefulness of the software for expert and non-expert users, respectively. Next, a use case of the tool is presented in Section 4, and perspectives for forthcoming improvements and developments are discussed in Section 5.

1 Overview of the Hintikka's world project

Q: What is the Hintikka's world project about?
A: First, let me say that the project name is inspired from other tools that have been developed to help people better acquainted with logic, namely *Tarski's world* ([8],[7]) and *Kripke's worlds* [17].

More precisely, Hintikka's world is a software[1] for experimenting Dynamic Epistemic Logic (DEL) ([6], [28], [27]). DEL extends epistemic logic ([22], [21])) and is designed for reasoning about knowledge and information change in a multi-agent system. A multi-agent system is an abstract model of some real situations which gather autonomous agents that can act and communicate, and that may have individual or collective goals. DEL has been designed to specify properties

[1] `http://people.irisa.fr/Francois.Schwarzentruber/hintikkasworld/`

of multi-agent systems. It is manifold. The first ingredient of a DEL specification is the description an initial situation by means of an *initial epistemic model*. The second ingredient of a DEL specification is the description of all the events that may occur along the time line by means of a finite set of *event models*. These two first ingredients form a *DEL presentation*.

The third ingredient of DEL is a language to express properties of the specified multi-agent system.

Q: What are the strong points in favor of DEL compared to other formalisms for specifying and analyzing multi-agent systems?
A: As opposed to other formalisms such as epistemic temporal logic [20] or alternating-time logic [1], DEL offers a powerful language to describe possible events. The language extends the traditional one of *STRIPS*, with pre/post-conditions [16]. Additionally, it is possible to pair events to reflect their indistinguishability for a given agent. The resulting structure where events are paired is called an *event model*.

This ability to describe a pairing of events is not considered in STRIPS and leads to a more involved operation when a possible event is triggered in a given world of the current epistemic model. The triggering of an event displays the information change caused by the event by an accurate mechanism based on the *update product* of the current epistemic model and the triggered event model.

Because of the pairing of events, I wish that you believe that I believe that DEL is well-suited to express higher-order knowledge (a knows that b knows ...). Note that higher-order knowledge of agents is relevant in many applications: game theory [5], robotics ([25], [13]), specifications of distributed systems [19], etc. Besides, the research community investigates a large amount of effort to solve a natural extension of classical planning [18], called the *epistemic planning*, where the plans to be synthesized are sequences of event models and the goal to be achieved with such plans is in general an epistemic property [3], [10], [11].

Q: Are there other softwares that deal with DEL?
A: Recently, model checking in DEL has been proved to be possible in practice [29], and brought forth the tool DEMO [30], [29]. Still, using DEMO requires some skills in logic and in Haskell, preventing the tool from being used by "non-experts". On the contrary, Hintikka's world is designed to reach out to a wider public, such as people who do not know anything about DEL, but can have a smell of what information change and high-ordered knowledge can be. Nevertheless, Hintikka's world also propounds expert-level functionalities, that do not exist in DEMO.

2 For expert users

Q: What functionalities Hintikka's world offers to experts?

A: Typically, with the custom entry in the scrolling menu, the user can specify a DEL presentation and "run" it. Before explaining in detail the functionalities, we recall the formal setting of DEL, focusing on the DEL presentations and on the update product, leaving apart the language to state properties.

DEL framework is grounded on two central notions: on the one hand, *epistemic models*, namely classic *Kripke structures* and the *possible world* semantics – widely used in logics of knowledge [15], and *event models*.

In the sequel, we let Ag denote a finite set of agents, with typical elements $i, j, \ldots$. Also we consider an infinite set of atomic propositions AP whose typical elements are $p, q, r, \ldots$.

Definition 1 (Epistemic models). *An* epistemic model *is a structure*

$$\mathcal{M} = (W, \{\xrightarrow{i}\}_{i \in \mathsf{Ag}}, V)$$

where W is the non-empty set of worlds*, for each agent $i \in \mathsf{Ag}$, $\xrightarrow{i} \subseteq W \times W$ is its* accessibility relation*, and $V : W \to 2^{\mathsf{AP}}$ is the* valuation *of worlds by atomic propositions.*

A specific world w of the epistemic model $\mathcal{M}$ may be distinguished (to highlight w as the "actual" world), in which case we note the resulting *pointed epistemic model* as $(\mathcal{M}, w)$.

Example 1 (consecutive numbers). Suppose that agents a and b are each given a number between 0 and 10. Suppose also that they commonly know that the difference between a's number and b's number is at most 1. For instance, the actual world may be a's number is 2 and b's number is 3. In that actual world, a imagines another possible world in which b's number is 1. We introduce atomic propositions of the form $p_{i,n}$ read as 'agent i's number is n'. The epistemic model corresponding to this example called *consecutive numbers* is $\mathcal{M} = (W, \{\xrightarrow{i}\}_{i \in \mathsf{Ag}}, V)$ where:

- $W = \{(n_a, n_b) \in \{0, \ldots, 10\}^2 \mid |n_a - n_b| = 1\}$;
- $(n_a, n_b) \xrightarrow{i} (n'_a, n'_b)$ iff $n_i = n'_i$ (agent i knows its own number);
- $V((n_a, n_b)) = \{p_{a,n_a}, p_{b,n_b}\}$.

Figure 1 shows the epistemic model, where the actual world is the one labeled by $\mathsf{a2}, \mathsf{b3}$.

Before defining events models, we briefly recall epistemic logic, which we shall denote by $\mathcal{L}_{\mathsf{EL}}$. The syntax of $\mathcal{L}_{\mathsf{EL}}$ is as follows: $\varphi \ ::= \ p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_i\varphi$.

The semantics of $\mathcal{L}_{\mathsf{EL}}$ rests upon epistemic models. The truth value of a formula is evaluated in a pointed epistemic model; as the semantics for the usual inductive cases is forthright, we only concern ourselves with the knowledge operator for arbitrary agent i and $\mathcal{L}_{\mathsf{EL}}$ formula φ: $(\mathcal{M}, w) \models K_i\varphi$ iff for all worlds w' such that $w \xrightarrow{i} w'$, $(\mathcal{M}, w') \models \varphi$. In layman's terms, $K_i\varphi$ means that

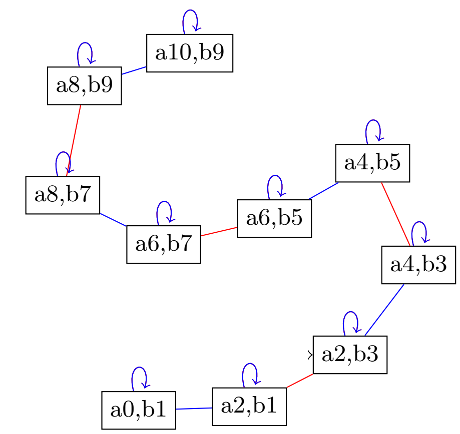

Fig. 1: Epistemic model for the consecutive numbers example.

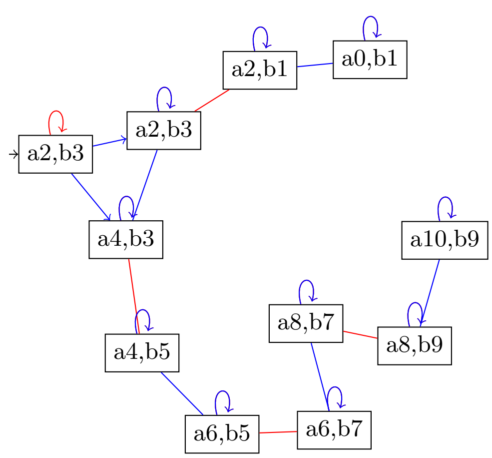

Fig. 2: The updated epistemic model for the consecutive numbers example, after the private announcement to a that b's number is 3.

proposition φ stands in all of the worlds thought by agent i to be possible when the real world is w. Formulas in $\mathcal{L}_{\mathsf{EL}}$ are used to describe pre and postconditions of events in event models.

Definition 2 (Event models). *An* event model $\mathcal{E}$ *is a structure*

$$(E, (\xrightarrow{i})_{i \in \mathit{Ag}}, \mathit{pre}, \mathit{post})$$

where E is a finite set (of events), each $\xrightarrow{i}$ is a binary (accessibility) relation on E, $\mathit{pre} : E \to \mathcal{L}_{\mathsf{EL}}$ is a precondition function, and $\mathit{post} : E \to \mathit{AP} \to \mathcal{L}_{\mathsf{EL}}$ is a postcondition function.

Analogously as for worlds, we designate $(\mathcal{E}, e)$ as a *pointed event model* where e is some event from $\mathcal{E}$ (the "true" event).

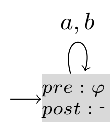

Fig. 3: The point event model for the public announcement of φ.

Example 2 (Pointed event model for public announcement). The *public announcement* of φ as considered in [24] corresponds to an event where it is common knowledge that a true message φ is received by all agents. The public announcement of φ is modeled by a pointed event model depicted in Figure 3. It has a single possible event, a reflexive loop on it for each agent. Its precondition is φ and its postcondition is empty (no assignment is made so that the values of atomic propositions remain unchanged).

We consider now a more involved example of a private announcement.

Example 3 (Pointed event model for private announcement). For the case of two agents a and b, the *private announcement* of the message φ to agent a is model-led by the following pointed event model of Figure 4.

The effect of events on worlds is formalized by means of the *update product.*

Definition 3 (Update product). *Given a pointed epistemic model $(\mathcal{M}, w_0)$ and a pointed event model $(\mathcal{E}, e_0)$ where $w_0 \models \mathit{pre}(e_0)$, the* update product *$(\mathcal{M}, w_0) \otimes (\mathcal{E}, e_0)$ is a new epistemic model (hence new worlds) defined as follows.*

$$(\mathcal{M}, w_0) \otimes (\mathcal{E}, e_0) = ((W^{\otimes}, (\xrightarrow{i}^{\otimes})_{i \in \mathit{Ag}}, V^{\otimes}), (w_0, e_0))$$

where

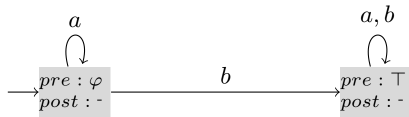

Fig. 4: For two agents a and b, the pointed event model for the private announcement of φ to agent a.

- $W^{\otimes} = \{(w, e) \in W \times E \mid (\mathcal{M}, w) \models \mathsf{pre}(e)\}$ *(each world satisfying an event's precondition gives rise to a new world through that event);*
- $(w, e) \xrightarrow{i}^{\otimes} (w', e')$ *iff* $(w \xrightarrow{i} w'$ *and* $e \xrightarrow{i} e')$ *(the epistemic relation for agent* i *conflates new worlds iff it conflated the associated old worlds as well as the events);*
- $V^{\otimes}(w, e) = \{p \in \mathsf{AP} \mid (\mathcal{M}, w) \models \mathsf{post}(e)(p)\}$ *(any new world observes the postcondition that the associated event required from the old world).*

For ease of notation and for the sake of the iteration process described below, a world $(w, e) \in W^{\otimes}$ is simply written we.

Example 4. Figure 2 shows $(\mathcal{M}, w_0) \otimes (\mathcal{E}, e_0)$ where $(\mathcal{M}, w_0)$ is given in figure 1 and $(\mathcal{E}, e_0)$ is given in 4 with $\varphi := p_{b,3}$.

Without loss of generality, instead of considering a finite family $\{(\mathcal{E}_k, e_k)\}_{k \in K}$ of pointed event model, we may consider pointed event models of the form $(\mathcal{E}, e_k)$ where $\mathcal{E}$ is the disjoint union of all $\mathcal{E}_k$. We then write $(\mathcal{M} \otimes \mathcal{E}^n, we_1 \dots e_n)$ for the epistemic model $(\mathcal{M}, w) \otimes (\mathcal{E}, e_1) \dots \otimes (\mathcal{E}, e_n)$ obtained by triggering the course of events $e_1 \dots e_n$. A *trace* of $\mathcal{M} \otimes \mathcal{E}^n$ is a world in $\mathcal{M} \otimes \mathcal{E}^n$, whose typical element will be denoted by τ.

In Hintikka's world, one can see the picture of the epistemic model $\mathcal{M} \otimes \mathcal{E}^k$. Very conveniently, it is possible to acquire the Tikz code of the picture, say to be inserted into a LaTeX document. All those who have written papers on DEL, know how tedious it is to manually compute the k-th update $\mathcal{M} \otimes \mathcal{E}^k$, and how unexciting it is to write the Tikz code that to depict it.

Another useful functionality is also offered. We name *DEL presentation* a pair $(\mathcal{M}, \mathcal{E})$, sometimes equipped with a family of distinguished events $\{e_k\}_{k \in K}$. In order to merge knowledge and time, one can also consider the *DEL structure* $\mathcal{M}\mathcal{E}^*$ induced by a DEL presentation $(\mathcal{M}, \mathcal{E})$. DEL structures are akin to models in logic of knowledge and time, such as *interpreted systems* in epistemic temporal logic [20]. Intuitively those structures are forests. A node at depth n of a tree in this forest denotes a trace τ of $\mathcal{M} \otimes \mathcal{E}^n$. The relationship between a parent node τ and its child τ' in a tree, reflects the dynamics[2] of the system when triggering

[2] the dimension along the time line.

an event, so that $\tau' = \tau e$ where e is the triggered event. Finally, traces at the same depth in the forest, say n, may be connected by the binary relation $\xrightarrow{i}$ reflecting the accessibility relation for agent i in $\mathcal{M} \otimes \mathcal{E}^n$.

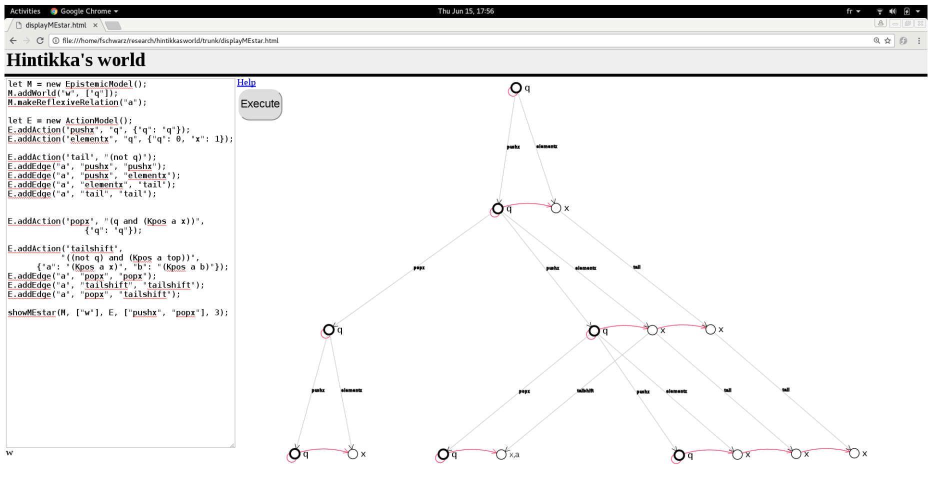

Fig. 5: A DEL structure generated by Hintikka's world

Definition 4 (DEL structures). *Let $\mathcal{M} = (W, \{\xrightarrow{i}\}_{i \in \mathsf{Ag}}, V)$ be an epistemic model and $\mathcal{E} = (E, (\xrightarrow{i})_{i \in \mathsf{Ag}}, \mathsf{pre}, \mathsf{post})$ be a event model. The* DEL structure *denoted by the DEL presentation $(\mathcal{M}, \mathcal{E})$ is the infinite structure*

$$\mathcal{M}\mathcal{E}^* = (WE^*, \rightarrow, \{\xrightarrow{i}\}_{i \in \mathsf{Ag}}, V^*),$$

where

- $WE^* = \bigcup_{n \geq 0} \{\tau \in WE^n \mid \tau$ *is a trace of* $\mathcal{M} \otimes \mathcal{E}^n\}$[3]
- $\rightarrow \subseteq WE^* \times E \times WE^*$ *is defined by* $(\tau, e, \tau') \in \rightarrow$ *whenever* $\tau \models \mathsf{pre}(e)$ *and* $\tau' = \tau e$ *(we simply write* $\tau \xrightarrow{e} \tau'$*),*
- $\tau \xrightarrow{i} \tau'$ *if* τ, τ' *are traces of* $\mathcal{M} \otimes \mathcal{E}^n$*, for some* n*, and* $\tau \xrightarrow{i} \tau'$ *in* $\mathcal{M} \otimes \mathcal{E}^n$*, and*
- $V^*(\tau)$ *is the valuation of* τ *in its update* $\mathcal{M} \otimes \mathcal{E}^n$*.*

In Hintikka's world, the DEL structure $\mathcal{M}\mathcal{E}^*$ can be displayed up to some depth. We came to developing this functionality while investigating the structural properties of DEL structures: whereas a DEL presentation whose event

[3] notice that a trace belongs to a unique update.

preconditions and postconditions are propositional has a regular language of traces; actually, such a DEL presentation denotes a DEL structure that is automatic[4][4,14]. On the other hand, an arbitrary DEL presentation may lead to a DEL structure with a non-regular set traces [14], and some DEL structures can even be proved to be non-automatic; this last observation results from the reduction of two-counter machine to the epistemic planning problem proposed in [11]. Therefore, offering an opportunity to observe the DEL structure is very informative, and paves the way to understanding the expressive power of DEL presentations. Figure 5 displays an example of a DEL structure generated by the software up to depth 3.

3 For non-expert users

Q: Who do you think non-expert can be?
A: Non-expert users can be our colleagues working in connected research areas; typically verification, model-checking, game theory, etc. Indeed most of those colleagues are not yet familiar with epistemic logic, which is unfortunately not yet fully established in this community. The DEL framework, although more complex than epistemic logic has the advantage of introducing the dynamics of the system and the information change. This way, the mathematical representation of knowledge becomes something alive and closer to their intuition, even higher-order knowledge arises naturally.

Other non-expert users can be colleagues from culturally remote fields such as roboticians, psychologists, etc.

But the simplicity of the tool, with its well-thought built-in examples, makes it attainable by budding scientists, such as students in computer science or in philosophy, etc. Besides, we already experimented the tool with psychiatrists and psychologists at the IME[5], with a very positive feedback regarding their understanding of Kripke models and of higher-order knowledge these structures capture.

We are currently working on promising test beds for the tool. In June 2017, the workshop Robolog 2017[6] will be presented to robotocians, in order to discuss the relevance of logical approach in decision making. Also, we currently develop a training course for PhD students on dynamic epistemic logic with an intensive use of our tool Hintikka's world [23]; this course will be given at the next European Agent Systems Summer School in August 2017[7].

On a broader level, with Hintikka's world, we want to promote the teaching of modal logic as an adequate setting that clearly separates orthogonal notions of syntax and semantics, as opposed to begin with teaching first-order logic.

Q: What does Hintikka's world offer to non-expert users?

[4] in the sense of [9].
[5] Institut médico-éducatif of Rennes, France
[6] `http://people.irisa.fr/Francois.Schwarzentruber/robolog2017/`.
[7] `http://easss2017.ipipan.waw.pl/`

A: At a first stage, non-expert user may find built-in examples, in which they can execute event models and observe the resulting updated epistemic model. In a second stage, users can develop their own examples.
Q: How can user access the models?
A: We use comic strips! So intuitive ...

We have made a great effort to offer an easy way to report epistemic models. Figure 6 depicts the graphical user interface for the consecutive numbers example. In this example, two agents a and b[8] are given a number and it is common knowledge that the two numbers are consecutive. Possible events are two announces: one is agent a announces that "she does not know b's number" and the other is agent b announces that "he does not know a's number".

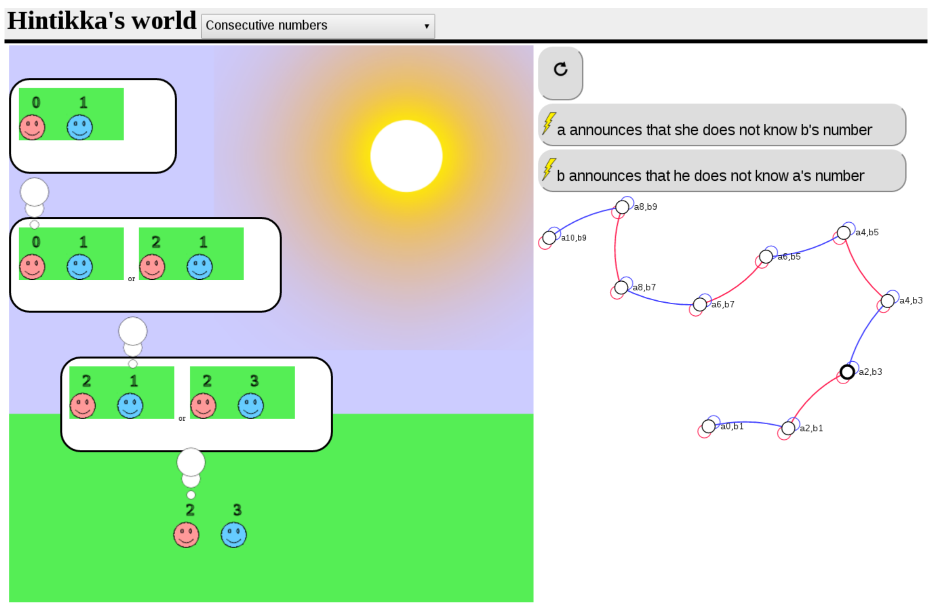

Fig. 6: Graphical user interface of *Hintikka's world*

What an agent visualises, that is all the possible worlds for her, is displayed as a thought bubble when clicking on her. In the initial situation of Figure 6, agent a has number 2 and agent b has number 3. One agent a has been clicked so that bubble reveals the possible words for her: she considers possible that agent b has number 1 and that agent b has number 3 (the actual world). Next agent b has been clicked in the former possible world, hence unfolding the pointed Kripke model of the current situation. At the same time, the user can observe the graph of the epistemic model where each possible world of the bubbles highlighted.

[8] In the comic strips the agents are with a face red and blue, respectively.

For each built-in example of the software, the worlds of the epistemic model are labeled by the relevant atomic propositions; for the consecutive numbers example of Figure 6, they are of the form `ai` for the fact "agent a has number i" and of the form `bj` for the fact "agent b has number j". On the top right of the comic strips, buttons for possible pointed event models (public announcement, public actions, private actions, etc.) are made available to the user. In Figure 6, the two announcements of the consecutive numbers example are proposed. By clicking one of them, the corresponding event takes place and the update product becomes the new current model. The user can then explore it as she could do for the epistemic model before the event was triggered.

4 Developing new examples

Q: You said that new examples can be developed. How should one proceed?
A: Do you have a particular example in mind?
Q: Hmmm, a friend of mine, that I will call H to ensure anonymity, would be delighted to hear about the *Russian cards* example. Do you know this example?

A: Of course, I know it! It is a setting with three agents a, b and c and seven cards $0, 1, 2, 3, 4, 5, 6$. Agents a and b hold three cards each and c holds the remaining card. The example demonstrates the existence of a public announcement leading to the common knowledge among a and b of their both hands, whereas c does not know their hands.

Let me tell you how easy it is to develop this example with Hintikka's world, which uses Javascript – a classic that most users should be familiar with, or easily get started for the few required knowledge needed in Hintikka's world. Notice that Javascript is a functional language with concise and elegant instructions (`map`, interactions, etc. [12]) that ease the description of the models.

Back to our development session, we should start by defining a new class – that we name `RussianCardsWorld` (see the architecture of Hintikka's world in Figure 9 on last page 13).

The class `RussianCardsWorld` is described in Figure 7a to define a world for the example. Its constructor takes as an input an array of propositions `trueProposititions` that reflects the actual world of the example. For instance, if agent c holds card 1, a holds cards 3, 5, 6 and b holds cards 0, 2, 4, we would consider input `truePropositions=["c1","a3","a5","a6","b0","b2","b4"]`. The instruction `super(truePropositions)` in the constructor code is a call to the constructor of the parent class `WorldValuation`, an abstract class for worlds in epistemic structures. The three remaining lines of code in the constructor are responsible for the position of the agents in the graphical user interface explained earlier. Also, the method `draw` allows the user to specify the graphical representation of a world for the developed example. It requires the standard `canvas` API of HTML5.

```
class RussianCardsWorld extends WorldValuation
{
  constructor(trueProposition) {
    super(trueProposition);
    this.agentPos["a"] = {x: 6+16, y: 32+16, r: 16};
    this.agentPos["b"] = {x: 48+16, y: 32+16, r: 16};
    this.agentPos["c"] = {x: 90+16, y: 32+16, r: 16};;
  }

  draw(context) {
    this.drawAgents(context);
   ...
  }
}
```

(a) Definition of the class RussianCardsWorld

```
...
let M = new EpistemicModel();
M.addWorld("w0123456",
new RussianCardsWorld(["a0","a1","a2","b3","b4","b5","c6"])
)
...
M.addEdge("a", "w0123456", "w0125634")
...
M.setPointedWorld("w0123456")
...
```

(b) An extract from the epistemic model specification.

```
...
agents.forEach(i =>
  M.addEdgeIf(i,
  (w1, w2) => [0, 1, 2, 3, 4, 5, 6].map((c) => i+c)
  .every((p) => (w1.modelCheck(p) == w2.modelCheck(p))
        )
              )
  );
...
```

(c) Another way to specify the epistemic model.

Fig. 7: The Russian cards example.

Q: WOW! Do you mean then that the GUI for the built-in examples would also work for this newly-developed Russian cards example? Clicking on an agent, seeing what she visualises, etc.?

A: Sure it will work! This is made possible by the fact that the new class inherits from the abstract class `WorldValuation`. But let us keep on with the new example of Russian cards.

You may now specify the pointed epistemic model as depicted in Figure 7b, or as in Figure 7c, by taking advantage of the functional nature of Javascript: this concise piece of code adds, for each agent i, an i-edge in M between two worlds $w1$ and $w2$ whenever the two worlds agree on the value on agent a's card numbers; formally, for every $p \in \{\texttt{a0}, \ldots, \texttt{a6}\}$, $w1 \models p$ iff $w2 \models p$.

Q: And how do you define event models?

A: Just like we specified epistemic models, as the mathematical structures are alike. Figure 8 is an example of an event model specification: this event describes agent a dropping Card 1 while other agents (b and c) do not notice it. Parameters of the method `addEvent` are as follows. The first parameter is the event name. The second one is its precondition, and the third one (optional) is its postcondition. For instance, the precondition of e is `a1` and its postcondition assigns false to `a1`.

```
let E = new ActionModel();
E.addEvent("e", "a1", {"a1": "bottom"});
E.addEvent("f", "top");
E.addEdge("a", "e", "e");
agents.forEach(i => E.addEdge(i, "f", "f");)
["b", "c"].forEach(i => E.addEdge(i, "e", "f"));
E.setPointedAction("e");
```

Fig. 8: The Russian cards example: an event model where agent a drops Card 1 and agents b and c do not notice it.

Once an event model `E` has been specified, the user can execute this event on her current epistemic model `M`, and obtain the new (updated) epistemic model; the needed piece of code is just `product(M, E)`. She can next iterate the execution of events that she has specified.

Q: The tool seems easy to use and very intuitive.

A: I am happy to hear that, and indeed colleagues who tried it gave us a very encouraging feedback.

5 Perspectives

Q: Your tool seems promising for teaching, explaining and experimenting dynamic epistemic logic. What do you plan for the future?
A: In the short term, we wish to improve the graphical user interface. To this aim, we have submitted a project in collaboration with the art school *l'institut supérieur des arts appliqués* (LISAA) in Rennes, France. We also want to implement heuristics for displaying the most relevant epistemic worlds when they are too many.

In the medium term, on a more technically side, we will connect *Hintikka's world* with model checkers for dynamic epistemic logic model checkers [30,29], while promoting symbolic appraoches [26]; we may also consider developing our own model checker to enrich the domain of the variables, such as enumerated types and infinite types like integers.

In a longer term, we target an involved functionality that solves epistemic planning problems. Those problems, acknowledged to be difficult ones have numerous applications and open the way to epistemic strategic reasoning in multi-agent systems. The generic problem of epistemic planning consists in synthesizing a finite sequence of events to be executed so that the reached epistemic situation fulfills a given epistemic formula [10]. Notice that we may modestly stick to *bounded* epistemic planning problems[9], because the general problem is undecidable [10], [3], [11].

Ultimately, the tool could be useful in several kinds of societal applications, such as the three following ones. First, it could contribute to teaching children how to reason about higher-order knowledge [2]. Second, it may provide an interface for displaying mental states of real human-aware robots [25], [13].

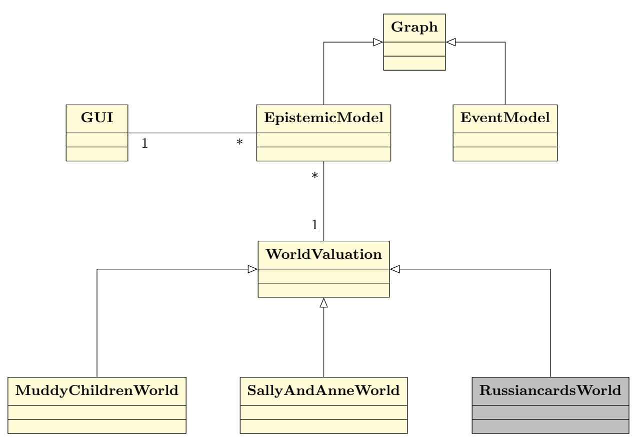

Fig. 9: Architecture of *Hintikka's world*

[9] that is by bounding the length of the plans.

Acknowledgments. We warmly thank all the members of the LogicA group: Tristan Charrier, Eva Soulier, Sébastien LeCong, Maxime Audinot, Florence Wacheux. Also, we are cheerful to thank Noël Plouzeau for dedicating his precious time to playing with Hintikka's world and letting us know that he enjoyed!

References

1. R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.
2. Burcu Arslan, Rineke Verbrugge, Niels Taatgen, and Bart Hollebrandse. Teaching children to attribute second-order false beliefs: A training study with feedback. In *Proceedings of the 37th Annual Meeting of the Cognitive Science Society, CogSci 2015, Pasadena, California, USA, July 22-25, 2015*, 2015.
3. Guillaume Aucher and Thomas Bolander. Undecidability in epistemic planning. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, 2013.
4. Guillaume Aucher, Bastien Maubert, and Sophie Pinchinat. Automata techniques for epistemic protocol synthesis. In *Proceedings 2nd International Workshop on Strategic Reasoning, SR 2014, Grenoble, France, April 5-6, 2014.*, pages 97–103, 2014.
5. Robert J. Aumann. Interactive epistemology I: knowledge. *Int. J. Game Theory*, 28(3):263–300, 1999.
6. Alexandru Baltag, Lawrence S Moss, and Slawomir Solecki. The logic of public announcements, common knowledge, and private suspicions. In *Proceedings of the 7th conference on Theoretical aspects of rationality and knowledge*, pages 43–56. Morgan Kaufmann Publishers Inc., 1998.
7. David Barker-Plummer, Jon Barwise, and John Etchemendy. Tarski's world: Revised and expanded. 2007.
8. Jon Barwise and John Etchemendy. The language of first-order logic including the macintosh version of tarski's world 4.0. 1993.
9. Achim Blumensath and Erich Grädel. Automatic Structures. In *Proceedings of 15th IEEE Symposium on Logic in Computer Science LICS 2000*, pages 51–62. IEEE, 2000.
10. Thomas Bolander and Mikkel Birkegaard Andersen. Epistemic planning for single and multi-agent systems. *Journal of Applied Non-Classical Logics*, 21(1):9–34, 2011.
11. Tristan Charrier, Bastien Maubert, and François Schwarzentruber. On the impact of modal depth in epistemic planning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 1030–1036, 2016.
12. Guy Cousineau and Michel Mauny. *The functional approach to programming.* Cambridge University Press, 1998.
13. Sandra Devin and Rachid Alami. An implemented theory of mind to improve human-robot shared plans execution. In *The Eleventh ACM/IEEE International Conference on Human Robot Interation, HRI 2016, Christchurch, New Zealand, March 7-10, 2016*, pages 319–326, 2016.
14. Gaëtan Douéneau-Tabot, Sophie Pinchinat, and François Schwarzentruber. Propriétés régulières des arbres. IRISA research report, 2016.

15. Ronald Fagin, Yoram Moses, Joseph Y Halpern, and Moshe Y Vardi. *Reasoning about knowledge*. MIT press, 2003.
16. Richard Fikes and Nils J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artif. Intell.*, 2(3/4):189–208, 1971.
17. Olivier Gasquet, Andreas Herzig, Bilal Said, and François Schwarzentruber. *Kripke's Worlds - An Introduction to Modal Logics via Tableaux*. Studies in Universal Logic. Birkhäuser, 2014.
18. Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: theory and practice*. Elsevier, 2004.
19. Joseph Y. Halpern and Ronald Fagin. Modelling knowledge and action in distributed systems. *Distributed Computing*, 3(4):159–177, 1989.
20. Joseph Y. Halpern and Moshe Y. Vardi. The complexity of reasoning about knowledge and time. 1. Lower bounds. *Journal of Computer and System Sciences*, 38(1):195–237, 1989.
21. Jaakko Hintikka. Reasoning about knowledge in philosophy: The paradigm of epistemic logic. In *Proceedings of the 1st Conference on Theoretical Aspects of Reasoning about Knowledge, Monterey, CA, March 1986*, pages 63–80, 1986.
22. Saul A Kripke. Semantical analysis of modal logic i normal modal propositional calculi. *Mathematical Logic Quarterly*, 9(5-6):67–96, 1963.
23. Sophie Pinchinat and François Schwarzentruber. Dynamic epistemic logic and its applications to plan/protocol synthesis, August 2017.
24. Jan Plaza. Logics of public communications. *Synthese*, 158(2):165–179, 2007.
25. Brian Scassellati. Theory of mind for a humanoid robot. *Auton. Robots*, 12(1):13–24, 2002.
26. Johan van Benthem, Jan van Eijck, Malvin Gattinger, and Kaile Su. Symbolic model checking for dynamic epistemic logic. In *Logic, Rationality, and Interaction - 5th International Workshop, LORI 2015 Taipei, Taiwan, October 28-31, 2015, Proceedings*, pages 366–378, 2015.
27. Johan van Benthem, Jan van Eijck, and Barteld P. Kooi. Logics of communication and change. *Inf. Comput.*, 204(11):1620–1662, 2006.
28. Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*. Springer, Dordecht, 2008.
29. Hans van Ditmarsch, Jan van Eijck, Ignacio Hernández-Antón, Floor Sietsma, Sunil Simon, and Fernando Soler-Toscano. Modelling cryptographic keys in dynamic epistemic logic with DEMO. In *Highlights on Practical Applications of Agents and Multi-Agent Systems - 10th International Conference on Practical Applications of Agents and Multi-Agent Systems, PAAMS 2012 Special Sessions, Salamanca, Spain, 28-30 March, 2012*, pages 155–162, 2012.
30. Jan van Eijck. Demo—a demo of epistemic modelling. In *Interactive Logic. Selected Papers from the 7th Augustus de Morgan Workshop, London*, volume 1, pages 303–362, 2007.