The Dynamic Turn in Strategy Logics

Blue Sky Ideas Track

Rustam Galimullin
University of Bergen
Bergen, Norway
rustam.galimullin@uib.no

Maksim Gladyshev
Utrecht University
Utrecht, The Netherlands
m.gladyshev@uu.nl

Munyque Mittelmann
LIPN, CNRS, Université Sorbonne Paris Nord
Villetaneuse, France
mittelmann@lipn.univ-paris13.fr

Nima Motamed
Utrecht University
Utrecht, The Netherlands
n.motamed@uu.nl

ABSTRACT

Strategy Logics are well-studied frameworks for the specification and verification of strategic abilities in multi-agent systems (MAS). However, the current generation of Strategy Logics is limited to static reasoning about fixed models of MAS. This limitation excludes a plethora of applications that require addressing dynamic changes and updates. Examples include verifying that a computer program executes correctly after an upgrade, automatically repairing MAS to meet safety requirements, and reasoning about robots operating in a dynamic environment. To address this limitation, we propose a new research agenda centered on enriching Strategy Logics with concepts and intuitions from Dynamic Epistemic Logic, aiming to develop a holistic and general framework that captures dynamic phenomena in MAS and facilitates their verification.

KEYWORDS

Strategy Logics; Dynamic Epistemic Logic; Verification; Synthesis

ACM Reference Format:
Rustam Galimullin, Maksim Gladyshev, Munyque Mittelmann, and Nima Motamed. 2026. The Dynamic Turn in Strategy Logics: Blue Sky Ideas Track. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 7 pages. https://doi.org/10.65109/WZKX4761

1 INTRODUCTION

Logics for reasoning about strategic abilities (Strategy Logics, or $\mathcal{SL}$ for short[1]) comprise a wide variety of formal frameworks devised to model, verify, and reason about the strategies of (coalitions of) autonomous agents in multi-agent systems (MAS). In this context, strategies (also known as plans or policies) describe how rational agents act to achieve their individual or collective goals. Strategic reasoning plays a crucial role in MAS as actions of one agent may influence the available strategies of another agent.

Specification and verification of MAS with strategy logics have been explored across a plethora of domains, such as neuro-symbolic reasoning [5], voting protocols [19, 64], and smart contracts [99, 100]. Properties that typically require verification can be broadly divided into two groups. *Functionality* properties require checking that honest or authorised agents can always achieve a promised or desired outcome. In the other group are *security*, or *safety*, properties that ensure that malicious or unauthorised agents are not able to complete specific, often harmful, tasks [63]. Thus, $\mathcal{SL}$ are especially prominent as an integral part of *safe AI*, where it is essential to ensure that a given system has provably correct and thus safe behaviour. For example, $\mathcal{SL}$ were used to ensure the functionality and safety of autonomous submarines [40], autonomous vehicles [68], and manufacturing robots [39].

Observe that many, if not all, areas of application of $\mathcal{SL}$ admit changing, or dynamic, scenarios. Reasoning about such scenarios, however, cannot be satisfactorily carried out using the existing $\mathcal{SL}$. For example, we may want to capture strategies arising for robots operating on arenas with varying conditions, or to reason about the effects of possible malfunctions in the robots. This echoes the motivation for resilience in computer science, i.e., the capacity to adapt to disruptive changes in the environment [51, 106]. Or, we may want to check whether a voting protocol remains secure after modifications affecting voters' strategies, such as switching from a single-winner to a multi-winner system. The same question applies to software upgrades. In short, we may want to express and verify complex properties of a system, such as *"the system is safe now, and after we implement this update (change, upgrade), it will still be safe"*. Statements like this cannot currently be adequately expressed and verified in classic, static $\mathcal{SL}$, as they *lack the necessary operators to capture model changes.* This creates a roadblock to efficient verification of dynamic MAS and hinders the automatic synthesis of model repairs.

In order to address this limitation, we propose the creation of *a new generation of strategy logics* capable of reasoning about dynamic phenomena in MAS. To this end, we consider the intuitions and results from *Dynamic Epistemic Logic* (DEL) [103] to be of significant importance. Numerous variants of DEL are used to model changes of agents' knowledge and beliefs resulting from information-changing events. On the semantic level, this is represented by changes, or updates, of the given DEL model (called *epistemic model*). In this paper, we do *not* deal with agents' knowledge; instead, we use

---

[1] We write $\mathcal{SL}$ to disambiguate the general family of logics for reasoning about strategic abilities and the eponymous Strategy Logic (SL) [84], which belongs to $\mathcal{SL}$.

The title is inspired by the agenda of the *dynamic turn* in logic [94, 95], where dynamic changes of models in modal logic take centre stage. See more on this in, e.g., [17, 54].

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. https://doi.org/10.65109/WZKX4761

intuitions from DEL to enrich $\mathcal{SL}$ and thus capture *reasoning about the dynamics of strategic abilities*, and hence engender *a general approach for reasoning about dynamic phenomena in MAS.*

To achieve this goal, we need to tackle three challenges (see Section 3). First, we need to design extensions of various $\mathcal{SL}$ with new operators that will capture the dynamics of the underlying model. Next, given the syntax and semantics of these operators, we must automatically synthesise updates that specify how to repair a faulty model. Finally, to support practical use, we need to develop verification and synthesis tools for this new generation of $\mathcal{SL}$.

2 STATE OF THE ART

Strategy Logics. In this work, we focus on $\mathcal{SL}$ interpreted on *concurrent game models* (CGMs) [10], which are essentially graphs with labelled nodes and transitions. These logics use the constructs $\langle\langle C\rangle\rangle\varphi$ that are read as 'coalition C can force φ (no matter what agents outside of C do)'. We can also read $\langle\langle C\rangle\rangle\varphi$ as 'coalition C has a strategy to make φ true'. The meaning of such constructs is determined by transitions in a CGM and the agents' strategies. Once constructs $\langle\langle C\rangle\rangle\varphi$ are added to the language of propositional logic, we obtain *coalition logic* (CL) [88, 89] that captures reasoning only about immediate (next-step) outcomes in game-like scenarios.

CL is arguably the simplest logic in the family $\mathcal{SL}$. Other notable representatives of the family are the *alternating-time temporal logic* (ATL) and its more expressive cousin ATL* [10, 75]. They extend CL with temporal features such as Always, Eventually, and Until. An even more expressive logic is the eponymous *strategy logic* (SL) [84] and its fragments [20, 83], which additionally allow explicit quantification over agents' strategies.

Many variants of the CGM-based logics mentioned so far, i.e., CL, ATL, and SL, have been proposed and studied in the literature. Examples include *ATL with explicit strategies* [107], *coalition action and strategy logics* [26, 30], *socially friendly and group protecting CLs* [56, 57], *resource-bounded ATL* [7, 87], *ATL with imperfect recall* [92], *SL with imperfect information* [22], and *probabilistic extensions of ATL* [35, 62, 85] and *SL* [11, 86]. These are all different logics with varying expressivity and complexities. Namely, the complexity of model checking ranges from solvable in P for CL and ATL, to PSPACE-complete and 2EXPTIME-complete for ATL* (with memoryless and perfect recall strategies, respectively), and to undecidable for both ATL and ATL* with perfect recall strategies and imperfect information [63]. This variety notwithstanding, all of the mentioned logics work on static models and do not admit model dynamics.

Example 2.1 (Adapted from [99]). Atomic swap is a smart contract in which agents A (for Anna) and B (for Brita) would like to swap their assets, a and b, respectively. The tricky part is that the parties may not behave according to the agreement, and may, for example, withhold their asset while receiving the asset of their counterpart. Thus, we cannot assume any trust between the agents. To verify the functionality of (a model of) the smart contract, we can express the situation when Anna and Brita have deposited their assets as $dep(A,a) \wedge dep(B,b)$, and the target condition that they have swapped the assets as $has(A,b) \wedge has(B,a)$. Then, a property we may want to check is 'Anna can ensure that whenever the assets are deposited, the target condition is eventually reached, no matter what Brita does'. This can be written as the ATL* formula

$$\varphi := \langle\langle A\rangle\rangle \mathsf{Always}(dep(A,a) \wedge dep(B,b) \rightarrow \mathsf{Eventually}(has(A,b) \wedge has(B,a))).$$

In the example, the underlying assumption is that the situation at hand is fixed and is not bound to change. However, it may be crucial to verify not only that φ is satisfied in the current model, but also that it will hold after we implement changes in the model (e.g., upgrading atomic swap to grant Anna and Brita more options). This cannot be done with the standard CL, ATL, ATL*, and SL, and we argue for the need for a new family of $\mathcal{SL}$ in Section 3.1.

Strategic Reasoning in DEL. We believe that ideas from DEL can enrich $\mathcal{SL}$ to allow for a general framework for reasoning about dynamics in MAS. Therefore, we propose to explore the DEL-to-$\mathcal{SL}$ direction, which hitherto has been considered only sporadically in the literature. The opposite $\mathcal{SL}$-to-DEL direction, though, has been investigated relatively thoroughly with examples including *concurrent DEL games* [79, 80], *alternating-time temporal DEL* [38], *coalition announcements* [3, 45], and other forms of *strategic multi-agent communication* (see, e.g., [1, 50]). We mention the work on incorporating strategic reasoning into DEL, because we want to argue that the two communities, those of $\mathcal{SL}$ and DEL, have much to offer each other, and with our work, we wish to foster interaction in the DEL-to-$\mathcal{SL}$ direction, which has been greatly overlooked.

Existing Approaches to Dynamics in $\mathcal{SL}$. Some dynamic phenomena have already been considered in the literature on $\mathcal{SL}$ with, perhaps, the prime example being *normative reasoning* for MAS. For normative reasoning, we want to specify which actions of agents conform to the system's norms and social laws (see, e.g., [2, 6, 8, 28, 49, 60, 93, 98]). The main idea there is to divide the set of available actions for each agent into those that are permitted and those that are prohibited. Then, after the implementation of such a norm or a law, further evaluation happens in an *updated CGM.*

Another example is *obstruction logics* [31–33], where agents' abilities to cooperate and execute their strategies can be hindered by an external force, called the *Demon*. As in the normative setting, the Demon is able to disable some transitions and thus impact the strategic abilities of the agents in a system. With such a setting, the authors strive to verify the safety of distributed systems against cyberattacks. A recent variant of the formalism [29] also introduces the *Angel*, who can restore transitions, and can either compete or cooperate with the Demon. A somewhat similar problem is called *module checking* [66, 73], where agents interact with a non-deterministic environment that may inhibit access to certain paths of the computation tree.

The presented examples of works on $\mathcal{SL}$ with dynamic features are concerned with somewhat particular applications: normative systems in the former, and cybersecurity in the latter. With this work, **our goal** is to pursue a **unifying general approach to reasoning about dynamic phenomena in MAS**. Preliminary steps towards reaching the goal were made with the introduction of *dynamic CL* [45, 46], which allows for reasoning about granting and revoking abilities to and from agents, and *logic for ATL model-building* (LAMB) [47], which allows for combining simple model-changing operations, like adding a state or redirecting an arrow, into complex updates. However, the potential of 'dynamifying' $\mathcal{SL}$ is largely unexplored and calls for a systematic investigation.

3 CHALLENGES

In this section, we present three challenges that we deem particularly important for the emerging research field of *dynamic strategy logics*. These challenges partially build on each other, starting from theoretical foundations and culminating in practical tools.

3.1 Reasoning About the Dynamics of Ability

The first challenge in developing a general reasoning framework for $\mathcal{SL}$ dynamics is to propose meaningful ways of updating, or changing, models and agents' strategies, and to formally study their expressivity and computational complexity. There is a broad set of system properties that can be changed to affect strategies: agents' actions, their effects, facts about the environment, the number of agents, and any combination thereof.

Example 3.1. Continuing with Example 2.1, assume that we want to introduce additional functionality to atomic swap by allowing each agent to reject the swap, thus restoring the *status quo*. To capture such an upgrade, we need to be able to modify the agents' available actions and the resulting outcomes. Endowing users A and B with the possibility to reject a swap alters their powers dramatically. In particular, if in the original contract agent A could force the target swap condition, expressed by the formula φ in Example 2.1, in the upgraded smart contract this is no longer the case, as agent B can reject the swap altogether. Formally speaking, we would like our new $\mathcal{SL}$ to be able to express properties like $[\text{add\_reject}]\neg\varphi$, where the dynamic operator $[\text{add\_reject}]$ updates the given model by providing A and B with the option to reject the swap.

Although our examples are quite simple, models of real-world MAS are usually large (see [40, 64]). Hence, *from the practical perspective*, having a well-defined semantics for model updates would allow us to automate the construction of an updated model, instead of trying to modify the model 'by hand'. Moreover, we envision our update operators as formulas of a logic, and there are two clear advantages of this approach. First, this would allow a succinct, high-level representation of the effects of an update, as opposed to a low-level full description of all changes in a particular model. Second, in this way our updates will be model-agnostic, meaning that we can verify how the same update affects different models.

As a starting point, we draw inspiration from *dynamic epistemic logic* (DEL) [103], which models how agents' knowledge changes through epistemic actions. Although DEL's *epistemic models* differ substantially from CGMs, both are essentially labelled graphs. We therefore expect that some high-level intuitions about updating the former will translate into intuitions about updating the latter.

Let us provide three examples of how DEL intuitions can guide us in the quest for defining and exploring CGM updates. These examples are meant as first approximations of our ideas and are by no means complete. First, restricting CGMs to only a set of 'allowed' or 'enabled' states may correspond to the action of a *public announcement* [90] in DEL, which models the effects of all agents publicly and simultaneously receiving the same piece of information. As a result of a public announcement, all states in an epistemic model that disagree with an announced formula are removed. Second, we can also restrict transitions. This is exactly the case of normative reasoning discussed above. Its closest DEL counterparts may be various types of *arrow updates* [12, 13, 70]. Finally, changing the facts about the environment can be modelled by using substitutions [69], i.e., changing valuations of propositions in states according to some condition.

We have given just three examples of how intuitions about DEL updates can help define interesting updates for various $\mathcal{SL}$. There are many more exciting possibilities to update a CGM, and we mentioned some preliminary results [45–47] in Section 2.

The model-changing nature of DEL often yields interesting results. For example, some ways of updating transitions in a model do not grant us any additional expressivity [70, 71], while others do [12, 13, 96]. We expect similar results for new $\mathcal{SL}$: some update operators will be expressively harmless, others will not.

New logics with update operators for CGMs can then be naturally extended further by allowing quantification over these operators. In this way, we shift the focus from the effects of a particular update to the question of the (non-)existence of an update leading to some (un)desired outcome. Allowing quantification over updates is not merely a potentially interesting extension of new dynamic $\mathcal{SL}$. Rather, it adds an extra, more general, dimension to verifying the functionality and security of a given system.

From the perspective of *security*, quantification will allow us to verify that none of the updates possible in a given system will lead to an unsafe state. From the perspective of *functionality*, we can use quantification to reason whether there is an update that will endow a system with certain desired properties.

Quantification in DEL is relatively well studied (see [101] for an overview), and as a rule it drastically impacts expressivity [14, 58, 102, 104] and computational complexity [48, 72]. Moreover, quantifying over DEL updates usually leads to quite unexpected results, and some classic logical questions become much more complex to answer. For example, even simple logics with public announcements become undecidable once we allow quantification over announcements [4]. At the same time, quantifying over seemingly complex *action models* [15, 16] that allow non-public actions such as private announcements is not only decidable, but adds no expressivity compared to the standard epistemic logic [58].

Hence, there is an exciting opportunity for a systematic exploration of quantified versions of the new $\mathcal{SL}$, and to study those that offer the best trade-off in the 'expressivity vs. complexity' dichotomy. Of special interest is the existential quantifier, as it may shed light on a possible solution to the synthesis problem described in Section 3.2. In particular, it could allow us to analyse if we can provide a *constructive* way to produce an update which, once applied to the initial model, results in an updated CGM that satisfies some desired property. We believe that this approach would be a great start towards addressing the problem of synthesising CGM updates. This problem is, however, not trivial: in the context of DEL, checking whether there is a desired update does not always produce such an update. This is true, for example, in the case of quantified public announcements, where verifying the existence of an announcement leading to a certain epistemic goal is done via checking all possible submodels of a given epistemic model and does not produce the desired announcement expressed in a formula of epistemic logic. Constructing such a formula post-hoc usually results in an exponentially long formula, and the formula is by no means guaranteed to be minimal [9].

3.2 Automated Synthesis

Once there are meaningful and succinct approaches to updating CGMs, the next challenge is to tackle the *synthesis problem*, which is defined as follows: *given a starting model and a target formula, construct an update that transforms the model so that the target formula is satisfied.* The crux of the problem is that we are not given a set of possibly successful updates; instead, we need to create it. This is relevant when remodelling the whole CGM from scratch is unfeasible or undesirable. The solution to the synthesis problem will provide a syntactic update, expressed in some desired logic, that can serve as a protocol or instruction telling the designers how to fix the model and the corresponding MAS.

Example 3.2. Going back to Example 2.1, assume that a current model of the atomic swap smart contract does not satisfy the property φ. One way to proceed would be to go back to the drawing board with domain experts and try either to fix the existing model by hand or create an entirely new model. Needless to say, both options are not always desirable or even feasible. What we would prefer instead is to automatically obtain an update [update] that makes φ true: $[\mathtt{update}]\varphi$. Such an update may be relatively simple, like [add_reject] from Example 3.1, or composite, consisting of a combination of model-changing operators.

Although it might seem that the existential quantification from Section 3.1 is an instance of the synthesis problem, this is not always the case. As we have pointed out in Section 3.1, *checking the existence of an update* leading to a certain outcome and *constructing the update* are not always synonymous, and the corresponding computational problems may even have different complexities. One of the reasons for such a discrepancy is that, at least in DEL, checking the existence of an update can be done at the level of models (e.g., we check all submodels in the case of public announcements), whereas constructing the corresponding update must happen at the level of syntax, since our updates are also syntactic objects in a given logic. Ideally, of course, one would prefer to check the existence of an update by explicitly constructing the desired update, if it exists.

A tempting challenge here is to investigate whether and under which conditions the synthesis problem for our new logics is solvable, and then determine its computational complexity. Identifying the theoretical computational complexity may guide one in the choice of new $\mathcal{SL}$ for which it is feasible to create software for automated verification and synthesis (see Section 3.3).

We expect DEL to offer helpful intuitions, given that the synthesis problem is well-studied for some of its variants [58, 105]. Another approach is to employ *model and strategy repair* methods used for temporal logics (see [23, 25, 34]), game descriptions [59], reachability games [44], and AI planning [21]. Automated synthesis of $\mathcal{SL}$ updates will allow us to provide model repairs for complex MAS with strategic and rational behaviour of the agents. For example, we could reason about how to update a given MAS such that it is a Nash equilibrium to execute only safe actions and strategies.

One may also want not only to solve the synthesis problem, but also to solve it efficiently. In particular, we may require synthesised updates to be minimal in their lengths or sizes so that they can be easily understood by a human designer of a given MAS.

3.3 Tools for Verification and Synthesis

The third challenge is to implement verification and synthesis software for the new dynamic $\mathcal{SL}$. There are quite a few successful tools for model checking classic, static, $\mathcal{SL}$ such as MCMAS [78], MCK [52], STV [74], and the recently developed VITAMIN [42]. Extending these existing tools to capture the new $\mathcal{SL}$, both with and without quantification, is a highly promising direction.

There will, however, be a challenge of finding a compact representation of updates, in addition to the most efficient ways to quantify over them. Naive implementations of model-checking algorithms for quantified new $\mathcal{SL}$ will almost certainly require time at least exponential in the size of a model and a formula (although they may require only a polynomial amount of space). One possible way to tackle the update representation issue is by using binary decision diagrams (BDDs) [27]. This approach seems promising as DEL updates have already been represented as BDDs in the SMCDEL tool [53] for DEL model checking (see also [18, 82, 97]).

Given a model and some desired property, a model checker, in general, can usually only tell whether the model satisfies the property. If the answer is 'no', then we, as designers of the model, have to analyse why the property is not satisfied, and how to change the model to satisfy it. If the given model is large, and if we want to ensure some complex temporal or strategic property, updating the model 'by hand' may prove to be a highly non-trivial task. That is why a further challenge lies in implementing extensions of model checkers with the means to solve the synthesis problem for a given, potentially infinite, class of updates. What such a class may be and how hard the corresponding synthesis problem is, will be informed by the theoretical results on the synthesis problem.

Of course, creating software for model checking strategy logics and solving their synthesis problems is not straightforward, and we expect some of the known problems in the field to arise for our logics. In particular, naive implementations of verification algorithms usually suffer from the state explosion problem [37]. To tackle this, one can employ some known techniques for various practice-oriented model checkers for $\mathcal{SL}$ and related logics, such as bounded model checking (see, e.g., [24, 61]), partial order reductions [67, 77], fixpoint approximations [65], and abstractions [36, 55, 76]. The development of tools for the synthesis of $\mathcal{SL}$ updates is a new challenge, and one can start by exploring the potential of ideas behind the tools for reactive synthesis from specifications expressed in *linear temporal logic* (LTL) [43]. Examples of such tools include BoSy [41], Strix [81], and ltlsynt [91].

4 CONCLUSION

We argue for a general approach to the dynamics of strategic abilities in MAS. This approach will not only systematise existing dynamic features of $\mathcal{SL}$, such as normative reasoning, but also capture a wide range of dynamic scenarios in computer science and AI. Drawing inspiration from Dynamic Epistemic Logic, our proposed research agenda centres on enriching the existing static $\mathcal{SL}$ with model updates. We highlighted three key challenges: defining meaningful model updates and their semantics, automatically synthesising updates that repair models, and developing practical verification and synthesis tools for the next generation of $\mathcal{SL}$.

REFERENCES

[1] Thomas Ågotnes, Philippe Balbiani, Hans van Ditmarsch, and Pablo Seban. 2010. Group announcement logic. *Journal of Applied Logic* 8, 1 (2010), 62–81. https://doi.org/10.1016/j.jal.2008.12.002

[2] Thomas Ågotnes, Wiebe van der Hoek, Juan A. Rodríguez-Aguilar, Carles Sierra, and Michael Wooldridge. 2007. On the logic of normative systems. In *Proceedings of the 20th IJCAI*, Manuela M. Veloso (Ed.). 1175–1180.

[3] Thomas Ågotnes and Hans van Ditmarsch. 2008. Coalitions and announcements. In *Proceedings of the 7th AAMAS*, Lin Padgham, David C. Parkes, Jörg P. Müller, and Simon Parsons (Eds.). IFAAMAS, 673–680.

[4] Thomas Ågotnes, Hans van Ditmarsch, and Tim French. 2016. The undecidability of quantified announcements. *Studia Logica* 104, 4 (2016), 597–640. https://doi.org/10.1007/s11225-016-9657-0

[5] Michael E. Akintunde, Elena Botoeva, Panagiotis Kouvaros, and Alessio Lomuscio. 2020. Verifying Strategic Abilities of Neural-symbolic Multi-agent Systems. In *Proceedings of the 17th KR*, Diego Calvanese, Esra Erdem, and Michael Thielscher (Eds.). 22–32. https://doi.org/10.24963/kr.2020/3

[6] Natasha Alechina, Giuseppe De Giacomo, Brian Logan, and Giuseppe Perelli. 2022. Automatic Synthesis of Dynamic Norms for Multi-Agent Systems. In *Proceedings of the 19th KR*, Gabriele Kern-Isberner, Gerhard Lakemeyer, and Thomas Meyer (Eds.). 12–21.

[7] Natasha Alechina, Brian Logan, Hoang Nga Nguyen, and Franco Raimondi. 2017. Model-checking for Resource-Bounded ATL with production and consumption of resources. *J. Comput. System Sci.* 88 (2017), 126–144. https://doi.org/10.1016/j.jcss.2017.03.008

[8] Natasha Alechina, Brian Logan, and Giuseppe Perelli. 2025. Synthesising Minimum Cost Dynamic Norms. In *Proceedings of the 34th IJCAI*, James Kwok (Ed.). ijcai.org, 3–11. https://doi.org/10.24963/IJCAI.2025/1

[9] Natasha Alechina, Hans van Ditmarsch, Rustam Galimullin, and Tuo Wang. 2021. Verification and Strategy Synthesis for Coalition Announcement Logic. *Journal of Logic, Language and Information* 30, 4 (2021), 671–700. https://doi.org/10.1007/s10849-021-09339-6

[10] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. 2002. Alternating-time temporal logic. *J. ACM* 49 (2002), 672–713. https://doi.org/10.1145/585265.585270

[11] Benjamin Aminof, Marta Kwiatkowska, Bastien Maubert, Aniello Murano, and Sasha Rubin. 2019. Probabilistic Strategy Logic. In *Proceedings of the 28th IJCAI*, Sarit Kraus (Ed.). 32–38. https://doi.org/10.24963/IJCAI.2019/5

[12] Carlos Areces, Raul Fervari, and Guillaume Hoffmann. 2015. Relation-changing modal operators. *Logic Journal of the IGPL* 23, 4 (2015), 601–627. https://doi.org/10.1093/jigpal/jzv020

[13] Guillaume Aucher, Johan van Benthem, and Davide Grossi. 2018. Modal logics of sabotage revisited. *Journal of Logic and Computation* 28, 2 (2018), 269–303. https://doi.org/10.1093/logcom/exx034

[14] Philippe Balbiani, Alexandru Baltag, Hans van Ditmarsch, Andreas Herzig, Tomohiro Hoshi, and Tiago de Lima. 2008. 'Knowable' as 'known after an announcement'. *Review of Symbolic Logic* 1, 3 (2008), 305–334. https://doi.org/10.1017/S1755020308080210

[15] Alexandru Baltag and Lawrence S. Moss. 2004. Logics for Epistemic Programs. *Synthese* 139, 2 (2004), 165–224. https://doi.org/10.1023/B:SYNT.0000024912.56773.5e

[16] Alexandru Baltag, Lawrence S. Moss, and Sławomir Solecki. 1998. The Logic of Public Announcements, Common Knowledge, and Private Suspicions. In *Proceedings of the 7th TARK*, Itzhak Gilboa (Ed.). Morgan Kaufmann, 43–56.

[17] Alexandru Baltag and Sonja Smets. 2012. The dynamic turn in quantum logic. *Synthese* 186, 3 (2012), 753–773. https://doi.org/10.1007/S11229-011-9915-7

[18] Gregor Behnke, Malvin Gattinger, Avijeet Ghosh, and Haitian Wang. 2025. Comparing State-Representations for DEL Model Checking. In *Proceedings of the 20th TARK (EPTCS, Vol. 437)*, Adam Bjorndahl (Ed.). Open Publishing Association, 233–250. https://doi.org/10.4204/EPTCS.437.21

[19] Francesco Belardinelli, Rodica Condurache, Cătălin Dima, Wojciech Jamroga, and Michal Knapik. 2021. Bisimulations for verifying strategic abilities with an application to the ThreeBallot voting protocol. *Information and Computation* 276 (2021), 104552.

[20] Francesco Belardinelli, Wojciech Jamroga, Damian Kurpiewski, Vadim Malvone, and Aniello Murano. 2019. Strategy Logic with Simple Goals: Tractable Reasoning about Strategies. In *Proceedings of the 28th IJCAI*, Sarit Kraus (Ed.). 88–94. https://doi.org/10.24963/IJCAI.2019/13

[21] Pascal Bercher, Sarath Sreedharan, and Mauro Vallati. 2025. A Survey on Model Repair in AI Planning. In *Proceedings of the 34th IJCAI*, James Kwok (Ed.). ijcai.org, 10371–10380. https://doi.org/10.24963/IJCAI.2025/1152

[22] Raphaël Berthon, Bastien Maubert, Aniello Murano, Sasha Rubin, and Moshe Y. Vardi. 2021. Strategy Logic with Imperfect Information. *ACM Transactions on Computational Logic* 22, 1 (2021), 1–51. https://doi.org/10.1145/3427955

[23] Raven Beutner, Tzu-Han Hsu, Borzoo Bonakdarpour, and Bernd Finkbeiner. 2024. Syntax-Guided Automated Program Repair for Hyperproperties. In *Proceedings of the 36th CAV (LNCS, Vol. 14683)*, Arie Gurfinkel and Vijay Ganesh (Eds.). Springer, 3–26. https://doi.org/10.1007/978-3-031-65633-0_1

[24] Armin Biere, Alessandro Cimatti, Edmund M. Clarke, Ofer Strichman, and Yunshan Zhu. 2003. Bounded model checking. *Advances in Computers* 58 (2003), 117–148. https://doi.org/10.1016/S0065-2458(03)58003-2

[25] Borzoo Bonakdarpour and Bernd Finkbeiner. 2019. Program Repair for Hyperproperties. In *Proceedings of the 17th ATVA (LNCS, Vol. 11781)*, Yu-Fang Chen, Chih-Hong Cheng, and Javier Esparza (Eds.). Springer, 423–441. https://doi.org/10.1007/978-3-030-31784-3_25

[26] Stefano Borgo. 2007. Coalitions in Action Logic. In *Proceedings of the 20th IJCAI*, Manuela M. Veloso (Ed.). 1822–1827.

[27] Randal E. Bryant. 1986. Graph-Based Algorithms for Boolean Function Manipulation. *IEEE Trans. Comput.* 35, 8 (1986), 677–691. https://doi.org/10.1109/TC.1986.1676819

[28] Nils Bulling and Mehdi Dastani. 2016. Norm-based mechanism design. *Artificial Intelligence* 239 (2016), 97–142. https://doi.org/10.1016/j.artint.2016.07.001

[29] Davide Catta, Rustam Galimullin, and Munyque Mittelmann. 2026. On Angels and Demons: Strategic (De)Construction of Dynamic Models. *CoRR* abs/2601.07690 (2026). https://doi.org/10.48550/ARXIV.2601.07690

[30] Davide Catta, Rustam Galimullin, and Aniello Murano. 2025. First-Order Coalition Logic. In *Proceedings of the 34th IJCAI*, James Kwok (Ed.). ijcai.org, 4410–4418. https://doi.org/10.24963/IJCAI.2025/491

[31] Davide Catta, Jean Leneutre, and Vadim Malvone. 2023. Obstruction Logic: A Strategic Temporal Logic to Reason About Dynamic Game Models. In *Proceedings of the 26th ECAI (Frontiers in Artificial Intelligence and Applications, Vol. 372)*, Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein, and Roxana Radulescu (Eds.). IOS Press, 365–372. https://doi.org/10.3233/FAIA230292

[32] Davide Catta, Jean Leneutre, Vadim Malvone, and Aniello Murano. 2024. Obstruction Alternating-time Temporal Logic: A Strategic Logic to Reason about Dynamic Models. In *Proceedings of the 23rd AAMAS*, Mehdi Dastani, Jaime Simão Sichman, Natasha Alechina, and Virginia Dignum (Eds.). IFAAMAS, 271–280. https://doi.org/10.5555/3635637.3662875

[33] Davide Catta, Jean Leneutre, Vadim Malvone, and James Ortiz. 2025. Coalition Obstruction Temporal Logic: A New Obstruction Logic to Reason About Demon Coalitions. In *Proceedings of the 34th IJCAI*, James Kwok (Ed.). ijcai.org, 21–28. https://doi.org/10.24963/IJCAI.2025/3

[34] George Chatzieleftheriou, Borzoo Bonakdarpour, Panagiotis Katsaros, and Scott A. Smolka. 2015. Abstract Model Repair. *Logical Methods in Computer Science* 11, 3 (2015). https://doi.org/10.2168/LMCS-11(3:11)2015

[35] Taolue Chen and Jian Lu. 2007. Probabilistic Alternating-time Temporal Logic and Model Checking Algorithm. In *Proceedings of the 4th FSKD*, J. Lei (Ed.). IEEE Computer Society, 35–39. https://doi.org/10.1109/FSKD.2007.458

[36] Edmund M. Clarke, Orna Grumberg, and David E. Long. 1994. Model Checking and Abstraction. *ACM Transactions on Programming Languages and Systems* 16, 5 (1994), 1512–1542. https://doi.org/10.1145/186025.186051

[37] Edmund M. Clarke, William Klieber, Milos Novácek, and Paolo Zuliani. 2011. Model Checking and the State Explosion Problem. In *Tools for Practical Software Verification (LNCS, Vol. 7682)*, Bertrand Meyer and Martin Nordio (Eds.). Springer, 1–30. https://doi.org/10.1007/978-3-642-35746-6_1

[38] Tiago de Lima. 2014. Alternating-time temporal dynamic epistemic logic. *Journal of Logic and Computation* 24, 6 (2014), 1145–1178. https://doi.org/10.1093/LOGCOM/EXS061

[39] Lavindra de Silva, Paolo Felli, Jack C. Chaplin, Brian Logan, David Sanderson, and Svetan M. Ratchev. 2017. Synthesising Industry-Standard Manufacturing Process Controllers. In *Proceedings of the 16th AAMAS*, Kate Larson, Michael Winikoff, Sanmay Das, and Edmund H. Durfee (Eds.). ACM, 1811–1813.

[40] Jonathan Ezekiel, Alessio Lomuscio, Levente Molnar, and Sandor M. Veres. 2011. Verifying Fault Tolerance and Self-Diagnosability of an Autonomous Underwater Vehicle. In *Proceedings of the 22nd IJCAI*, Toby Walsh (Ed.). IJCAI/AAAI, 1659–1664. https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-279

[41] Peter Faymonville, Bernd Finkbeiner, and Leander Tentrup. 2017. BoSy: An Experimentation Framework for Bounded Synthesis. In *Proceedings of the 29th CAV (LNCS, Vol. 10427)*, Rupak Majumdar and Viktor Kuncak (Eds.). Springer, 325–332. https://doi.org/10.1007/978-3-319-63390-9_17

[42] Angelo Ferrando and Vadim Malvone. 2024. VITAMIN: A Compositional Framework for Model Checking of Multi-Agent Systems. *CoRR* abs/2403.02170 (2024). https://doi.org/10.48550/ARXIV.2403.02170

[43] Bernd Finkbeiner. 2016. Synthesis of Reactive Systems. In *Dependable Software Systems Engineering*, Javier Esparza, Orna Grumberg, and Salomon Sickert (Eds.). NATO Science for Peace and Security Series - D: Information and Communication Security, Vol. 45. IOS Press, 72–98. https://doi.org/10.3233/978-1-61499-627-9-72

[44] Pierre Gaillard, Fabio Patrizi, and Giuseppe Perelli. 2023. Strategy Repair in Reachability Games. In *Proceedings of the 26th ECAI (Frontiers in Artificial Intelligence and Applications, Vol. 372)*, Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein, and Roxana Radulescu (Eds.). IOS Press, 780–787. https://doi.org/10.3233/FAIA230344

[45] Rustam Galimullin and Thomas Ågotnes. 2021. Dynamic Coalition Logic: Granting and Revoking Dictatorial Powers. In *Proceedings of the 8th LORI*

(LNCS, Vol. 13039), Sujata Ghosh and Thomas Icard (Eds.). Springer, 88–101. https://doi.org/10.1007/978-3-030-88708-7_7

[46] Rustam Galimullin and Thomas Ågotnes. 2022. Action Models for Coalition Logic. In *Proceedings of the 4th DaLí (LNCS, Vol. 13780)*, Carlos Areces and Diana Costa (Eds.). Springer, 73–89. https://doi.org/10.1007/978-3-031-26622-5_5

[47] Rustam Galimullin, Maksim Gladyshev, Munyque Mittelmann, and Nima Motamed. 2025. Changing the Rules of the Game: Reasoning About Dynamic Phenomena in Multi-Agent Systems. In *Proceedings of the 24th AAMAS*, Sanmay Das, Ann Nowé, and Yevgeniy Vorobeychik (Eds.). IFAAMAS / ACM, 829–838. https://doi.org/10.5555/3709347.3743601

[48] Rustam Galimullin and Louwe B. Kuijer. 2023. Satisfiability of Arbitrary Public Announcement Logic with Common Knowledge is Σ^1_1-hard. In *Proceedings of the 19th TARK (EPTCS, Vol. 379)*, Rineke Verbrugge (Ed.). 260–271. https://doi.org/10.4204/EPTCS.379.21

[49] Rustam Galimullin and Louwe B. Kuijer. 2024. Synthesizing Social Laws with ATL Conditions. In *Proceedings of the 23rd AAMAS*, Mehdi Dastani, Jaime Simão Sichman, Natasha Alechina, and Virginia Dignum (Eds.). IFAAMAS / ACM, 2270–2272. https://doi.org/10.5555/3635637.3663130

[50] Rustam Galimullin and Fernando R. Velázquez-Quesada. 2025. Topic-Based Communication Between Agents. *Studia Logica* 113, 5 (2025), 1195–1243. https://doi.org/10.1007/S11225-024-10119-Z

[51] William A. Galston. 2020. Efficiency Isn't the Only Economic Virtue. *The Wall Street Journal* (10 March 2020). https://www.wsj.com/articles/efficiency-isnt-the-only-economic-virtue-11583873155 Opinion.

[52] Peter Gammie and Ron van der Meyden. 2004. MCK: Model Checking the Logic of Knowledge. In *Proceedings of the 16th CAV (LNCS, Vol. 3114)*, Rajeev Alur and Doron A. Peled (Eds.). Springer, 479–483. https://doi.org/10.1007/978-3-540-27813-9_41

[53] Malvin Gattinger, Stefano Volpe, Bas Laarakker, and Daniel Miedema. 2024. SMCDEL — A symbolic model checker for Dynamic Epistemic Logic. https://doi.org/10.5281/zenodo.11061438 Version 1.3.0.

[54] Nina Gierasimczuk, Vincent F. Hendricks, and Dick de Jongh. 2014. Logic and Learning. In *Johan van Benthem on Logic and Information Dynamics*, Alexandru Baltag and Sonja Smets (Eds.). Springer, 267–288. https://doi.org/10.1007/978-3-319-06025-5_10

[55] Patrice Godefroid, Michael Huth, and Radha Jagadeesan. 2001. Abstraction-Based Model Checking Using Modal Transition Systems. In *Proceedings of the 12th CONCUR (LNCS, Vol. 2154)*, Kim Guldstrand Larsen and Mogens Nielsen (Eds.). Springer, 426–440. https://doi.org/10.1007/3-540-44685-0_29

[56] Valentin Goranko. 2023. Logics for Strategic Reasoning of Socially Interacting Rational Agents: An Overview and Perspectives. *Logics* 1, 1 (2023), 4–35. https://doi.org/10.3390/logics1010003

[57] Valentin Goranko and Sebastian Enqvist. 2018. Socially Friendly and Group Protecting Coalition Logics. In *Proceedings of the 17th AAMAS*, Elisabeth André, Sven Koenig, Mehdi Dastani, and Gita Sukthankar (Eds.). IFAAMAS, 372–380.

[58] James Hales. 2013. Arbitrary Action Model Logic and Action Model Synthesis. In *Proceedings of the 28th LICS*. IEEE Computer Society, 253–262. https://doi.org/10.1109/LICS.2013.31

[59] Yifan He, Munyque Mittelmann, Aniello Murano, Abdallah Saffidine, and Michael Thielscher. 2025. Repairing General Game Descriptions. In *Proceedings of the 22nd KR*, Magdalena Ortiz, Renata Wassermann, and Torsten Schaub (Eds.). 386–395. https://doi.org/10.24963/kr.2025/38

[60] Andreas Herzig, Emiliano Lorini, Frédéric Moisan, and Nicolas Troquard. 2011. A dynamic logic of normative systems. In *Proceedings of the 22nd IJCAI*, Toby Walsh (Ed.). 228–233.

[61] Xiaowei Huang. 2015. Bounded model checking of strategy ability with perfect recall. *Artificial Intelligence* 222 (2015), 182–200. https://doi.org/10.1016/J.ARTINT.2015.01.005

[62] Xiaowei Huang, Kaile Su, and Chenyi Zhang. 2012. Probabilistic Alternating-Time Temporal Logic of Incomplete Information and Synchronous Perfect Recall. In *Proceedings of the 26th AAAI*, Jörg Hoffmann and Bart Selman (Eds.). AAAI Press, 765–771. https://doi.org/10.1609/AAAI.V26I1.8214

[63] Wojciech Jamroga. 2015. *Logical Methods for Specificaion and Verification of Multi-Agent Systems*. Institute of Computer Science, Polish Academy of Sciences.

[64] Wojciech Jamroga, Michal Knapik, and Damian Kurpiewski. 2018. Model Checking the SELENE E-Voting Protocol in Multi-agent Logics. In *Proceedings of the 3rd E-Vote-ID (LNCS, Vol. 11143)*, Robert Krimmer, Melanie Volkamer, Véronique Cortier, Rajeev Goré, Manik Hapsara, Uwe Serdült, and David Duenas-Cid (Eds.). Springer, 100–116. https://doi.org/10.1007/978-3-030-00419-4_7

[65] Wojciech Jamroga, Michal Knapik, Damian Kurpiewski, and Lukasz Mikulski. 2019. Approximate verification of strategic abilities under imperfect information. *Artificial Intelligence* 277 (2019). https://doi.org/10.1016/J.ARTINT.2019.103172

[66] Wojciech Jamroga and Aniello Murano. 2015. Module Checking of Strategic Ability. In *Proceedings of the 14th AAMAS*, Gerhard Weiss, Pinar Yolum, Rafael H. Bordini, and Edith Elkind (Eds.). ACM, 227–235.

[67] Wojciech Jamroga, Wojciech Penczek, Teofil Sidoruk, Piotr Dembinski, and Antoni W. Mazurkiewicz. 2020. Towards Partial Order Reductions for Strategic Ability. *Journal of Artificial Intelligence Research* 68 (2020), 817–850. https://doi.org/10.1613/JAIR.1.11936

[68] Maryam Kamali, Sven Linker, and Michael Fisher. 2018. Modular Verification of Vehicle Platooning with Respect to Decisions, Space and Time. In *Proceedings of the 6th FTSCS (Communications in Computer and Information Science, Vol. 1008)*, Cyrille Artho and Peter Csaba Ölveczky (Eds.). Springer, 18–36. https://doi.org/10.1007/978-3-030-12988-0_2

[69] Barteld Kooi. 2007. Expressivity and completeness for public update logics via reduction axioms. *Journal of Applied Non-Classical Logics* 17(2) (2007), 231–254.

[70] Barteld Kooi and Bryan Renne. 2011. Arrow Update Logic. *Review of Symbolic Logic* 4, 4 (2011), 536–559. https://doi.org/10.1017/S1755020311000189

[71] Barteld P. Kooi and Bryan Renne. 2011. Generalized arrow update logic. In *Proceedings of the 13th TARK-2011*, Krzysztof R. Apt (Ed.). ACM, 205–211. https://doi.org/10.1145/2000378.2000403

[72] Louwe B. Kuijer. 2017. Arbitrary arrow update logic with common knowledge is neither RE nor co-RE. In *Proceedings of the 16th TARK (EPTCS, Vol. 251)*, Jérôme Lang (Ed.). 373–381. https://doi.org/10.4204/EPTCS.251.27

[73] Orna Kupferman, Moshe Y Vardi, and Pierre Wolper. 2001. Module checking. *Information and Computation* 164, 2 (2001), 322–344.

[74] Damian Kurpiewski, Wojciech Jamroga, and Michal Knapik. 2019. STV: Model Checking for Strategies under Imperfect Information. In *Proceedings of the 18th AAMAS*, Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor (Eds.). IFAAMAS, 2372–2374.

[75] François Laroussinie, Nicolas Markey, and Ghassan Oreiby. 2008. On the Expressiveness and Complexity of ATL. *Logical Methods in Computer Science* 4, 2 (2008). https://doi.org/10.2168/LMCS-4(2:7)2008

[76] Claire Loiseaux, Susanne Graf, Joseph Sifakis, Ahmed Bouajjani, and Saddek Bensalem. 1995. Property Preserving Abstractions for the Verification of Concurrent Systems. *Formal Methods in System Design* 6, 1 (1995), 11–44. https://doi.org/10.1007/BF01384313

[77] Alessio Lomuscio, Wojciech Penczek, and Hongyang Qu. 2010. Partial Order Reductions for Model Checking Temporal-epistemic Logics over Interleaved Multi-agent Systems. *Fundamenta Informaticae* 101, 1-2 (2010), 71–90. https://doi.org/10.3233/FI-2010-276

[78] Alessio Lomuscio, Hongyang Qu, and Franco Raimondi. 2017. MCMAS: an open-source model checker for the verification of multi-agent systems. *International Journal on Software Tools for Technology Transfer* 19, 1 (2017), 9–30. https://doi.org/10.1007/s10009-015-0378-x

[79] Bastien Maubert, Sophie Pinchinat, and François Schwarzentruber. 2019. Reachability Games in Dynamic Epistemic Logic. In *Proceedings of the 28th IJCAI*, Sarit Kraus (Ed.). ijcai.org, 499–505. https://doi.org/10.24963/IJCAI.2019/71

[80] Bastien Maubert, Sophie Pinchinat, François Schwarzentruber, and Silvia Stranieri. 2020. Concurrent Games in Dynamic Epistemic Logic. In *Proceedings of the 29th IJCAI*, Christian Bessiere (Ed.). 1877–1883. https://doi.org/10.24963/IJCAI.2020/260

[81] Philipp J. Meyer, Salomon Sickert, and Michael Luttenberger. 2018. Strix: Explicit Reactive Synthesis Strikes Back!. In *Proceedings of the 30th CAV (LNCS, Vol. 10981)*, Hana Chockler and Georg Weissenbacher (Eds.). Springer, 578–586. https://doi.org/10.1007/978-3-319-96145-3_31

[82] Daniel Miedema and Malvin Gattinger. 2023. Exploiting Asymmetry in Logic Puzzles: Using ZDDs for Symbolic Model Checking Dynamic Epistemic Logic. In *Proceedings of the 19th TARK (EPTCS, Vol. 379)*, Rineke Verbrugge (Ed.). Open Publishing Association, 407–420. https://doi.org/10.4204/EPTCS.379.32

[83] Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y. Vardi. 2012. What Makes ATL* Decidable? A Decidable Fragment of Strategy Logic. In *Proceedings of the 23rd CONCUR (LNCS, Vol. 7454)*, Maciej Koutny and Irek Ulidowski (Eds.). Springer, 193–208. https://doi.org/10.1007/978-3-642-32940-1_15

[84] Fabio Mogavero, Aniello Murano, and Moshe Y. Vardi. 2010. Reasoning About Strategies. In *Proceedings of the 30th FSTTCS (LIPIcs, Vol. 8)*, Kamal Lodaya and Meena Mahajan (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 133–144. https://doi.org/10.4230/LIPIcs.FSTTCS.2010.133

[85] Nima Motamed, Natasha Alechina, Mehdi Dastani, Dragan Doder, and Brian Logan. 2023. Probabilistic Temporal Logic for Reasoning about Bounded Policies. In *Proceedings of the 32nd IJCAI*, Edith Elkind (Ed.). ijcai.org, 3296–3303. https://doi.org/10.24963/ijcai.2023/367

[86] Chunyan Mu, Nima Motamed, Natasha Alechina, and Brian Logan. 2025. Probabilistic Strategy Logic with Degrees of Observability. In *Proceedings of the 39th AAAI*, Toby Walsh, Julie Shah, and Zico Kolter (Eds.). AAAI Press, 15092–15099. https://doi.org/10.1609/AAAI.V39I14.33655

[87] Hoang Nga Nguyen, Natasha Alechina, Brian Logan, and Abdur Rakib. 2018. Alternating-time temporal logic with resource bounds. *Journal of Logic and Computation* 28, 4 (2018), 631–663. https://doi.org/10.1093/logcom/exv034

[88] Marc Pauly. 2001. *Logic for Social Software*. Ph.D. Dissertation. ILLC, University of Amsterdam, The Netherlands.

[89] Marc Pauly. 2002. A modal logic for coalitional power in games. *Journal of Logic and Computation* 12, 1 (2002), 149–166. https://doi.org/10.1093/logcom/12.1.149

[90] Jan Plaza. 2007. Logics of public communications. *Synthese* 158, 2 (2007), 165–179. https://doi.org/10.1007/s11229-007-9168-7

[91] Florian Renkin, Philipp Schlehuber-Caissier, Alexandre Duret-Lutz, and Adrien Pommellet. 2022. Dissecting ltlsynt. *Formal Methods System Design* 61, 2 (2022), 248–289. https://doi.org/10.1007/S10703-022-00407-6

[92] Pierre-Yves Schobbens. 2004. Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science* 85, 2 (2004), 82–93. https://doi.org/10.1016/s1571-0661(05)82604-0

[93] Yoav Shoham and Moshe Tennenholtz. 1995. On Social Laws for Artificial Agent Societies: Off-Line Design. *Artificial Intelligence* 73, 1-2 (1995), 231–252. https://doi.org/10.1016/0004-3702(94)00007-N

[94] Johan van Benthem. 1996. *Exploring logical dynamics.* CSLI.

[95] Johan van Benthem. 2003. Logic and the Dynamics of Information. *Minds and Machines* 13, 4 (2003), 503–519. https://doi.org/10.1023/A:1026296600182

[96] Johan van Benthem, Krzysztof Mierzewski, and Francesca Zaffora Blando. 2020. The Modal Logic of Stepwise Removal. *The Review of Symbolic Logic* 15, 1 (2020), 36–63. https://doi.org/10.1017/s1755020320000258

[97] Johan van Benthem, Jan van Eijck, Malvin Gattinger, and Kaile Su. 2018. Symbolic model checking for Dynamic Epistemic Logic - S5 and beyond. *Journal of Logic and Computation* 28, 2 (2018), 367–402. https://doi.org/10.1093/LOGCOM/EXX038

[98] Wiebe van der Hoek, Mark Roberts, and Michael J. Wooldridge. 2007. Social laws in alternating time: effectiveness, feasibility, and synthesis. *Synthese* 156, 1 (2007), 1–19. https://doi.org/10.1007/S11229-006-9072-6

[99] Ron van der Meyden. 2018. On the specification and verification of atomic swap smart contracts. *CoRR* abs/1811.06099 (2018). http://arxiv.org/abs/1811.06099

[100] Ron van der Meyden. 2019. On the specification and verification of atomic swap smart contracts (extended abstract). In *Proceedings of the 1st ICBC*. IEEE, 176–179. https://doi.org/10.1109/BLOC.2019.8751250

[101] Hans van Ditmarsch. 2023. To be announced. *Information and Computation* 292 (2023), 1–42. https://doi.org/10.1016/j.ic.2023.105026

[102] Hans van Ditmarsch, Tim French, Rustam Galimullin, and Louwe B. Kuijer. 2025. Modal Logic for Simulation, Refinement, and Mutual Ignorance. In *Proceedings of the 20th TARK (EPTCS, Vol. 437)*, Adam Bjorndahl (Ed.). Open Publishing Association, 379–398. https://doi.org/10.4204/EPTCS.437.30

[103] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. 2008. *Dynamic epistemic logic*. Synthese Library, Vol. 337. Springer.

[104] Hans van Ditmarsch, Wiebe van der Hoek, Barteld Kooi, and Louwe B. Kuijer. 2017. Arbitrary arrow update logic. *Artificial Intelligence* 242 (2017), 80–106. https://doi.org/10.1016/j.artint.2016.10.003

[105] Hans van Ditmarsch, Wiebe van der Hoek, Barteld Kooi, and Louwe B. Kuijer. 2020. Arrow update synthesis. *Information and Computation* 275 (2020). https://doi.org/10.1016/j.ic.2020.104544

[106] Moshe Y. Vardi. 2020. Efficiency vs. resilience: what COVID-19 teaches computing. *Commun. ACM* 63, 5 (2020), 9. https://doi.org/10.1145/3388890

[107] Dirk Walther, Wiebe van der Hoek, and Michael J. Wooldridge. 2007. Alternating-time temporal logic with explicit strategies. In *Proceedings of the 11th TARK-2007*, Dov Samet (Ed.). 269–278. https://doi.org/10.1145/1324249.1324285