CTL.STIT: enhancing ATL to express important multi-agent system verification properties

Jan Broersen
Department of Information and Computing Sciences
Utrecht University
The Netherlands
broersen@cs.uu.nl

ABSTRACT

We present the logic CTL.STIT, which is the join of the logic CTL with a multi-agent strategic *stit*-logic variant. CTL.STIT subsumes ATL, and adds expressivity to it that we claim is very important for using the formalism for multi-agent system verification. We argue extensively that the extra expressivity is important for extending ATL to a language for reasoning about norms, strategic games, knowledge games, conditional strategies, etc. Also we compare the logic's suitability for multi-agent system verification with verification formalisms based on dynamic logic. We will give a number of arguments in favor of *stit*-formalisms. But, which paradigm to use ultimately depends on what kind of properties we want to verify.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Modal logic*

General Terms

Theory, Verification

Keywords

Logics for agency, Verification of multiagent systems, stit theory, Reasoning about strategies

1. INTRODUCTION

The logic ATL [4] and its fragment CL [29] have gained quite some popularity among logicians working on formal logical models for knowledge representation and verification in multi-agent systems [20, 25, 3]. However, the kind of reasoning about multi-agent *interaction* possible in ATL is rather limited. The main interaction modeled by ATL is the one that can be described by the slogan "two can do more than one". Axiomatically, this reasoning is represented by ATL's central super-additivity axiom. In words, the axiom says: If A can do φ and B can do ψ, together A and B can do $\varphi \wedge \psi$. Of course, ATL also has temporal expressivity, borrowed from CTL [16]. For instance, in the just explained interaction, φ and ψ can be temporal formulas, like that a certain condition will hold next, at some point in the future (liveness) or henceforth (safety). This temporal expressivity can be enhanced, for instance, to get ATL*, or an even stronger fixed-point (mu-calculus) variant. But the role of temporal expressivity for system verification is known since the 90ies. This paper is concerned with the *interactions* between agents and the properties of interaction we want to verify.

Cite as: CTL.STIT: enhancing ATL to express important multi-agent system verification properties, Jan Broersen, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 683-690

In the context of MAS-verification, the reasoning represented by ATL's super-additivity axiom is relevant for checking reachability (liveness) properties: is a certain coalition capable of achieving a certain goal? Do the agents in the coalition need each other to achieve the goal, or can they already achieve it individually or as a sub-group? The same questions can be asked for safety properties: can groups guarantee some property is preserved over time? Do they need each other to preserve the property? That is, essentially, the kind of reasoning possible in ATL, and indeed, this kind of reasoning is obviously relevant for multi-agent system verification. However, there are much more interaction properties we might want to verify of MAS-systems. In section 2 we will give several of them.

Our answer to the aim to extend the expressiveness of ATL in a way beneficial for verification, comes in the form of a strategic *stit*-logic that addapts, corrects and simplifies earlier work [12, 11]. For those unfamiliar with the *stit*-framework: the characters 'stit' are an acronym for 'seeing to it that'. *stit*-logics [6, 7] originate in philosophy, and can be described as endogenous logics of agency, that is, logics of agentive action where actions are not made explicit in the object language. To be more precise, expressions $[A\ stit : \varphi]$ of *stit*-logic stand for 'agents A see to it that φ', where φ is a (possibly) temporal formula. However, where philosophers write '$[A\ stit : \varphi]$', we prefer to write '$[A\ \mathsf{stit}]\varphi$' to denote the same notion, to be more in line with standard modal notation. The main virtue of *stit*-logics is that, unlike most (if not all) other logical formalisms, they can express that a choice or action is actually performed / taken / executed by an agent. In particular, in the logic of the present paper we can express that a *strategy* is executed. Within the philosophical community working on the *stit*-framework of Belnap [6, 7] and Horty [26], it was an open problem how to define a suitable notion of *strategic stit*. The strategic *stit*-operator we propose here, is a possible answer. This extends the expressiveness of logics like ATL in an essential way, since ATL can only talk about strategic *abilities*. One of the things our semantics shows, is how we can make the implicit quantifications in the semantics of the ATL operators explicit in the object language: the central ATL operators will each be decomposed into three individual modal quantifiers.

First, in section 2 we explain why ATL lacks important expressive power for functioning as a verification language. Then, in section 3 we give the semantics of our logic, that combines the temporal operators of CTL with strategic agency (*stit*) operators. Section 4 then compares the logic with the logic ATL. In section 5 we discuss the next-time fragment of ATL and CTL.STIT by making a comparison with the logic XSTIT that assumes a fundamentally different semantics. Section 6 discusses possible variations of the logic. Then in section 7 we give some arguments for using *stit* rather than dynamic logic for a logic aimed at expressing properties for multi-agent system verification. Section 8 concludes.

2. MOTIVATIONS FOR ENHANCING ATL

The first thing to mention here is assumption guarantee reasoning [1]. Assumption guarantee reasoning was developed in the context of formal system verification to take advantage of the fact that systems are often described as build from separate sub-systems. The idea is that instead of proving the correctness of a system as a whole, we may switch to proving the behavioral correctness of sub-systems, *modulo* assumptions of correct behavior of other components. Notably, this kind of reasoning[1] *is* implemented in the well-known ATL model checker Mocha [5]. In Mocha one can check an ATL formula on a model defined in terms of a language called 'reactive modules'. In this reactive modules language one can encode that a component (module) behaves in a certain way, thus enabling model checking under this "guarantee assumption" [23]. However, model checking is not the same as reasoning. The reactive modules language is not part of the ATL language, so Mocha does not enable reasoning modulo the behavior of other components (agents) in one single logical object language. For that ATL has to be extended. In stead of encoding fixed behavior of agents in models by means of the reactive modules language, we need to be able to express that an agent is acting in a certain way directly in the logical object language. In particular we need expressivity of the form 'coalition C *is currently performing* a strategy ensuring φ'. If we can express that in an extension of ATL, we can express assumption-guarantee properties using an ordinary material conditional. We could express, for instance, verification properties of the form "if agents 1 ensures the communication channel stays up and running (a safety property), then agents 2 and 3 *can* negotiate to reach an agreement (a liveness property)". It is clear the 'if' part of this conditional cannot be expressed in ATL, since ATL can only talk about abilities (as in the 'then' part of the conditional).

This brings us to the second argument for having to endow ATL with the expressive capacity to talk about performance of action. An interaction property following from super-additivity is 'regularity'. Regularity is a strong property. It says that an agent can do something only if it can ensure an outcome irrespective of what the others do (called 'β-effectivity' in game theory [28]). Then, strictly speaking, a skill like opening a door is not something an agent can do in general. There can be many circumstances that prevent an agent from opening a door. Somebody else, on the other side, keeps it closed. Or it is locked. On grounds of examples like these, one can argue that regularity is much too strong for reasoning about abilities. If we agree the central operators of ATL express group ability (and the consensus is they do), than this is ability of a very idealized kind. The underlying problem here is well-known in the area of reasoning about action and change. It is the *qualification* problem [19]. The qualification problem is the problem of how to deal, in action descriptions, with the fact that it is very hard to foresee *all* the necessary conditions for execution of an action and give a sufficient condition. And very often these necessary conditions concern the question whether or not other agents perform certain acts concurrently. In the example of the door, ability to open it depends on the condition whether or not another agent keeps it closed. Or in the example of agents 2 and 3 reaching an agreement, it is the condition of whether or not agent 1 keeps up the communication channel. So, what one would like to add to the analysis of ability as represented by ATL, is that in general there are many *conditions* under which ability is assumed. And these conditions are often that other agents behave in a 'normal' way. This leads to exactly the same requirement as for the argument based on assumption guarantee reasoning (we might even say that they are the same argument, although their origins are quite different): we need an operator to conditionalize on what strategies other agents currently perform.

Before giving more arguments for wanting to be able to express that agents and groups of agents currently perform certain strategies, we want to point to other work on extensions of ATL that has a related motivation. It has been recognized by several authors that it would be good to define extensions of ATL where strategies are not only implicit in the semantics, but are brought more to the foreground. For instance, in CATL [33] and related formalism [34] strategies appear as functions σ in the object language. Van Benthem has called strategies 'the unsung heroes of game theory' [8], referring to the fact that the operators in languages like ATL only quantify over strategies, and not give them as object language level constructs. Van Benthem [32], proposes to use dynamic logic for that. So, these works share an important characteristic with the work in this paper: the objective to get rid of the quantifications over strategies that are hard coded in the semantics of the modal operators of ATL. However, the present approach does not use dynamic logic, but *stit*-logic to accomplish this. This is more natural, since the ATL view on action which is based on 'effectivity' is the same as in *stit*-approaches. In section 7 we will come back to the point of using *stit* instead of dynamic logic. A second difference with the work in this paper is that conditionalization on strategy execution (of other agents) is modeled by different means. The strategic *stit*-operators we use in this paper enable us to conditionalize on strategy execution simply by using the standard material implication, which, from a logic perspective, has many advantages. In a system like CATL, conditionalization is implicit: it is with respect to the strategy functions inside modal operators: no attempt is made to define the conditionalization in terms of stand alone modalities in combination with the material implication.

A third motivation for the extension of ATL we propose originates in deontic reasoning. We might say that deontic logic is about reasoning which strategies are 'good' and which are 'bad' from a normative perspective. However, again, this often depends on dynamic aspects of the worlds, in particular, on what agents or others are doing. First, normative modalities are often conditional on what agents are doing. For instance If you drive your car, you have to carry your license. The condition is here the execution / being in process of some action: you driving your car (note that the condition is *not* the static one of simply being inside a car; for being insider a car, but not driving, you do not need a license). Then, to reason about this conditional obligation, we need to be able to represent this dynamic condition. The logic we present here enables us to simply represent this kind of conditional obligations using Anderson's reduction: if an agent drives a car and does not carry a licence, he brings about a violation. Other examples from deontic logic that present a clear motivation concern contrary to duty (CTD) norms. Consider for example the famous[2] dynamic version of the Chisholm paradox [13]: the gentle murderer [17]. The challenge here is to model the CTD sentences: "It is forbidden to kill,

[1] Where Abadi and Lamport, in their '95 paper spoke about 'assumption-guarantee' reasoning, the researchers developing Mocha speak of 'assume-guarantee' reasoning.

[2] That is, among deontic logicians.

but if you kill, you have to kill gently". The condition in the CTD norm is obviously dynamic: it depends on whether or not the agent performs the act of killing.

Our fourth motivation is a ramification of the objective to find logics modeling the reasoning underlying solution concepts from game theory [28]. For modeling this kind of reasoning, again, we need to reason modulo the moves of other players in the game (as is also observed in [33]). For instance, for a move to be strictly dominating (a solution concept similar to Savage's 'sure thing' principle from decision theory [31]), it must be the best move whatever the other agents do. Or for a move to be Nash, it must be a best response to a best response of the opponents. And, finally, the reasoning behind iterated elimination of dominated strategies concerns a nested series of conditional assumptions about other agents' moves.

3. A SEMANTIC CHARACTERIZATION OF THE LOGIC

In this section we present the formal syntax and semantics of CTL.STIT. The acronym CTL.STIT refers to the fact that the logic combines the temporal expressivity of CTL with a *stit*-logic. The *stit*-logic is what Belnap and Horty call a 'strategic' *stit*-logic. This concerns the fact that choices are not viewed as one-shot actions, but as extensive form plans possibly involving series of subsequent choices.

In the syntax we have a operator $\Box\varphi$ for 'historical necessity' (inevitability/settledness) of φ, an operator $[A\ \mathsf{sstit}]X\varphi$ for 'group A strategically ensures that next φ, an operator $[A\ \mathsf{sstit}]G\varphi$ for 'group A strategically ensures that henceforth φ, and an operator $[A\ \mathsf{sstit}](\psi U \varphi)$ for 'group A strategically ensures that at some future point φ, while ψ holds until then. The acronym sstit comes from 'strategically seeing to it'. We can view the syntax of CTL.STIT as follows. The syntax of CTL.STIT builds on the syntax of CTL by replacing the CTL's path quantifiers A and E by the semantically very similar historical necessity modality $\Box$ and its dual $\Diamond$, and by prefixing every temporal operator by a strategic *stit*-operator.

Definition 3.1 (Syntax). *Well-formed formulas of the language $\mathcal{L}_{CTL.STIT}$ are defined by:*

$$\begin{aligned}\varphi \quad := \quad & p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box\varphi \mid \\ & [A\ \mathit{sstit}]X\varphi \mid [A\ \mathit{sstit}]G\varphi \mid [A\ \mathit{sstit}](\varphi U \varphi)\end{aligned}$$

The p are elements from a countable infinite set of propositional symbols $\mathcal{P}$, and A is a subset of a finite set of agent names Ags. We use natural numbers as agent names. We use the notation $\overline{A} \equiv_{def} Ags \setminus A$ to refer to complementary agent sets. We use $\varphi, \psi, \ldots$ to represent arbitrary well-formed formulas. We use the standard propositional abbreviations, the standard notation for the duals of modal boxes (that is, diamonds) and the following:

Definition 3.2 (Syntactic abbreviation).

$$[A\ \mathit{sstit}]F\varphi \equiv_{def} [A\ \mathit{sstit}](\top U \varphi)$$

We now go on to define the semantic structures for CTL.STIT. The main new issue to consider in defining a semantics, is the interpretation of the strategic *stit*-modalities. For this we need to define what we mean by a strategy. In the literature, strategies are most often defined as a mapping from states to choices in states. The choices can have names (multi-player game models), or not (alternating transition systems). Here we take an equivalent, though slightly different viewpoint: strategies *are* sets of system histories. In semantics for ATL [20] based on game structures (forms) where choices have names, one has to define an outcome function mapping action names to outcomes (histories). This outcome function is then said to define whether or not a history 'complies' with a strategy expressed in terms of named actions to be executed in system states. But, since the names of actions play no role whatsoever in the object language and axiomatizations of logics like ATL, we prefer to define strategies directly as sets of histories instead of referring to them indirectly by using names that play no role in the logic (this would be different, of course, if we would use a dynamic logic to talk about the structures). Compliance of a history with a strategy can then simply be expressed with the membership relation.

A second feature of the semantics that needs close explanation are the units of evaluation, or 'worlds' as they are called in possible world semantics. The units of evaluation determine the basic elements with respect to which we want to assess the truth of formulas. We explained in section 2 that we aim to have a logic that enables us (1) to conditionalize on dynamic aspects like other agents executing a certain action or strategy, and (2) to use standard material implication for expressing this conditionalization. We can only achieve this by introducing dynamic elements as units of evaluation. 'Truth' is then a property of possible answers to the question whether or not certain actions or strategies (by certain agents) are (in the process of) being executed, or not. So we introduce 'dynamic states' as the units of evaluation. The term 'dynamic state' is an oxymoron aiming to point to the dynamic aspect of an agent's state. We can perfectly well see the action we are executing, strategy we are taking, or program we are running as a state we are in. The dynamic states of our semantics take a static system state, a history and a list of strategies (one for each agent in the system) as components. So, the formulas of CTL.STIT are evaluated against tuples $\langle s, h, \alpha_1, \alpha_2, \ldots \alpha_n \rangle$, where s is a static system state, h a history, and $\alpha_1, \alpha_2, \ldots \alpha_n$ a strategy profile. Then, the truth of formulas is evaluated against the background of a *current* state, a *current* history, and a *current* strategy-profile. If, under this semantics, we want to consider more classical truths that do *not* depend on dynamic aspects like histories and strategies, we can use the historical necessity operator $\Box$. In particular, if $\Box\varphi$ holds, φ can be said to hold 'statically'. In *stit*-theory, one would say that φ is 'moment determinate'. We also says that φ is 'settled'[3], which refers to the fact that it is completely independent of any action currently taken by any agent in the system.

We now first give the definition of the modal frames. Then afterwards, we explain the different items of the definition using the frame visualization in figure 1.

Definition 3.3 (Semantic frames). *A frame is a tuple $\mathcal{F} = \langle S, H, \{sT(x) \mid x \in Ags\}, R_X, \{R_A \mid A \subseteq Ags\}\rangle$ such that:*

1. *S is a non-empty set of static multi-agent system states. Elements of S are denoted s, s', etc.*

2. *H is a non-empty set of possible system histories of the form $\ldots s_{-2}, s_{-1}, s_0, s_1, s_2, \ldots$ with $s_x \in S$ for $x \in \mathbb{Z}$. Elements of H are denoted h, h', etc. For s, t appearing on h we write $s <_h t$ in case s appears strictly before t on the history h.*

[3]Settledness does *not* necessarily mean that a property is *always* true in the future (as often thought). Settledness may, for instance, apply to the condition that φ occurs 'some' time in the future, or to some other temporal property. So, settledness is a universal quantification over the *branching* dimension of time, and *not* over the dimension of duration.

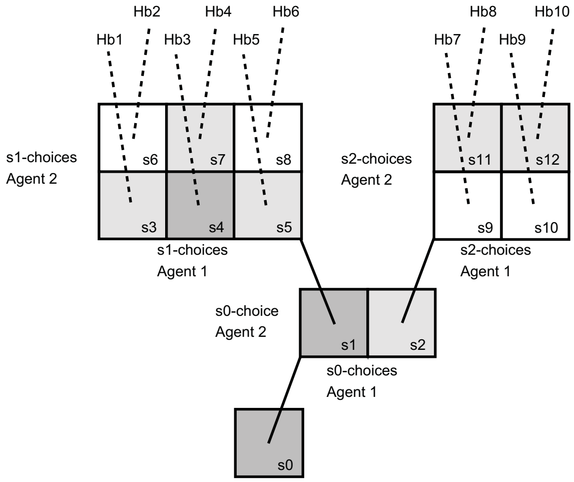

Figure 1: Visualization of a strategy profile in a partial two agent CTL.STIT frame

3. *$sT(x)$ yields for each $x \in Ags$ a non-empty set of strategies. Strategies are non-empty sets of system histories. For agent 1, the strategies $sT(1)$ are denoted α_1, β_1, etc. A strategy profile*[4] *relative to $sT(x)$ is a list of strategies $\alpha_1, \alpha_2, \ldots \alpha_n$, where $\{1, 2, \ldots, n\} = Ags$ and $\alpha_x \in sT(x)$ for any x. For strategy profiles we will use the vector notation '$\vec{\alpha}$' when we need to be more concise. For any strategy profile $\alpha_1, \alpha_2, \ldots \alpha_n$ relative to $sT(x)$ and any $s \in S$:*

   (a) *there is an $h \in H$ with s on h and $\forall x \in Ags,\ h \in \alpha_x$*

   (b) *if s on h and $\forall x \in Ags,\ h \in \alpha_x$ and if s on h' and $\forall x \in Ags,\ h' \in \alpha_x$ then $h = h'$*

4. *Dynamic states are tuples $\langle s, h, \alpha_1, \alpha_2, \ldots \alpha_n \rangle$, where:*

   (a) *$s \in S$, $h \in H$, and $\alpha_1, \alpha_2, \ldots \alpha_n$ is a strategy profile relative to $sT(x)$*

   (b) *s appears on h*

   (c) *$\forall x \in Ags,\ h \in \alpha_x$*

5. *R_X is a 'next-state' relation over dynamic states. That is, $\langle s, h, \alpha_1, \alpha_2, \ldots \alpha_n \rangle R_X \langle s', h', \beta_a, \beta_b, \ldots \beta_k \rangle$ only if $h = h'$, $\forall x \in Ags, \alpha_x = \beta_x$, and s' is the successor of s on the history h.*

6. *The R_A are 'effectivity' equivalence classes over dynamic states such that $\langle s, h, \alpha_1, \alpha_2, \ldots \alpha_n \rangle R_A \langle s', h', \beta_a, \beta_b, \ldots \beta_k \rangle$ if and only if $s = s'$, and $\forall x \in A, \alpha_x = \beta_x$.*

7. *$R_X \circ R_\emptyset \subseteq R_\emptyset \circ R_X$*

Items **1**, **2**, **3** and **4** define the structure of the units of evaluation: the dynamic states. System histories are ordered sets of static system states, strategies are sets of system histories and a strategy profile is a choice of strategy for each agent in the system. Figure 1 visualizes a (partial) frame and a strategy profile in it, from the viewpoint of state $s0$. Colons that are grey represent profile choices of agent 1, rows that are grey represent profile choices of agent 2. No $s2$-profile choice is depicted for agent 1, because we only look at the profile from the perspective of $s0$, and in this profile, for agent 1, $s2$ is not a reachable state from $s0$.

Condition **3(a)** ensures that intersections of strategies of different agents, as seen from a particular state s, are never empty. This implements the *stit*-requirement of *independence of agency* (no agent can choose a strategy that instantaneously affects which strategies are in the strategy repertoire of other agents).

Conditions **3(a)** and **3(b)** ensure that through any static state there is *exactly one* history complying to all strategies of a strategy profile. This reflects the idea, also assumed in ATL and CL, that a choice of strategy for each agent in the system determines the next static state. This means that, strictly speaking, we do not have to introduce h here as an independent element of the units of evaluation, since we can define it as the intersection of the strategies in the profile. The reason that we do this anyway is that in section 6 we want to discuss the possibility to drop condition **3(b)** and allow for non-determinism introduced by the agents' environment. In figure 1 the bundle of histories $Hb3$ through the darker grey little squares is the bundle of histories containing the unique $s0$-history determined by the intersection of the strategies of agents 1 and 2 in the profile.

Item **4** defines the basic units of evaluation. Note that the difference with classical multi-agent or group *stit*-models [7] is that states are not partitioned by one shot actions, but by strategies. This generalization is the essential step for defining our *strategic* version of the *stit*-operator.

Item **5** defines the relation R_X to be a 'next-state' relation over dynamic states. Note that system states may occur more than ones on a system history. A system might even stay in the same state forever. So, a system state should not be confused with a 'moment'. Only if we consider the occurrence of a system state at some point on the history, we may think of this occurrence as a moment.

Item **6** says that R_A reaches all dynamic states that only deviate from the current dynamic state in the sense that agents not among A perform a choice different from the current one. This reflects the basic idea of alternating time temporal logic (ATL), saying that acting or choosing is ensuring a condition irrespective of what other agents do or choose.

Condition **7** enforces that the dynamic states based on the next static state originate from a subset of the dynamic states based on the current static state. In figure 1, when going a step forward from $s0$, for instance with the grey profile as the current profile, the alternative profiles we can consider from $s1$ are a subset of the profiles we could consider as alternatives in $s0$. This is because in $s0$ agent 1 still has an alternative leading to $s2$ that he has lost once arrived at $s1$.

Now we are ready to define the formal semantics of the language $\mathcal{L}_{\mathsf{CTL.STIT}}$. The semantics is multi-dimensional, and the truth conditions are quite standard. First we define models based on the frames of the previous definition.

Definition 3.4 (Models). *A frame $\mathcal{F} = \langle S, H, \{sT(a) \mid a \in Ags\}, R_X, \{R_A \mid A \subseteq Ags\}\rangle$ is extended to a model $\mathcal{M} = \langle S, H, \{sT(a) \mid a \in Ags\}, R_X, \{R_A \mid A \subseteq Ags\}, \pi\rangle$ by adding a valuation π of atomic propositions:*

- *π is a valuation function $\pi : P \longrightarrow 2^{S \times H \times sT(a)}$ assigning to each atomic proposition the set of dynamic states in which they are true.*

Note that truth assignments to propositional atoms may be different for different dynamic states based on the same static system state. This raises questions. We explained that we want truth to be relative to dynamic features of the world to enable conditionalization using the standard material implication. This is perfectly all

[4]In the game forms of game theory strategy profiles are refereed to by means of names associated with the choices of agents in system states. Here we abstract from names of choices, as explained.

right for formulas expressing dynamic features of the system, that is for all formulas containing modalities. However, for formulas containing no modalities at all, i.e., the formulas expressing static features of the system in the current system state, we might defend a different opinion. For such formulas, let's denote them by meta-variables π, one would expect that they have the *same* truth evaluation for any dynamic state based on the same static state. We could accomplish this by giving an alternative definition of models relative to the frames of definition 3.3. In the definition of models we can impose the constraint that all dynamic states sharing the same static states have identical assignments of truth values to propositional atoms. This would then result in a validity $\pi \rightarrow \Box\pi$, but, only for the *modality-free* formulas π. So, this would lead us to a setting that is slightly non-standard from a modal logic perspective. Although we think it is perfectly possible to do this, we prefer not to pursue this variation on the semantics here, because we do not think it adds an interesting or essential element to the reasoning we aim to capture with the logic. Furthermore, the property $\pi \rightarrow \Box\pi$ would only be based on the unwarranted idea that histories through a static system state actually all somehow 'share' this state. There is no need to assume that. For evaluating static properties, we simply only look at the present static system state compatible with the actual dynamic state, and for evaluating dynamic properties we consider all dynamic states reachable from the current dynamic state.

Definition 3.5 (Truth, validity, logic). *Truth* $\mathcal{M}, \langle s, h, \vec{\alpha}\rangle \models \varphi$, *of a* CTL.STIT*-formula* φ *in a dynamic state* $\langle s, h, \vec{\alpha}\rangle$ *of a model* $\mathcal{M} = \langle S, H, \{sT(a) \mid a \in Ags\}, R_X, \{R_A \mid A \subseteq Ags\}, \pi\rangle$ *is defined as (suppressing the model denotation '*$\mathcal{M}$*'):*

$\langle s, h, \vec{\alpha}\rangle \models p$	$\Leftrightarrow$	$\langle s, h, \vec{\alpha}\rangle \in \pi(p)$
$\langle s, h, \vec{\alpha}\rangle \models \neg\varphi$	$\Leftrightarrow$	*not* $\langle s, h, \vec{\alpha}\rangle \models \varphi$
$\langle s, h, \vec{\alpha}\rangle \models \varphi \wedge \psi$	$\Leftrightarrow$	$\langle s, h, \vec{\alpha}\rangle \models \varphi$ *and* $\langle s, h, \vec{\alpha}\rangle \models \psi$
$\langle s, h, \vec{\alpha}\rangle \models \Box\varphi$	$\Leftrightarrow$	*if* $\langle s, h, \vec{\alpha}\rangle R_\emptyset \langle s, h', \vec{\beta}\rangle$ *then* $\langle s, h', \vec{\beta}\rangle \models \varphi$
$\langle s, h, \vec{\alpha}\rangle \models [A\ \mathsf{sstit}]X\varphi$	$\Leftrightarrow$	*for all* $h', \vec{\beta}$ *such that* $\langle s, h, \vec{\alpha}\rangle R_A \langle s, h', \vec{\beta}\rangle$ *and for all* s' *such that* $\langle s, h', \vec{\beta}\rangle R_X \langle s', h', \vec{\beta}\rangle$ *it holds that* $\langle s', h', \vec{\beta}\rangle \models \varphi$
$\langle s, h, \vec{\alpha}\rangle \models [A\ \mathsf{sstit}]G\varphi$	$\Leftrightarrow$	*for all* $h', \vec{\beta}$ *such that* $\langle s, h, \vec{\alpha}\rangle R_A \langle s, h', \vec{\beta}\rangle$ *and for all* s' *such that* $s \leq_{h'} s'$ *it holds that* $\langle s', h', \vec{\beta}\rangle \models \varphi$
$\langle s, h, \vec{\alpha}\rangle \models [A\ \mathsf{sstit}](\psi U \varphi)$	$\Leftrightarrow$	*for all* $h', \vec{\beta}$ *such that* $\langle s, h, \vec{\alpha}\rangle R_A \langle s, h', \vec{\beta}\rangle$ *it holds that* $\exists t$ *on* h' *with* $s \leq_{h'} t$ *such that* (1) $\langle t, h', \vec{\beta}\rangle \models \varphi$ *and* (2) $\forall r$ *with* $s \leq_{h'} r <_{h'} t$ *we have* $\langle r, h', \vec{\beta}\rangle \models \psi$

Validity on a CTL.STIT*-model* $\mathcal{M}$ *is defined as truth in all dynamic states of the* CTL.STIT*-model. General validity of a formula* φ *is defined as validity on all possible* CTL.STIT*-models. The logic* CTL.STIT *is the subset of all general validities of* $\mathcal{L}_{\mathsf{CTL.STIT}}$ *over the class of* CTL.STIT*-models.*

4. COMPARING CTL.STIT AND ATL

In this section we show that CTL.STIT embeds ATL [4], and thus also CTL [16] and CL [29]. Also we mention validities for CTL.STIT that do not hold for ATL.

Definition 4.1 (Mapping ATL to CTL.STIT). *We define a mapping from* ATL *modalities to* CTL.STIT *modalities according to:*
$\langle\langle A\rangle\rangle X\varphi \equiv_{def} \Diamond[A\ \mathsf{sstit}]X\varphi$,
$\langle\langle A\rangle\rangle G\varphi \equiv_{def} \Diamond[A\ \mathsf{sstit}]G\varphi$,
$\langle\langle A\rangle\rangle(\varphi U \psi) \equiv_{def} \Diamond[A\ \mathsf{sstit}](\varphi U \psi)$.
Modality-free formula parts of ATL *formulas are mapped to identical modality-free formula parts of* CTL.STIT *formulas.*

Theorem 4.1. *The mapping of definition 4.1 embeds the logic* ATL *in the logic* CTL.STIT*.*

We will now discuss the proof of this theorem. We do not give the details of the proof, because, among other things, we would need to present the more classical semantics of ATL, which takes up too much space for the purposes of this paper. What we need to prove is that the mapping of definition 4.1 preserves the logic in both directions. Now, showing that the mapping preserves validities in one direction is equivalent with showing that the mapping preserves satisfiability in the opposite direction. Then, a straightforward strategy to show the embedding is first to show that the theorems following from the known axiomatization of ATL, after translation using the mapping, are all valid in the semantics of CTL.STIT, and second to show that this same direction of the mapping also preserves satisfiability.

Proposition 4.2. *The following formulas, resulting from applying the mapping of definition 4.1 to the* ATL *axiomatization in [21] are valid in* CTL.STIT*:*

(Live)	$\Box\langle A\ \mathsf{sstit}\rangle X\top$
(Term)	$\Diamond[A\ \mathsf{sstit}]X\top$
(Ags-Max)	$\Box\langle\emptyset\ \mathsf{sstit}\rangle X\varphi \rightarrow \Diamond([Ags\ \mathsf{sstit}]X\varphi$
(SA)	$\Diamond[A\ \mathsf{sstit}]X\varphi \wedge \Diamond[B\ \mathsf{sstit}]X\psi \rightarrow \Diamond([A\ \mathsf{sstit}]X\varphi \wedge [B\ \mathsf{sstit}]X\psi)$ *for* $A \cap B = \emptyset$
(FPG)	$\Diamond[A\ \mathsf{sstit}]G\varphi \leftrightarrow \varphi \wedge \Diamond[A\ \mathsf{sstit}]X\Diamond[A\ \mathsf{sstit}]G\varphi$
(GFPG)	$\Diamond[\emptyset\ \mathsf{sstit}]G(\chi \rightarrow (\varphi \wedge \Diamond[A\ \mathsf{sstit}]X\chi)) \rightarrow \Diamond[\emptyset\ \mathsf{sstit}]G(\chi \rightarrow \Diamond[A\ \mathsf{sstit}]G\varphi)$
(FPU)	$\Diamond[A\ \mathsf{sstit}](\varphi U \psi) \leftrightarrow \neg\psi \rightarrow (\varphi \wedge \Diamond[A\ \mathsf{sstit}]X\Diamond[A\ \mathsf{sstit}](\varphi U \psi))$
(LFPU)	$\Diamond[\emptyset\ \mathsf{sstit}]G((\neg\varphi \rightarrow (\psi \wedge \Diamond[A\ \mathsf{sstit}]X\chi)) \rightarrow \chi) \rightarrow \Diamond[\emptyset\ \mathsf{sstit}]G(\Diamond[A\ \mathsf{sstit}](\psi U \varphi) \rightarrow \chi)$

Also the translation of the Hilbert style derivation rules of the axiomatization in [21] results in sound rules for CTL.STIT.

This establishes one direction of the proof for an embedding. We might call this direction the 'completeness' direction for the ATL to CTL.STIT translation of definition 4.1, since it says that everything that is valid in ATL, is valid in the fragment of the CTL.STIT language we are mapping to. But we also need to establish the other direction, the 'soundness' direction. We propose to do this by proving that also satisfiability is preserved in the mapping from ATL to the language fragment of CTL.STIT. We are not going to give the mapping of ATL models to CTL.STIT models, since we have not given the classical semantics for ATL. But readers familiar with ATL can picture for themselves that any '[ATL formula, satisfying ATL structure]'-pair can be mapped quite straightforwardly to a '[CTL.STIT formula, satisfying CTL.STIT structure]'-pair. The mapping of formulas is, of course, again according to definition 4.1, and the mapping of models is not too difficult to picture (in the originating classical ATL structure, leave out all the names for choices, etc., and focus on the histories and strategies in terms of sets of histories). Preservation of satisfiability of arbitrary ATL formulas can then be proven by induction over the formula structure.

The ensuing question is what the logic is of the CTL.STIT-part that extends ATL. A definite answer in the form of a complete axiomatization of CTL.STIT we cannot give yet, but below we list some validities that are not in the ATL fragment.

Proposition 4.3. *The following are validities of CTL.STIT that are not in the ATL fragment determined by definition 4.1:*

	the S5 validities for $\Box$
	the KD validities for each $[A\ \mathsf{sstit}]X$
(Det)	$\langle Ags\ \mathsf{sstit}\rangle X\varphi \rightarrow [Ags\ \mathsf{sstit}]X\varphi$
(C-Mon')	$[A\ \mathsf{sstit}]X\varphi \rightarrow [A \cup B\ \mathsf{sstit}]X\varphi$
($\emptyset$-SettX)	$[\emptyset\ \mathsf{sstit}]X\varphi \leftrightarrow \Box[Ags\ \mathsf{sstit}]X\varphi$
(DS-incl)	$\Box[Ags\ \mathsf{sstit}]X\varphi \rightarrow [Ags\ \mathsf{sstit}]X\Box\varphi$

5. COMPARING CTL.STIT AND XSTIT

In this section we investigate *stit*-properties of the logic. In particular, we make the comparison with the logic XSTIT as first presented in [10] whose main distinguishing feature is that *stit*-actions as represented by the central XSTIT modality $[A\ \mathsf{xstit}]\varphi$ take effect in next states. It is rather natural then to suspect that XSTIT is the CTL.STIT fragment associated with the definition $[A\ \mathsf{xstit}]\varphi \equiv_{def} [A\ \mathsf{sstit}]X\varphi$. However, that is not the case.

Proposition 5.1. *The logic XSTIT [10] is not the fragment of the logic CTL.STIT determined by* $[A\ \mathsf{xstit}]\varphi \equiv_{def} [A\ \mathsf{sstit}]X\varphi$.

A very simple reason for the translation not to work is that in a recent update of the XSTIT logic, the X operator is also a standalone operator. However, there are more fundamental and more interesting semantic reasons. This first is that in XSTIT (contrary to what is claimed in [10]) we do not have the Ags-maximality property that we have in CTL.STIT, ATL and CL. In our semantics, and in ATL, Ags-maximality is ensured by condition **3(b)** in definition 3.4. In section 6 on variations on the logic, we will briefly discuss dropping this condition. Ags-maximality is an interesting property for several reasons. Note first that the axiom for Ags-maximality as given in proposition 4.2 is not in Sahlqvist form [9]. Worse, it is a version of the well-known McKinsey formula, that is not first-order definable. Roughly, Ags-maximality says the following: if some dynamic state is possible as such, that is, if even the empty set of agents has no power to exclude it from occurring, then the complete set of agents Ags actually has a winning strategy for that state. XSTIT does not satisfy this property, because (in its updated recent version) it allows for non-determinism even for the actions of Ags.

Another reason for the mismatch between XSTIT and the next-time fragment of CTL.STIT concerns the *stit*-property of 'no choice between undivided histories'. XSTIT satisfies this property, resulting in, for instance, the XSTIT validity $[A\ \mathsf{xstit}]\varphi \rightarrow [Ags\ \mathsf{xstit}]\Box\varphi$. Translating this, via the proposed translation, to CTL.STIT yields $[A\ \mathsf{sstit}]X\varphi \rightarrow [Ags\ \mathsf{sstit}]X\Box\varphi$. However, this is not valid in CTL.STIT. And it is not valid for an interesting reason, pointing to an oddity of the ATL and CTL.STIT semantics for the next operator. As said, in XSTIT the property captures the idea of 'no choice between undivided histories' that says that if an agent has a choice between two strategies, it cannot be the case that there are system histories in these two strategies that come together in some future state. However, in ATL and CTL.STIT this is not the case. in ATL and CTL.STIT agents *can* choose between histories that are undivided in next states. The point is that the choice for performing a particular strategy not only fixes a local choice for the current static state, but also fixes *all* local choices for all future static states. This means that in ATL and CTL.STIT the histories selected for the next state are a subset of the histories that would result from a strictly local choice, as in the semantics of XSTIT.

6. VARIATIONS ON THE LOGIC

One of the merits of the fairly standard modal semantics we gave is that it enables us to think freely about strengthening or weakening the logic using familiar techniques from modal logic. One option we already pointed to in section 5 is to drop the property of maximality. This corresponds to dropping the Ags-maximality axiom in definition 4.2, and dropping condition **3(b)** of the frames of definition 3.4. This more liberal setting allows for non-determinism at the level of Ags-choices. This leaves room for the environment of the agents in Ags to decide: the environment determines which of the histories admitted by the choices of Ags is the actual history.

Another interesting variation to pursue is to play around with the capabilities of the empty set of agents. First we need to have some idea about what it is that this empty set represents. In our view the empty set should not be identified with the environment. As said, we view the environment as a 'force' that may decide on non-determinism due to lack of control by Ags. Under that interpretation, the environment is like an extra agent that if it *would* be added to Ags, would again ensure Ags-maximality. Properties like C-Mon' tell us that in this logic the empty set cannot be seen as an 'extra agent' in that sense. But it can be seen as an extra agent in another sense. More in particular, we may adopt the view that the empty set of agents can be identified with the system designer. From C-Mon' it follows that whatever the empty set of agents does, is done by all the groups (including singleton groups) inhabiting the system. However, this does not exclude that we endow the empty set with real choices (in the sense that there are also alternatives). These choices would than form the constraints within which the agents inhabiting the system are free to choose. So, we could see these choices as design choices for the multi-agent system as a whole. How to make these ideas concrete in terms of logical properties is material for future research.

A natural direction for strengthening the logic is to separate the agency operators from the temporal CTL-operators, and define the semantics and axiomatics for these. The present semantics would enable us to do so quit straightforwardly. The problem with this is that the logic of the stand alone agency properties we then get is a product logic [24] (see the product-like structure in condition **7** of definition 3.4). And for three or more dimensions (here agents) S5 product logics are not finitely axiomatizable and undecidable. Because of the syntactic coupling of agency operators with temporal operators, our logic is not a product. Actually, due to the coupling, what we have is a variant on so called 'flow-products' [18]. Flow products avoid the bad meta-logical properties of products.

Finally, we come back to assumption guarantee-reasoning, as explained in section 2. The central problem of the original paper [2] concerning this type of reasoning is how to 'conjoin' behavior specification under the assumptions under which the components of the system are proven to behave correctly. The interesting problem emerging there is whether or not we can conjoin two specifications that have mutually been proven to behave correct modulo *each other's* behavior. So if component A can ensure $F\varphi$ given that B ensures $G\psi$, and component B can ensure $G\psi$ given that A ensures $F\varphi$, can we conclude, for instance, that $A \cup B$ can ensure $F(\varphi \wedge \psi)$? If we translate this question to the multi-agent system setting, and, more in particular, to the logic presented in this paper, we come to the question of whether or not the following *strong* super-additivity axiom holds.

Definition 6.1 (Strong super-additivity).

(SSA) $([B\ sstit]X\psi \rightarrow \Diamond[A\ sstit]X\varphi \wedge$
$[A\ sstit]X\varphi \rightarrow \Diamond[B\ sstit]X\psi) \rightarrow$
$\Diamond([A\ sstit]X\varphi \wedge [B\ sstit]X\psi)$ *for* $A \cap B = \emptyset$

The property is stronger than super-additivity (SA); the antecedent of the central implication is weaker, because the two ability expressions in it are made conditional on each other. It is not difficult to see that this property is not obeyed by our system. A fairly simple counter model can be made where all of $[B\ \mathsf{sstit}]X\psi$ and $[A\ \mathsf{sstit}]X\varphi$ and $\Diamond([A\ \mathsf{sstit}]X\varphi \wedge [B\ \mathsf{sstit}]X\psi)$ and $\Diamond[A\ \mathsf{sstit}]X\varphi$ and $\Diamond[A\ \mathsf{sstit}]X\varphi$ are false (making that the central implication is not obeyed thus yielding that the whole formula is false). However, on intuitive grounds, there is something to say for adding this property to the system. For now, we leave the issue for further research.

7. DYANAMIC LOGIC VERSUS STIT

We want to briefly address the question why we do not pursue a logical verification language that is based on dynamic logic. This requires some introduction, partly historical.

A multi-agent system is a computational system. Any computational system can be considered at many different abstraction levels of description. At the lowest level there is the machine it runs on. Just on top of that the machine code. Yet a level higher, maybe some higher order programming language, and so on. In case of a multi-agent system, on the higher end of this spectrum there are the believes, goals, intentions, actions in terms of which we describe the agent system (Dennet's intentional stance [15]).

Now, dynamic logic [30, 22] was designed for reasoning about programs, which in the hierarchy of abstraction described above is significantly below the highest level of description in terms of believes, goals, intentions, actions. Through the years, researchers have claimed that we could combine both levels of abstraction in one system, one logic. Examples are the KARO-framework [27] and the system of Cohen and Levesque [14], that aim to combine BDI notions with dynamic logic (in Cohen and Levesque's case all encoded in first-order logic). Whether or not that is a good idea, depends on the goals of proposed logics. If one wants to reason, within the same logic, about both (1) programming features like loop-invariants and weakest preconditions, and (2) high level BDI concepts, then this is indeed the best way to go. However, another approach, one that we would like to advocate here, is to design separate logics for separate levels of abstraction. Relations between logics for separate abstraction levels can than be laid by developing representation theorems. And here is where we think *stit*-theory can contribute to the picture. For describing the dynamics and actions on the higher BDI-level of abstraction, *stit*-theory seems more suited than dynamic logic. After all, when describing actions of agents at higher levels of abstraction, we do not use while loops, if, then else constructions, or test operators. More likely we describe an agents strategies in terms of a finite list of condition-actions pairs, of the form { if p_1 do q_1, if p_2 do $q_2, \ldots,$ if p_n do q_n}. In the strategic *stit*-logic of the present paper we express that as:

$$\begin{array}{l}[A\ \mathsf{sstit}]G(\\ (p_1 \rightarrow [A\ \mathsf{sstit}]Xq_1)\wedge \\ \vdots \\ (p_n \rightarrow [A\ \mathsf{sstit}]Xq_n))\end{array}$$

Note that what we really need a *strategic stit*-logic to be able to express this. Only in a *strategic* version of *stit* it makes sense to use the 'G' operator inside the *stit*-modality. It reflects that agents might have to perform different actions in different future states to realize the right outcome (the right system histories). And indeed, what actions the group A has to perform in which states is expressed by the finite list of condition-action pairs. Note also that we can also define strategies where a choice depends not only on the condition in the present state, but on the conditions in a series of states and the actions leading from one to the other. Finally, note that we can represent partial strategies in this was: by following the strategy starting from a certain state, we do not necessarily reach states for which an action is specified in the list of condition-action pairs.

Now readers familiar with dynamic logic and the papers describing strategies in terms of process models [32] will say that the representation dynamic logic looks rather similar. That is no surprise. Let us think about the relation between action descriptions in *stit* and in dynamic in a more systematic way. We might say that the basic modalities of dynamic logic are $[a]\varphi$, with a an atomic action name and φ a guaranteed postcondition of the action. We can increase expressivity of this basic modality by considering a fixed point semantics over the basic modalities, and allow for formulas like $\nu Z.\ \varphi \wedge [a][b]Z$, which in the notation of dynamic logic is written as $[(a;b)^*]\varphi$. Now, to compare this with *stit*-type formalisms, we can take as the central modality for these formalisms something of the form $\Diamond[A\ \mathsf{stit}]X\varphi$. The difference with the dynamic logic view is that inside the box there is a group of agents instead of an action name. Like for the basic dynamic logic modality, we can define a fixed point semantic with respect to the basic *stit*-modality. We then get, for instance formulas like $\nu Z.\ \varphi \wedge \Diamond[A\ \mathsf{stit}]X(a \wedge \Diamond[A\ \mathsf{stit}]X(b \wedge Z)$, which translates to the formula $\varphi \wedge \Diamond[A\ \mathsf{sstit}]G(\varphi \rightarrow \Diamond[A\ \mathsf{sstit}]X(a \wedge \Diamond[A\ \mathsf{sstit}]X(b \wedge \varphi)))$ of CTL.STIT. The point is that this *stit*-formula seems to express the same kind of information as the dynamic logic formula. The only difference being that the action name is not inside the modal box, but transformed to a proposition that is taken to be the effect of the action. This reflects the core difference between the views on action of both formalisms: in dynamic logic, the action is identified with its name, in *stit* with its effect. The advantage of the *stit*-view however, is that the modal box can be 'filled' with agent names. Agent interaction properties can then be studied in terms of multimodal characterization axioms (think about super-additivity, coalition monotony, etc.). However, if we leave the action (strategy) names *inside* the modal boxes, we cannot do that, which explains why until now nobody has come up with a satisfactory theory of agency in dynamic logic. The message is thus that by identifying actions with their effects, as in *stit*, we do not give up expressivity, while we gain that we can study agent interaction straightforwardly using standard modal techniques.

This is not the whole story. If we accept the above arguments and use *stit*-like formalism as specification languages, we still have the problem that for verification, we have to relate to real programs that are specified in terms of the names of basic programming operations. So, actually, we have to make the step from *stit*-like formulas, stating multi-agent system interaction requirements on higher levels of abstraction, to dynamic logic like formalism aimed at describing properties of the programs. We need representation theories for that. The step described above, where we relate the effect description of (joint) actions, to names of basic steps in a program would be a central step in such a representation theorem. We leave this issue to future research.

8. CONCLUSION

This papers defines the logic CTL.STIT, which is the join of the logic CTL with a strategic *stit*-logic. The popular logic ATL is subsumed. We argue that the additional expressive power of CTL.STIT enables us to express properties that are important for using this type of logics for multi-agent system verification. The semantics for the logic is of a new kind, and more standard than other semantics for ATL. The advantage is that we can more easily adapt the logic to different interaction properties. As one example of such a property we mentioned *strong* super-additivity. Finally, we discuss the similarities and differences with verification languages based on dynamic logic. The paper does not discuss how the present framework provides an excellent setting for investigating interactions with motivational modalities like intentions, desires and obligations, and informational modalities, like knowledge and belief. This is left for future research.

9. REFERENCES

[1] M. Abadi and L. Lamport. Conjoining specifications. *ACM Transactions on Programming Languages and Systems*, 17(3):507–534, 1995.

[2] M. Abadi and S. Merz. An abstract account of composition. *Mathematical Foundations of Computer Science*, 1995.

[3] T. Ågotnes, V. Goranko, and W. Jamroga. Alternating-time temporal logics with irrevocable strategies. In D. Samet, editor, *TARK*, pages 15–24, 2007.

[4] R. Alur, T. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.

[5] R. Alur, T. A. Henzinger, F. Mang, S. Qadeer, S. K. Rajamani, and S. Tasiran. Mocha: Modularity in model checking. In *Proceedings of the Tenth International Conference on Computer-aided Verification (CAV 1998)*, volume 1427 of *Lecture Notes in Computer Science*, pages 521–525. Springer, 1998.

[6] N. Belnap and M. Perloff. Seeing to it that: A canonical form for agentives. *Theoria*, 54:175–199, 1988.

[7] N. Belnap, M. Perloff, and M. Xu. *Facing the future: agents and choices in our indeterminist world*. Oxford, 2001.

[8] J. v. Benthem. In praise of strategies. Unpublished manuscript, 2007.

[9] P. Blackburn, M. d. Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001.

[10] J. Broersen. A complete stit logic for knowledge and action, and some of its applications. In M. Baldoni, T. C. Son, M. van Riemsdijk, and M. Winikoff, editors, *Declarative Agent Languages and Technologies VI (DALT 2008)*, volume 5397 of *Lecture Notes in Computer Science*, pages 47–59, 2009.

[11] J. Broersen. A stit-logic for extensive form group strategies. *Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference on*, 3:484–487, 2009.

[12] J. Broersen, A. Herzig, and N. Troquard. A STIT-extension of ATL. In M. Fisher, editor, *Proceedings Tenth European Conference on Logics in Artificial Intelligence (JELIA'06)*, volume 4160 of *Lecture Notes in Artificial Intelligence*, pages 69–81. Springer, 2006.

[13] R. Chisholm. Contrary-to-duty imperatives and deontic logic. *Analysis*, 24:33–36, 1963.

[14] P. Cohen and H. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(3):213–261, 1990.

[15] D. Dennett. *The intentional stance*. The MIT Press, 1987.

[16] E. Emerson. Temporal and modal logic. In J. v. Leeuwen, editor, *Handbook of Theoretical Computer Science, volume B: Formal Models and Semantics*, chapter 14, pages 996–1072. Elsevier Science, 1990.

[17] J. Forrester. Gentle murder, or the adverbial Samaritan. *Journal of Philosophy*, 81(4):193–197, 1984.

[18] D. Gabbay and V. Shehtman. Flow products of modal logics. Unpublished manuscript, 1999.

[19] M. Ginsberg and D. Smith. Reasoning about action II: The qualification problem. *Artificial Intelligence*, 35:311–342, 1988.

[20] V. Goranko and W. Jamroga. Comparing semantics of logics for multi-agent systems. *Synthese*, 139(2):241–280, 2004.

[21] V. Goranko and G. van Drimmelen. Complete axiomatization and decidability of alternating-time temporal logic. *Theor. Comput. Sci.*, 353(1-3):93–117, 2006.

[22] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. The MIT Press, 2000.

[23] T. A. Henzinger, S. Qadeer, and S. K. Rajamani. You assume, we guarantee: methodology and case studies. In *Proceedings of the Tenth International Conference on Computer-aided Verification (CAV 1998)*, volume 1427 of *Lecture Notes in Computer Science*, pages 440–451. Springer, 1998.

[24] A. Herzig and F. Schwarzentruber. Properties of logics of individual and group agency. In C. Areces and R. Goldblatt, editors, *Advances in Modal Logic*, volume 7, pages 133–149. College Publications, 2008.

[25] W. v. d. Hoek and M. Wooldridge. On the logic of cooperation and propositional control. *Artificial Intelligence*, 164(1-2):81–119, 2005.

[26] J. Horty. *Agency and Deontic Logic*. Oxford University Press, 2001.

[27] J.-J. Meyer, W. v. d. Hoek, and B. v. Linder. A logical approach to the dynamics of commitments. *Artificial Intelligence*, 1999.

[28] M. Osborne and A. Rubinstein. *A course in game theory*. MIT Press, Cambridge, Mass. [u.a.], 1994.

[29] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.

[30] V. Pratt. Semantical considerations on Floyd-Hoare logic. In *Proceedings 17th IEEE Symposium on the Foundations of Computer Science*, pages 109–121. IEEE Computer Society Press, 1976.

[31] L. Savage. *The Foundations of Statistics*. John Wiley and Sons, New York, 1954.

[32] J. Van Benthem. Extensive games as process models. *J. of Logic, Lang. and Inf.*, 11(3):289–313, 2002.

[33] W. van der Hoek, W. Jamroga, and M. Wooldridge. A logic for strategic reasoning. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 157–164, New York, NY, USA, 2005. ACM.

[34] D. Walther, W. van der Hoek, and M. Wooldridge. Alternating-time temporal logic with explicit strategies. In *TARK '07: Proceedings of the 11th conference on Theoretical aspects of rationality and knowledge*, pages 269–278, New York, NY, USA, 2007. ACM.