A simple logic of tool manipulation
(extended abstract)

Nicolas Troquard[1]

Abstract. Tools are viewed in this extend abstract as artefactual agents: agents whose goals, or function, have been attributed. We put forward an interpretation of tool usage as a social interaction between the tool and its user. Precisely, this social interaction is one of where the tool assists the user to bring about something. We lay out the first principles for a logical approach to reason about the creation and the use of tools. We also discuss some meta-logical properties of the framework.

1 Introduction

Technology is pervasive in our social environment. So much that our societies have been regarded as a huge socio-technical systems. Hence, there is an increasing need for rigorous methods to reason about socio-technical systems, model them, and verify them against a non-ambiguous specification. As formal logics have been successfully applied to the engineering of distributed systems in computer science and electronics, it seems natural to capitalize on them for engineering socio-technical systems as well.

Socio-technical systems are systems where agents in a general sense (entities capable of autonomous choices), interact with designed artefacts. Of these designed artefact, the artefactual agents, or *tools*, are especially relevant to understand the interactions in our societies. The present abstract lays out the first principles for a logical approach to reason about the creation and the use of tools.

The paradigm of multi-agent systems is general enough to encompass socio-technical systems. A tool can be seen as a particular kind of agent: one whose *function*, or goal (or still *telos*, in Aristotle's terminology) has been designed. The function of a tool is to bring about some state of the world when manipulated in a certain manner. Put another way, the function of a tool is to achieve something reactively to the agency of a user agent. We discuss this in Section 3.

Here, our study is formal. We build our logical framework upon Kanger, Pörn, and others' logic of *bringing-it-about*, that we review in Section 2. It already allows to represent in a rigorous manner events of function attribution, and events of actual usage. The full logic extends the logic of bringing-it-about with the means to talk about temporal statements. Prominently, it allows to express the properties that govern the life-cycle of a function of a tool, from its coming into existence to its destruction. We address this in Section 4.

The next section covers the foundations of the logical framework we use to reason about tool manipulation. The reader familiar with the philosophical and formal aspects of logics of agency may only browse it quickly as it contains no original research. A reader unfamiliar even with logical arguments may work the courage and maybe understand, if only a bit, the whys and hows of these specific logics for multi-agent systems.

2 Bringing-it-about logic of agency

Logics of agency are the logics of modalities E_x for where x is an acting entity, and $E_x\phi$ reads "x brings about ϕ", or "x sees to it that ϕ". This tradition in logics of action comes from the observation that action is better explained by what it brings about. It is a particularly adequate view for *ex post acto* reasoning. In a linguistic analysis of action sentences, Belnap and others ([1, 2]) adopt the *paraphrase thesis*: a sentence ϕ is agentive for some acting entity x if it can be rephrased as x sees to it that ϕ. Under this assumption, all actions can be captured with the abstract modality. It is regarded as an umbrella concept for direct or indirect actions, performed to achieve a goal, maintaining one, or refraining from one.

In this paper, we will use the logics of bringing-it-about (BIAT). It has been studied over several decades in philosophy of action, law, and in multi-agent systems ([10], [12], [11], [5], [14], [15], [6], [13], [9], [19]). Following [15], we will then integrate one modality A_x (originally noted H_x) for every acting entity x, and $A_x\phi$ reads "x tries to bring about ϕ".

The philosophy that grounds the logic was carefully discussed by Elgesem in [5]. Suggested to him by Pörn, Elgesem borrows from theoretical neuroscientist Sommerhoff ([16]) the idea that agency is the actual bringing about of a goal towards which an activity is directed. Elgesem's analysis leans also on Frankfurt ([8, Chap. 6]) according to whom, the pertinent aspect of agency is the manifestation of the agent's guidance (or control) towards a goal.

One needs a set of agents Agt and a set of atomic propositions Atm. The language of BIAT extends the language of propositional logic over Atm, with one operator E_i and one operator A_i for every agent $i \in \mathsf{Agt}$. The formula $\phi \wedge \psi$ means that the property ϕ holds and ψ holds. The formula $\neg\phi$ means that the property ϕ does not hold. The remaining logical connectives can be defined in terms of "$\wedge$" and "$\neg$". The formula $\phi \vee \psi$ means that either the property ϕ holds or the property ψ holds. The formula $\phi \rightarrow \psi$ means that if it is the case that ϕ then it is also the case that ψ. The formula $\phi \leftrightarrow \psi$ indicates that the previous implication holds and so does $\psi \rightarrow \phi$. We use $\top$ to represent a tautological truth.

Formally, the language L is defined by the following grammar:

$$\phi \quad ::= \quad p \quad | \quad \neg\phi \quad | \quad \phi \wedge \phi \quad | \quad E_i\phi \quad | \quad A_i\phi$$

where $p \in \mathsf{Atm}$, and $i \in \mathsf{Agt}$.

A formula of the language is a convenient and rigorous way to characterise properties of interactions between agents. For instance,

[1] LOA-ISTC-CNR Trento, Italy, email: troquard@loa.istc.cnt.it

imagine that $\mathsf{deadcoyote}$ represents the property of a world where the coyote is dead. The formula $(E_i A_j \mathsf{deadcoyote}) \wedge \neg \mathsf{deadcoyote}$ then represents the property that agent i brings about that the agent j attempts to brings about that the coyote is dead, and the coyote is not dead.

For any formula ϕ of L, we write $\vdash \phi$ to mean that ϕ is a theorem of the logic. The base principles of BIAT (where i is an individual agent) are:

(prop)	$\vdash \phi$, when ϕ is a classical tautology
(notaut)	$\vdash \neg E_i \top$
(success)	$\vdash E_i\phi \to \phi$
(aggreg)	$\vdash E_i\phi \wedge E_i\psi \to E_i(\phi \wedge \psi)$
(attempt)	$\vdash E_i\phi \to A_i\phi$
(ree)	if $\vdash \phi \leftrightarrow \psi$ then $\vdash E_i\phi \leftrightarrow E_i\psi$
(rea)	if $\vdash \phi \leftrightarrow \psi$ then $\vdash A_i\phi \leftrightarrow A_i\psi$

The set of all the previous principles is the axiomatics of the logic of bringing-it-about. Every base principle captures a key logical aspect of agency. BIAT extends propositional classical logic (prop). An acting entity never exercises control towards a tautology (notaut). Agency is an achievement, that is, the culmination of a successful action (success). Agency aggregates (aggreg). Every actual agency requires an attempt (attempt). The agency (resp. attempt) for a property is equivalent to the agency (resp. attempt) for any equivalent property (ree) (resp. (rea)). So, shaking hand with Zorro is equivalent to shaking hand with Don Diego Vega. Trying to spot the morning star is equivalent to trying to spot the evening star, and it is equivalent to trying to spot Venus.

The decidability of BIAT is important for its practical application in reasoning about socio-technical procedures. The proof is an adaptation of the fact that the satisfiability problem of the minimal modal logic with (aggreg) is PSPACE-complete. (See, e.g., [20].) The full proof for the fragment without the A_i operators is presented in [17]. Completing the proof is straightforward.

Proposition 1 *Let a formula ϕ in the language of BIAT. The problem of deciding whether $\vdash \phi$ is decidable. It is PSPACE-complete.*

This means that we can algorithmically decide of the validity of any property expressed in the language of BIAT. To put it bluntly, a computer can automatically reason for us about properties of action and attempts of agents.

3 Tool function and usage

We may assume that some agents in Agt are acting entities in the general sense, while others are artefactual agents. In the interest of simplicity, in this extended abstract we will assume that we have exactly one particular agent u that we call a "user", and exactly one particular artefactual agent t that we call a "tool".

Tool function. The nature of the activity of a tool is reactive to the (tentative) activity of a user. Hence, the activity of a tool is directed towards goals of the form:

$$A_u\phi \to \phi$$

That is, the *telos* or goal of a tool is "if it is the case that the user attempts ϕ then it is the case that ϕ".

The tool actually exercises its control over such a goal when it brings it about:

$$E_t(A_u\phi \to \phi)$$

Tool usage. We formalise an event of tool usage as an event in which a tool *assists* a user to obtain a goal. The description of the event "the user u achieves ϕ by using the tool t" is as follows.

$$[u : t]\phi \overset{\text{def}}{=} E_t(A_u\phi \to \phi) \wedge A_u\phi$$

So u achieves ϕ by using t when t has the function to bring about ϕ whenever u attempts ϕ, and u attempts ϕ.

This pattern is a particular instance of a more general one. In [3], we use the general pattern to study assistance and help between two acting social entities. In fact, this very pattern is a case of assistance.

It is a *successful* use because we have the following expected property by applying (success) and (prop):

Proposition 2 $\vdash [u : t]\phi \to \phi$

It is an assistance event for three reasons. First, there is an *assistee*, the user. It is a goal of u to bring about ϕ and u does try. Second, there is an *assistant*, the tool. t's guidance is reactive to u's goodwill in the action. Here, the goal of u is that ϕ holds if j tries to bring about ϕ. Third, despite Prop. 2, it is the case that $[u : t]\phi \wedge \neg E_u\phi \wedge \neg E_t\phi$ is a consistent formula. That is, it is possible that t successfully assists u to bring about ϕ, and still, neither t nor u brings about ϕ. Hence, the success of the event of tool usage described by $[t : u]\phi$ comes from some cohesion between u and t. (This cohesion is exploited in [18] to characterise group agency in BIAT.)

Grounding the user's attempts. It might seem rather arbitrary to reduce the usage of a tool to achieve ϕ, to u's attempt to achieve ϕ. This is a harmless simplification which abstracts away from the actual manipulations of the tool that the user must perform to use its functions. For instance, if t is a gun, the user might need to pull the trigger for the gun to fire and kill the coyote: this would correspond to the function $E_t(E_u\mathsf{trigger} \to \mathsf{deadcoyote})$.

Now, the fact that the user kills the coyote by using the gun is captured by:

$$E_t(E_u\mathsf{trigger} \to \mathsf{deadcoyote}) \wedge E_u\mathsf{trigger}$$

The gap between the specific manipulation of the gun and the attempt to kill the coyote can be filled in the logical theory. For instance, by stipulating the following:

$$A_u\mathsf{deadcoyote} \leftrightarrow E_u\mathsf{trigger} \vee E_u\mathsf{rope} \vee \ldots$$

It explains u's attempt of killing the coyote as the act of pulling the trigger, or passing a rope around the coyote's neck (rope), or possibly doing other relevant actions.

4 Tools as agents with designed functions

A tool is an artefact. It is what it does, and it does so because its function has been designed and attributed by a creator. In our simple setting, the user will also be the creator.

To express the properties pertaining to the existence of a tool function and the persistence of a tool function we will use the additional expressiveness of tense logics. In the following $\phi\mathcal{S}\psi$ reads that ϕ holds ever since ϕ does; $\phi\mathcal{U}^w\psi$ reads that ϕ holds until ϕ does, or ψ never occurs. ($\mathcal{U}^w$ is the *weak* until of tense logic.) At the end of this section we briefly discuss the technicalities concerning the addition of the temporal dimension.

Attributing a function. The logic can express that u attributes the function of assisting her to achieve ϕ as follows:

$$E_u E_t(A_u\phi \to \phi)$$

So, u brings about that t brings about that ϕ holds whenever u tries to achieve ϕ.

Existence of a function. A tool is an artefact. Its functions have been designed by the creator/user. We adopt the following principle.

$$E_t(A_u\phi \to \phi) \to$$
$$(E_t(A_u\phi \to \phi)\mathcal{S}E_u E_t(A_u\phi \to \phi)) \vee (E_u E_t(A_u\phi \to \phi)) \quad (1)$$

In English, if t has a function then either (i) there is a time strictly in the past where u attributed this function to t, and t has consistently held the function ever since, or (ii) u attributes this function to t at the present time.[2]

Persistence of a function. The sort of agency $E_t(A_u\phi \to \phi)$ that a tool has, is different from the sort of agency $E_a\gamma$ that a natural agent a has. If $E_a\gamma$ holds at some time, it is no assurance that $E_a\gamma$ will hold after. The agent a's goals are ever changing and so is her activity towards them. This is different for $E_t(A_u\phi \to \phi)$ because it is intended to reflect some designed function attributed to an artefact.

The activity of a tool persists. At least it persists until its function is altered by u. When a chimp takes out the leaves of a thin branch to use it as a stick and collect ants, the function of the stick will be the same the next hour, and the hour after that. Unless eventually the chimp crushes it. We then adopt the next principle:[3]

$$E_t(A_u\phi \to \phi) \to$$
$$E_t(A_u\phi \to \phi)\mathcal{U}^w E_u\neg(E_t(A_u\phi \to \phi)) \quad (2)$$

Meta-logical analysis. Adding a temporal dimension, we have considerably complicated the logical framework. However, it is in fact easy to provide a rigorous semantics to the new language by using Finger and Gabbay's *temporalisations* ([7]). We can restrict the class of all model to the constraints for which Principle 1 and Principle 2 are canonical, and we obtain the *class of models for tool manipulation*.

Combining the axiomatics of BIAT, the axiomatics of Since-Until tense logic ([4, 21]), Principle 1, and Principle 2, we immediately obtain an axiomatic theory that is sound and complete wrt. the class of models for tool manipulation.

Since BIAT is decidable (Prop 1), and so is Since-Until tense logic, a general result of Finger and Gabbay can even be applied to assert that the reasoning problem in the resulting theory is decidable.

ACKNOWLEDGEMENTS

This research is part of the project *Logical Analysis of Socio-Technical Systems* (LAST) funded by the program Marie Curie People Action COFUND Trentino.

[2] This principle must be adapted accordingly if we have more than one creator in the system.

[3] Again, this principle must be adapted accordingly if we have more than one agent in the system who can alter the function of the tool.

REFERENCES

[1] Nuel Belnap and Michael Perloff, 'Seeing to it that: a canonical form for agentives', *Theoria*, **54**(3), 175–199, (1988).

[2] Nuel Belnap, Michael Perloff, and Ming Xu, *Facing the Future (Agents and Choices in Our Indeterminist World)*, Oxford University Press, 2001.

[3] E. Bottazzi and N. Troquard, 'A philosphical and logical analysis of help'. Working title, 2013.

[4] John P. Burgess, 'Axioms for tense logic. I. "since" and "until"', *Notre Dame J. Formal Logic*, **23**(4), 367–374, (1982).

[5] Dag Elgesem, *Action theory and modal logic*, Ph.D. dissertation, Universitetet i Oslo, 1993.

[6] Dag Elgesem, 'The modal logic of agency', *Nordic J. Philos. Logic*, **2**(2), (1997).

[7] Marcelo Finger and Dov M. Gabbay, 'Adding a temporal dimension to a logic system', *Journal of Logic, Language and Information*, **1**, 203–233, (1992).

[8] Harry Frankfurt, *The Importance of what We Care About*, Cambridge University Press, 1988.

[9] Jonathan Gelati, Antonino Rotolo, Giovanni Sartor, and Guido Governatori, 'Normative autonomy and normative co-ordination: Declarative power, representation, and mandate', *Artificial Intelligence and Law*, **12**, 53–81, (2004).

[10] Stig Kanger and Helle Kanger, 'Rights and Parliamentarism', *Theoria*, **32**, 85–115, (1966).

[11] Lars Lindahl, *Position and Change – A Study in Law and Logic*, D. Reidel, 1977.

[12] Ingmar Pörn, *Action Theory and Social Science: Some Formal Models*, Synthese Library 120, D. Reidel, Dordrecht, 1977.

[13] Lambèr Royakkers, 'Combining deontic and action logics for collective agency', in *Legal Knowledge and Information Systems. Jurix 2000: The Thirteenth Annual Conference*, eds., Joost Breuker, Ronald Leenes, and Radboud Winkels, pp. 135–146. IOS Press, (2000).

[14] Felipe Santos and José Carmo, 'Indirect Action, Influence and Responsibility', in *Proc. of DEON'96*, pp. 194–215. Springer-Verlag, (1996).

[15] Felipe Santos, Andrew Jones, and José Carmo, 'Responsibility for Action in Organisations: a Formal Model', in *Contemporary Action Theory*, eds., G. Holmström-Hintikka and R. Tuomela, volume 1, 333–348, Kluwer, (1997).

[16] Gerd Sommerhoff, 'The Abstract Characteristics of Living Systems', in *Systems Thinking: Selected Readings*, ed., F. E. Emery, Penguin, Harmonsworth, (1969).

[17] N. Troquard, 'Reasoning about coalitional agency and ability in the logics of "bringing-it-about"'. Under review, 2012.

[18] N. Troquard, 'The social fabric of cohesive group agency: an abstract logical framework'. Under review, 2013.

[19] Nicolas Troquard, 'Coalitional Agency and Evidence-based Ability', in *Proc. of AAMAS'12*, eds., Conitzer, Winikoff, Padgham, and van der Hoek, pp. 1245–1246. IFAAMAS, (2012).

[20] Moshe Vardi, 'On the Complexity of Epistemic Reasoning', in *Proc. of Fourth Annual Symposium on Logic in Computer Science (LICS'89)*, pp. 243–252. IEEE Computer Society, (1989).

[21] Ming Xu, 'On some u, s-tense logics', *Journal of Philosophical Logic*, **17**(2), 181–202, (1988).