A formal theory for conceptualizing artefacts and tool manipulations

Nicolas TROQUARD [1]
Laboratory for Applied Ontology, ISTC-CNR, Trento, Italy

Abstract. Artefacts (physical and institutional) are ubiquitous of our social environment. We live in a tight network of socio-technical systems, which are systems where agents interact with created objects. There is an increasing need for rigorous methods to model, specify, and reason about socio-technical systems in general, and about artefacts and their functions in particular. We propose a formal theory that serves at the conceptualization of artefacts and their manipulations: design, implementation, existence, use, and persistence.

Keywords. ontology, logic, artefacts, functions, engineering

Introduction

Technology (physical and institutional) is pervasive in our social environment. So much that our societies have been regarded as large 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 and institutional systems as well.

Socio-technical systems are systems where natural agents (entities capable of autonomous choices), interact with designed artefacts. Of these designed artefact, *tools* are especially relevant to understand the interactions in our societies. A tool can be seen as a particular kind of agent: one whose *function*, or final end (or still *telos*, in Aristotelian terms) has been designed.[2] The function of a tool is that some state of the world is realized when the tool is manipulated in a certain context and in a certain manner. A stapler joins sheets of paper when it is loaded and is operated properly.

The paradigm of multi-agent systems is general enough to capture socio-technical systems. Here, our study is at the intersection of MAS and the realm of knowledge representation and reasoning. We aim to provide with a formal pre-ontology which captures and assists to the rigorous reasoning about the following commonsensical features of artefacts and tools:

[1]Nicolas Troquard, LOA-ISTC-CNR, Trento, Italy; E-mail: troquard@loa.istc.cnr.it. This work was supported by a Marie Curie fellowship (project "LASTS") under grant PCOFUND-GA-2008-226070.

[2]The teleological view on functions is not confined to philosophy but is also practically employed in engineering [23].

- Artefacts are agents, acting upon the world: physical world and institutional world;
- Artefacts have designed functions and realized functions; (designed vs. implemented/effective)
- *Functions of artefacts have been attributed;*
- Functions of artefacts persist;
- Artefacts can malfunction;
- Artefacts can have non designed features; (effective functions that are by-products of their implementation);
- *Artefacts and tools in particular can be used by other agents (natural or artefactual) to bring about (possibly) otherwise unattainable.*

Our focus is then on the abstract characteristics of artefacts and tools. We look particularly at the logical form of their functions. We also address their dynamic properties. First, providing a genealogical analysis: how they come to exist, and how they persist. Second, we propose an analysis of the actions carried out *with* tools: how they are used, what one brings about with them.

A disclaimer is in order. The present note lays out the primordial ontological choices towards a logical approach to conceptualize artefact functions, and reason about the *creation*, the *existence* and the *use of tools*. Although we tend to show a natural 'allegiance' to ontologies, the present work is not a work of ontology proper: we do not answer *completely* the question "what is an artefact?". Less than an ontology, we present a pre-ontology. It is designed to support the development of proper formal ontologies of artefacts and tools. However, our work is representational as we provide the language to speak about functions, artefacts, their creation, their use, and their persistence. Hence, the present work is to propose a conceptualization of a domain of application, with a carefully chosen set of primitives and of logical relations between them. If it is not directly suitable for application run-time reasoning, it provides "*conceptual handles* with which to carry out a coherent and structured analysis of domains of interest." [7].

We build our logical framework on the basis of a first-order theory to talk about static facts. In this work and in the interest of brevity, we sketch a language with a limited number of *ad hoc* predicates. Instead, an existing fully fledged foundational ontology could be used.[3] We build our logical framework for agency upon Kanger, Pörn, and others' logic of *bringing-it-about*. It already lets one to represent in a rigorous manner events of function attribution, and events of actual tool usage. The full logic extends the logic of bringing-it-about with the means to talk about temporal statements. Prominently, the tense logic allows for expressing the properties that govern the life-cycle of a function of a tool, from its coming into existence to its destruction.

The logical framework is presented and its components are motivated in Section 1. The key logical postulates are also listed. Functions, function manipulations, and function life-cycles are addressed in Section 2. Section 3 provides a few examples. We model in our language how agents use artefacts to bring about states of affairs. We provide in Section 4 a few pointers to related issues that have been tackled in earlier literature in ontology, logics, and multi-agent systems. Finally, we conclude and discuss last issues in Section 5.

[3] A foundational ontology is formulated in terms of general, axiomatized, cognitively and philosophically motivated concepts. Examples are BFO (`http://www.ifomis.org/bfo`), and DOLCE (`http://www.loa.istc.cnr.it/DOLCE.html`).

1. Logical framework

We need some very basic elements of foundational ontology and will use a first-order theory to formalize them. We need also a notion of practical causation—agency—and of purposeful attempts and will use the modalities of the logics of bringing-it-about. Finally, we need some means to talk about temporal properties of the life-cycle of artefacts and will use the Until-Since tense logic. We present their conceptual motivations step by step. We end the section by formalizing the basic postulates of the resulting logic.

1.1. A first-order and modal logic of agency

Before introducing any modality, we define what constitutes the basic logical framework. It is a first-order theory, and will allow us to define the categories that are relevant to the formulation of an ontology of artefacts, and offers a way to quantify over individual objects.

We let Var a set of individual variables, and a set of functions Fun. Although there is no difference between constants and 0-ary functions, we also let Const a set of individual constants. In the interest of intelligibility we will use a variety of lettering for variables and constants (a, t, x, i, r, v, etc) that conveys the best intuitions. We let Pred a set of predicate symbols. Among these, we have in particular the monadic predicates AGT, ART, FUN, $HNDF$, NAT, OCC, ROL, and TOL, which will correspond to the main properties that the individuals can have in our simple ontology. We will use explicitly the following glossary:

$AGT(a)$	(term) a is (denotes) an agent	$ART(t)$	t is an artefact
$FUN(t)$	t is a functioning artefact	$HNDF(t)$	t is has non-designed features
$NAT(a)$	a is a natural agent	$OCC(e)$	e is an occurring event
$ROL(r)$	r is a role	$TOL(t)$	t is a tool

The status of artefacts as an ontological class was already acknowledged in the literature [5]. We use ART to denote this property. The property AGT corresponds to the classes of agentive objects in DOLCE [29]: `Agentive Physical objects` and `Agentive Social Objects`. Roles are concepts that are introduced in an extension of DOLCE [30] and replicated in [9].
Also, the binary predicate RO will be a relation between objects.

$$RO(a,r) \quad \text{agent } a \text{ occupies the role } r$$

The terms of the theory are recursively defined by the grammar $t ::= v \mid c \mid f(t,\ldots,t)$ where $v \in \mathsf{Var}$, $c \in \mathsf{Const}$, and $f \in \mathsf{Fun}$ is an n-ary function. So a term is a variable, a constant, or the instantiation of a function with terms.

Modalities 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 [3] 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 re-

garded 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 (E.g., [25], [32], [14], [36]). Following [33], we will also integrate one modality A_x (originally noted H_x) for every acting entity x. In [33], $A_x\phi$ reads "x tries to bring about ϕ".

The modalities are abstract and are meant so. Some observations and formal principles are crucial nonetheless to the intelligibility of their specific use in this note.

The philosophy that grounds the logic of the E_x modality was carefully discussed by Elgesem in [14]. Elgesem borrows from theoretical neuroscientist Sommerhoff 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 [16, Chap. 6] according to whom, the pertinent aspect of agency is the manifestation of the agent's guidance (or control) towards a goal. It is generally acknowledged that one cannot exercise control over the tautological truths of the world. Hence, it is never the case that an agent brings about a tautology. The crucial characteristic is that bringing about is a successful agency. Hence, something is true whenever it is brought about by some agent. This does not mean that conversely, something is true only if it is brought about. However, it implies that no agent can bring about a logical contradiction, for this contradiction would have to be true of the world. These assumptions will be reflected in the axiomatization below.

Trying, the modality A_x, has received much less attention. It is a modality of possibly unsuccessful agency. In other words, it is consistent that something does not hold when an agent tries to bring it about. What it must reflect however, is a volitional attitude. Trying is purposeful and it is how it should be read.

On the other hand, actual agency can be without purpose. One can bring about some state of affairs, and even be held responsible for it, and yet have not done it on purpose. Accidents, fixed action patterns, impulsive actions, and other natural drives of nature are a reality of our world. The modality E_x should then not be read as a necessarily purposeful bringing about.

One needs a specific set of agents $\mathsf{Agt} \subseteq \mathsf{Const}$. The language of BIAT extends the language of first-order logic with one operator E_i and one operator A_i for every agent $i \in \mathsf{Agt}$. Although recent work in BIAT preferred a purely propositional basis, a logic of bringing about extending a first-order language has been proposed early on in the modern history of the logics of agency. See in particular [32, Ch. 1]. Semantic aspects can be retrieved from [1].

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

$$\phi ::= P(t,\ldots,t) \mid t = t \mid \neg\phi \mid \phi \wedge \phi \mid \forall v.\phi \mid E_i\phi \mid A_i\phi$$

where $P \in \mathsf{Pred}$ is an n-ary predicate, $v \in \mathsf{Var}$, and each occurrence t is a term as defined by the previous BNF, $i \in \mathsf{Agt}$. (Other logical connectives $\vee$, $\rightarrow$, $\leftrightarrow$, $\exists$, and constants $\bot$, $\top$ are defined as usual.) A formula of the language L is a convenient and rigorous way to characterize properties of interactions between agents. For instance, consider $FUN(\mathsf{device})$ that represent the property of a world where the device is functioning. The formula $(E_iA_jFUN(\mathsf{device})) \wedge \neg FUN(\mathsf{device})$ then represents the property that agent i brings about that the agent j attempts to bring about that the device is functioning, and

the device is not functioning. Intuitively, it represents the end result of a scenario where i incited in some way j to take actions towards making the device work, but the device is not actually functioning. It indicates a failed attempt on the part of j, as well as a failed delegation on the part of i.

1.2. Until-Since tense logic

There are two approaches to adding a temporal dimension to an existing logic system: (i) the first-order "internalization" where all predicates and modalities are given the additional parameter of a time index, (ii) the "external" approach consisting in extending the language with temporal modalities. We opt for the latter for it is a modular extension, and because the temporal properties are then arguably easier to write and read. Our background theory described in Sec. 1.1 is temporalized with Until-Since linear tense logic as in [15]. The modalities of $\mathscr{U}$ and $\mathscr{S}$ were introduced by Kamp [24].

Formally, the language $T(L)$, the language of the temporalization of L, is defined by the following grammar:

$$\gamma ::= \phi \mid \neg\gamma \mid \gamma \wedge \gamma \mid \gamma \mathscr{U} \gamma \mid \gamma \mathscr{S} \gamma$$

where ϕ is a formula in L. The additional expressiveness of tense logic will help us to express the life-cycle of artefacts: the properties pertaining to the existence of an artefact function and the persistence of an artefact function. In the following $\phi \mathscr{S} \psi$ reads that ϕ holds ever since ψ does; $\phi \mathscr{U}^w \psi$ reads that ϕ holds until ψ does, or ψ never occurs. ($\mathscr{U}^w$ is the *weak* until of tense logic and can be defined from $\mathscr{U}$.)

1.3. Basic properties

For any formula ψ we will write $\vdash \psi$ to state that ψ is a theorem of our ontology: ψ is the instance of an axiom, or it can be derived by applying axioms and rules of inferences. We provide the key axioms and rules of inferences in this section, and discuss a few other properties that do not hold the status of theorem.

Static elements of ontology First, our theory is a first-order theory.

(fol) $\vdash \phi$, when ϕ is a classical tautology in FOL

Since we have a dedicated set of agents, we may as well bring it forth:

(a1) $\vdash AGT(i)$, when $i \in \mathsf{Agt}$ (a2) $\vdash \neg AGT(i)$, when $i \notin \mathsf{Agt}$

It is an hypothesis of the literature on artefacts, and the present paper as well, that agents, artefacts and tools are intimately related. Rather involved (temporal) relationships will be devised later, but some basic taxonomy can already be presented.

(a3) $\vdash TOL(t) \rightarrow ART(t)$ (a4) $\vdash ART(t) \rightarrow AGT(t)$
(a5) $\vdash ART(i) \rightarrow \neg NAT(i)$ (a6) $\vdash NAT(i) \rightarrow AGT(i)$
(a7) $\vdash RO(a,r) \rightarrow AGT(a) \wedge ROL(r)$ (a8) $\vdash FUN(t) \rightarrow ART(t)$

Some simple inferences are possible already. (E.g., no tool is a natural agent.) Deeper conclusions about the static world would require the richer glossary of a foundational ontology. Instead we will focus on the interplay of our few primitives with the modalities of our language $T(L)$.

Bringing about and trying The base principles of BIAT (where i is an individual agent) are:

(notaut)	$\vdash \neg E_i \top$	(success)	$\vdash E_i\phi \rightarrow \phi$
(ree)	if $\vdash \phi \leftrightarrow \psi$ then $\vdash E_i\phi \leftrightarrow E_i\psi$	(rea)	if $\vdash \phi \leftrightarrow \psi$ then $\vdash A_i\phi \leftrightarrow A_i\psi$

An acting entity never exercises control towards a tautology (notaut). Agency is an achievement, that is, the culmination of a successful action (success). 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.

BIAT is a weak logic and then rather permissive to the supplementation of domain specific principles. We thus add the following ones which will offer the logical basis for realistic function manipulations later:

(b1)	$\vdash E_i(\psi \wedge A_j\phi) \rightarrow E_iA_j\phi$	(b2)	$\vdash E_i(\psi \wedge \neg A_j\phi) \rightarrow E_i\neg A_j\phi$
(b3)	$\vdash E_i(\psi \wedge E_j\phi) \rightarrow E_iE_j\phi$	(b4)	$\vdash E_i(\psi \wedge \neg E_j\phi) \rightarrow E_i\neg E_j\phi$

Postulating these, while $E_i(\phi \wedge \psi) \rightarrow E_i\phi$ does not hold in general, puts the focus of BIAT's E_i modality on function manipulations: bringing about that something (does not) (tries to) bring about.

Tense logic We omit altogether the axiomatization of Until-Since logic to save space. The main reference is Xu [37] which improved upon Burgess [10].

Mixed properties One essential ontological status of properties of objects concerns their rigidity. Usually, the means to talk about rigidity is provided by the addition of an alethic modality in the language. But our temporal dimension suffices and is more precise, too. We have said and will emphasize it again later that artefacts' functions are attributed. Before any function has been attributed to an object, this object is not an artefact. The object with the property of being an artefact once was not an artefact. This is captured by:

$$\text{(us0)} \quad \vdash ART(a) \rightarrow \top \mathscr{S} \neg ART(a)$$

The formula is equivalent to saying that if a is an artefact there was a moment in the past where a was not an artefact.[4] We say that the property of being an artefact is anti-rigid. See [20].

It is worth noting at that point that our work addresses the design and implementation of individual, *token* artefacts, as opposed to *types* of artefacts. The normal process is then to attribute some (desired) function to an object and to implement the function in the object. For instance, when creating a technical (physical) tool, one needs not start with a physical object, even when the implementation yields a physical object. One may first attribute a function to what DOLCE calls a `Mental Object` which is a `Non-Physical Object`. This mental object exists as a concept in the mind of a designer. It exists possibly on paper as a chart, in which case the chart is not the physical object but a mere projection of it. Later, this mental object can transform into a physical object via implementation.

[4] The past operator P of temporal logic is defined as $P\phi = \top \mathscr{S} \phi$.

As we ultimately obtain an implemented and functioning tool, the object is the same as the one designed earlier. Only its properties have changed, and in particular the one of being an artefact. It is related to a much debated issue in applied ontology. In [8], the authors instead favour a multiplicative approach: when an object is selected for a purpose then a new object, an artefact, comes into being. (See also [20] for details.) Being an artefact becomes a rigid property. In this note we do not adopt a multiplicative approach.

2. Functions

A function can be any state of affairs ϕ towards which an activity may be directed, or towards which an activity may be desirably directed.

When t is an artefact, and $E_t\phi$ is deemed[5] true, the formula ϕ is a telos of t, or a realized function. When $A_t\phi$ is deemed true, the formula ϕ is a designed or attributed function of t. It should be thus clear by now that we are after a notion of what we may call, borrowing from computer science, *denotational* function instead of an *operational* one. We are interested in what the artefact does, not how it does it.

Functions as states of affairs instead of mathematical mappings may at first seem too abstract. But they have gained their ontological status before [17]. We reconcile state of affairs to input-output mappings in the following section. For illustrative purposes, we will then proceed to characterize a variety of such functions. We can only hope to give a good sense that our simple setting is versatile in this regard. We will also see how they can be usefully manipulated as well as we will explain their life-cycle as functions of an artefact.

2.1. Normal form

We justify our notion of function by a commonplace interpretation of the material implication in classical logic.[6] A *Function Normal Form* is a formula: $\phi_i \rightarrow \phi_o$ where ϕ_i can be thought of as the *input* to the function while ϕ_o may be thought of as the *output* of the function. Alternatively, ϕ_i may be seen as the *guard* of the function and ϕ_o may be seen as the *consequence product* of the function.

It is a normal form because every formula can be written in this form. It is indeed a trivial observation that: $\phi \leftrightarrow (\top \rightarrow \phi)$ is a classical tautology. That is, a formula ϕ corresponds in itself to a function whose consequence product is realized for *any* guard.

Normal forms are not unique, though. The formulas $\top \rightarrow (\phi \vee \neg\psi)$, $\psi \rightarrow \phi$, and $\neg\phi \rightarrow \neg\psi$ are equivalent formulas in function normal form. Principles (ree) and (rea) ensure that bringing about (resp. attempting) a function is equivalent to bringing about (resp. attempting) *any* of its normal forms.

2.2. Logical forms of functions

"User-specific" functions The following types of functions are crucial in security and safety system environment.

[5] Defining what the truth of a formula is would require the definition of a semantics. It would be easy but prolix. See also Section 5.

[6] We use the material implication for simplicity. A similar analysis could be done, e.g., in linear logic [31].

A function is agent specific when its guard involves the bringing about or the trying of some specific agent. Functions specific to agent a have the form: $E_a\phi_i \rightarrow \phi_o$ (resp. $A_a\phi_i \rightarrow \phi_o$). The function is to have ϕ_o true when agent a brings about (resp. tries to bring about) ϕ_i.

A function can also be specific to agency *in a role*. Artefacts with such functions are ubiquitous in organizations where for instance, pronouncing some dedicated words have an institutional effect only if the person uttering them occupies a specific role. Functions specific to the role r have the form: $(\exists v.RO(v,r) \wedge E_v\phi_i) \rightarrow \phi_o$.

Among the agent specific functions, we can find the striking cases of assistance and refraining functions that we comment now.

Assistance functions In their famous paper "The Extended Mind" [13], Clarke and Chalmers discuss at length the example that the couple pen/paper is part of the cognition that allows someone making a long multiplication. A person without extraordinary mental capabilities would rarely be able to multiply big numbers in their head. Give them a sheet of paper and a pen, and a reasonably trained child will not fail once to give the right result. The function of the artefact pen/paper is a simple function of assistance. Let ϕ_i is "the result to a long multiplication". One function attributed to the pen/paper artefact is that ϕ_i holds whenever a reasonably trained agent a tries to ϕ_i. That is $A_a\phi_i \rightarrow \phi_i$. Assistance functions populate most cybernetic systems (e.g., aircraft landing systems) and are likely bound to become pervasive (e.g., nerve controlled prosthetics). "When you think about something and don't really know much about it, you will automatically get information," said Google CEO Larry Page. "Eventually you'll have an implant, where if you think about a fact, it will just tell you the answer" [28]. If we accept that thinking about a fact is trying to know its truth value, what Page describes is an assistance function.

Refraining functions Refraining functions are less popular than assistance functions, but they are very little different from a logical perspective. Hence, cybernetic implants may well be used to (assist to) refrain from producing some results. Such functions have the form: $A_a\phi_i \rightarrow \neg\phi_i$. That is, ϕ_i is not the case when agent a tries to bring about ϕ_i. In other words, the function is to ensure that a's attempt to bring about the specific state of affairs ϕ_i is not successful. E.g., it can be artefact that brings about only the function $A_aSCRATCHED(a) \rightarrow \neg SCRATCHED(a)$ would be a *non-coercing*[7] artefact to help a patient to not scratch, after surgery for instance. (Although it would not prevent from a spontaneous 'un-attempted' scratch.)

Functions on perdurant occurrences With a predicate OCC where $OCC(e)$ means that the perdurant (or event) e occurs, it is simple to represent a function that forces an event e_o to occur whenever an event e_i does: $OCC(e_i) \rightarrow OCC(e_o)$.

Because our base language is a first-order theory, we may also capitalize on its very notion of function, that is, on the objects in Fun. A function of an artefact can be associated to a term f, and we can represent it in normal form as: $OCC(e) \rightarrow OCC(f(e))$.

[7] It is not the case that $E_t\neg E_aSCRATCHED(a)$ or $E_tE_a\neg SCRATCHED(a)$ follow from $E_t(A_aSCRATCHED(a) \rightarrow \neg SCRATCHED(a))$.

2.3. Design and implementation

From commonsense definition an artefact is the product of an agent. The intentional stance on functions has been taken among others by Searle [34], according to whom an artefact has a function only in relation to *human intentionality*.

Although it is a common assumption that it is the product of an intentional agent [4], we do not make such an assumption in the strict sense of the word "intentional" often used, e.g., in AI. We only want to acknowledge that the attribution of a function to an artefact is brought about by an agent, whether it is intentional, accidental or impulsive. This offers the advantage to greet spontaneous creations into our class of artefacts. For instance, when an agent impulsively attributes to a plastic folder the function of shielding against some water thrown at her, the ontology is consistent with the water-shield being an artefact. It also leaves open the possibility of artefacts creating artefacts.[8]

We refute more resolutely Searle's reference to *human*'s agency. Experimental research in behavioral biology clearly shows non human animals can create and use artefacts. Corvids (crows, magpies) or primates (chimps, capuchin monkeys) are well-known adepts of tools. The impact on the ontology is little, though: we just do not necessarily need to rely on a property of "being human" to make sense of artefacts.

Designing an artefact, an agent can attribute a function to an object (non necessarily a physical object) simply by bringing about that one of the object's purpose is that ϕ holds: $E_u A_t \phi$.

Analogously, an agent can implement a function to an object simply by bringing about that one of the object's telos is ϕ: $E_u E_t \phi$.

2.4. Life-cycle

Hilpinen proposes a no-fuss definition of artefacts: "an object is an artefact if and only if it has an author" [22]. The functions of artefacts and tools have been designed by an agent. A necessary condition for the existence of a designed function attributed to an artefact is then the following:

$$\text{(us1)} \quad \vdash (ART(t) \wedge A_t \phi) \rightarrow (A_t \phi . \mathscr{S}(\exists v . E_v A_t \phi)) \vee (\exists v . E_v A_t \phi)$$

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

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

A function attributed to an artefact persists. At least it persists until its function is altered by some agent. That is, if an object is an artefact with some attributed function, then it holds the attributed function until some agent updates the design in a way that the function is not attributed anymore.

[8] The intentionality of agent a's actions could be stipulated by the axiom $NAT(a) \wedge E_a \phi \rightarrow A_a \phi$.

$$\text{(us2)} \quad \vdash (ART(t) \wedge A_t\phi) \rightarrow A_t\phi \mathscr{U}^w (\exists v.E_v \neg A_t\phi)$$

Note that a weak until tense operator is used, meaning that the agent that will deprive an artefact from its function may never exist.

Can the same be argued for implemented functions? Some activities of a tool persist indeed. When a chimp takes the leaves off a twig to use it as a stick and fish for termites, the realized function of the stick will be the same the next hour, and the hour after that. Unless eventually the chimp crushes it. But an artefact can lose its effective function by some natural accident, or by normal wear. The stick implemented by the chimp will probably have lost its function within two years just by normal decomposition, and without the intervention of an agent. An analogous argument can be made about the existence of a realized function. An artefact can effectively perform some function by the nature of its built, or by the purely logical consequence of its built. Objects can be elected to the status of artefact by the mere selection by an agent and without physical modification. Classical examples of the literature on artefacts are a pebble chosen to serve as a paperweight, or a fallen tree used as a bench. Both the pebble and the fallen tree held their realized function. It was not implemented by an agent. We must then conclude that there is no counterparts to the two previous postulates for the realized functions of artefacts.

2.5. Malfunction and non-designed features

In [8], the authors propose a necessary and sufficient condition to decide whether an artefact is malfunctioning at some instant. As they write "It simply does not possess all the capacities attributed to it". It is simple because they have the means to quantify over capacities; Capacities are terms in their framework. The closer we have to a capacity of an artefact t is a state of affairs ϕ such that $A_t\phi$ holds. We have no means of quantifications over states of affairs. We will then have to content ourselves with a sufficient condition for malfunction:

$$\text{(malfun)} \quad \vdash ART(t) \wedge A_t\phi \wedge \neg E_t\phi \rightarrow \neg FUN(t)$$

If t is an artefact with the designed function ϕ and does not realize ϕ then t is not a functioning artefact.

In a very similar way, we have a sufficient condition to establish that a tool has non-designed features:

$$\text{(hndf)} \quad \vdash ART(t) \wedge E_t\phi \wedge \neg A_t\phi \rightarrow HNDF(t)$$

3. Manipulations of tool tokens with examples

We can finally present ways of conceptualizing the creation and the use of artefacts.

Making an artefact, designing and implementing functions Designing a certain *type* of object, artefact, or tool, is done at a systemic level. Say a tool of some type T has exactly the functions ϕ_1 and ϕ_2. The fact that "x is a tool of kind T" is simply captured axiomatically as:

$$(\dagger) \quad \vdash T(x) \leftrightarrow A_x\phi_1 \wedge A_x\phi_2$$

A tool x is an implemented T, that is an object of type IT can be captured as:

$$(\dagger\dagger) \quad \vdash IT(x) \leftrightarrow E_x\phi_1 \wedge E_x\phi_2$$

From there, the design of a *token* t as a artefact of type T can be brought about by an "engineer" agent *eng*: $E_{eng}T(t)$.

From (†) and (ree), we infer that $E_{eng}(A_t\phi_1 \wedge A_t\phi_2)$. In turn, by (b1) and (prop) we can infer that indeed, agent *eng* designs t with the function ϕ_1 and also designs t with the function ϕ_2: in formula $E_{eng}A_t\phi_1 \wedge E_{eng}A_t\phi_2$. This is crucial to the start of the life-cycle of the functions.

Actually implementing t to have these functions is done symmetrically with $E_{eng}IT(t)$. From it, and with (††), (ree), and (b3) we can infer that $E_{eng}E_t\phi_1 \wedge E_{eng}E_t\phi_2$.

How is a tool used, now? We present two brief examples. One on the usage of an assisting function, one on the usage of a role-specific function.

Case of use in assisting events We formalize here an event of tool usage as an event in which a tool *successfully assists* a user to obtain a goal. The description of the event "the agent (user) a achieves ϕ with the assistance of the tool t" is like so: $[AE(a,t)]\phi \stackrel{\text{def}}{=} E_t(A_a\phi \rightarrow \phi) \wedge A_a\phi$. So a achieves ϕ by using t when t has the function to bring about ϕ whenever a attempts ϕ, and a attempts ϕ.

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

Proposition 1 $\vdash [AE(a,t)]\phi \rightarrow \phi$

It can be qualified as an assistance event for three reasons. First, there is an *assistee*, the user. It is a volition of a to bring about ϕ and a does try. Second, there is an *assistant*, the tool. t's guidance is reactive to a's goodwill in the action. Here, the realized function of a is that ϕ holds if a tries to bring about ϕ. Third, despite Prop. 1, it is the case that $[AE(a,t)]\phi \wedge \neg E_a\phi \wedge \neg E_t\phi$ is a consistent formula. That is, it is possible that t successfully assists a to bring about ϕ, and still, neither t nor u brings about ϕ. Hence, the success of the event of tool usage described by $[AE(t,a)]\phi$ comes from some cohesion between a and t.

Case of use in institutional events We formalize here an event of tool usage as an institutional event. The description of the event "the agent a, acting in the role r, achieves ϕ_o in the institution *inst*, by doing ϕ_i" is as follows: $[IE(a,r,inst,\phi_i)]\phi_o \stackrel{\text{def}}{=} E_{inst}((\exists v.RO(v,r) \wedge E_v\phi_i) \rightarrow \phi_o) \wedge E_a\phi_i \wedge RO(a,r)$. It is a *successful* institutional action because we have again the following expected property:

Proposition 2 $\vdash [IE(a,r,inst,\phi_i)]\phi_o \rightarrow \phi_o$

Say that in the institution $corp$, when an agent holding the role of head of human resources (*hhr*) signs a employment contract with Ann, then Ann is hired. That is, we assume that $corp$ is an institutional tool that has (among others) the implemented function: $(\exists v.RO(v,hhr) \wedge E_vSIGNED(\mathsf{ann})) \rightarrow HIRED(\mathsf{ann})$. For an agent a, $E_aSIGNED(\mathsf{ann})$ stands for a signs an employment contract with Ann. The proposition $HIRED(\mathsf{ann})$ stands for Ann is hired. Now, say that John is an agent, and holds the role of head of human resources. That is, $AGT(john)$ and $RO(john,hhr)$. He can successfully hire Ann

by simply bringing about that $SIGNED(\mathsf{ann})$ is the case. Indeed, it is readily seen that: $[IE(john, hhr, corp, SIGNED(\mathsf{ann}))]HIRED(\mathsf{ann}) \rightarrow HIRED(\mathsf{ann})$.

4. Related literature

We have already referred to related work in the previous sections. We present here a selective overview of the literature on artefacts and functions. The references [23] and [27] are relevant work in philosophy of engineering. They contain recent cross-disciplinary surveys on functions and artefacts.

Artefacts in applied ontology Artefacts are often seen as a kind of physical objects [8,6] in applied ontology. An artifact α is a physical object which an agent (or group of agents) creates by two, possibly concurrent, intentional acts: the selection of a material entity (as the only constituent of α) and the attribution to α of a quality or capacity [8]. Moreover, in [8] artefacts are explicitly classified as `Non-Agentive Physical Objects`.

[5] has only a category for "technical artefacts". Things are rather different in [6] where artefacts can be further qualified as technical, engineering, ontological, by-product, or nature-made. In CYC/OPENCYC, the top-class of artefacts represents `Artefact Generic` and collects the "things created by agents". Hence, "legal agreements" are found in the class, alongside "paintings", "hammers" and "bird nests". The first falls into the subclass of `Artefact Intangible`. An interesting dimension of artefacts in OPENCYC is represented by the subclasses `Artefact Agentive` and `Artefact NonAgentive`. With our axiom (a4), we only acknowledged the former kind.

Ontology of functions Garbacz [17] proposes a theory of functions as states of affairs. Mizoguchi and Kitamura [26] and Arp and Smith [2] each develop a foundational ontology of functions, respectively as extensions of YATO and BFO. Borgo et al. [5] formalize two accounts of engineering functions in the unified framework of the DOLCE ontology. They formalize Chandrasekaran and Josephson's *Functional Basis* (FB) approach [12] and Stone and Wood's *Functional Representations* (FR) approach [35]. FB functions are most similar to ours. They are a kind of perdurant performing, so to say, transformations of flows. Flows can be input flows, or output flows. A flow has to be of one and of only one of the following kinds: `Material Flow`, `Energy Flow`, or `Signal Flow`.

Logic and multi-agent systems In philosophical logic, agency for a function bears some resemblance with the modal approach to strict implication: $\Box(\phi \rightarrow \psi)$. More relevant to MAS, when *inst* is an (institutional) artefact, $E_{inst}(\phi_1 \rightarrow \phi_2)$ may be interpreted as ϕ_1 *counts as* ϕ_2 in the context of *inst*.

This modal approach was exploited by Grossi et al. [19] for the counts-as. Three modal logics are proposed and three sorts of counts-as are formalized: classificatory, proper classificatory, and constitutive. They are essentially incomparable with our formalization as $E_{inst}(\phi_1 \rightarrow \phi_2)$. A first reason is that some axioms are assumed by them that are not theorems of our theory, and conversely. A second reason is that they admit some logical principles of interactions between different institutions, while we have not assumed anything in particular. A third reason is that the underlying modal logics used by Grossi et al.'s are normal modal logic, while our modalities are non normal. However, their formalization of *proper classificatory* counts-as decidedly goes the way of a non normal logic that obeys (notaut), a central principle of bringing-it-about.

In [21], Herzig et al. present a logic that enables the reasoning about event generations (causal and counts-as). The counts-as component, noted $\alpha_1 \overset{r}{\rightsquigarrow} \alpha_2$ says that the event α_1 caused while acting in role r counts as the institutional event α_2. The authors assume only one institution that we can designate as *inst* here. In our formalism, $\alpha_1 \overset{r}{\rightsquigarrow} \alpha_2$ corresponds to $E_{inst}((\exists v.(RO(v,r) \wedge E_v OCC(\alpha_1))) \rightarrow OCC(\alpha_2))$. That is, the (institutional) tool *inst* brings about that event α_1 occurs whenever an agent holding the role r brings about the occurrence of event α_1.

The bringing-it-about modality E_{inst} where *inst* is an institutional agent is not new. It is a prominent use of the modality in MAS (e.g., [11]).

5. Conclusions

We have proposed a formal theory to conceptualize artefacts and tool manipulation. The basic bricks that constitute our formal world is a first order theory. It allows one to represent what elementary objects are and what are their ontological relationships. Agents, natural and artefactual, then act upon the world. This is conceptualized in the logics of bringing-it-about. Crucially, this permits to talk about the *design*, the *implementation*, and the *use* of artefacts by natural agents. Finally, the logic is temporalized by means of Until-Since tense logic. This allows talking about the life-cycle of artefactual functions.

The central MAS feature is the manipulation of artefacts by agents: design, implementation, and use (Section 3). As far as we are aware there is no comparable logical theory to acknowledge the *use* by an agent of the function of another agent to bring about some state of affairs that may have been impossible to achieve otherwise.

Here, conceptual work plays the lead at the expense of technical work. To serve as a conceptual handle for the analysis of tool manipulation, it would be desirable to have a semantics for our language $T(L)$. As a matter of fact, it is easy to provide a rigorous semantics to the new language by combining a semantics of first-order bringing-it-about [18,1] with the semantics for Until-Since logic, by means of Finger and Gabbay's *temporalizations* [15]. We can restrict the class of all models to the constraints for which (us0), (us1) and (us2) are canonical, and obtain a *class of models for tool manipulation*.

References

[1] H. L. Arló-Costa and E. Pacuit. First-order classical modal logic. *Studia Logica*, 84(2):171–210, 2006.

[2] R. Arp and B. Smith. Function, Role, and Disposition in Basic Formal Ontology. In *Proceedings of Bio-Ontologies Workshop (ISMB 2008)*, pages 45–48, 2008.

[3] N. Belnap, M. Perloff, and M. Xu. *Facing the Future (Agents and Choices in Our Indeterminist World)*. Oxford University Press, 2001.

[4] P. Bloom. Intention, history, and artifact concepts. *Cognition*, 60:1–29, 1996.

[5] S. Borgo, M. Carrara, P. Garbacz, and P. E. Vermaas. Formalizations of functions within the DOLCE ontology. In *Proc. of TMCE 2010*, pages 113–126, 2010.

[6] S. Borgo, M. Franssen, P. Garbacz, Y. Kitamura, R. Mizoguchi, and P. E. Vermaas. Technical artifact: An integrated perspective. In *Formal Ontologies Meet Industry, Proceedings of the Fifth International Workshop (FOMI 2011)*, volume 229 of *Frontiers in Artificial Intelligence and Applications*, pages 3–15. IOS Press, 2011.

[7] S. Borgo and C. Masolo. Ontological foundations of DOLCE. In S. Staab and R. Studer, editors, *Handbook on Ontologies (Second Edition)*, pages 361–382. Springer Verlag, 2009.

[8] S. Borgo and L. Vieu. Artefacts in formal ontology. In *Handbook of Philosophy of Technology and Engineering Sciences*, pages 273–308. Elsevier, 2009.

[9] E. Bottazzi and R. Ferrario. Preliminaries to a DOLCE Ontology of Organizations. *Journal of Business Process Integration and Management*, 4(4):225–238, 2009.

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

[11] J. Carmo and O. Pacheco. Deontic and action logics for organized collective agency modeled through institutionalized agents and roles. *Fund. Inform.*, 48:129–163, 2001.

[12] B. Chandrasekaran and J. R. Josephson. Function in Device Representation. *Engineering with Computers*, 16(3-4):162–177, 2000.

[13] A. Clark and D. J. Chalmers. The extended mind. *Analysis*, 58(1):7–19, 1998.

[14] D. Elgesem. The modal logic of agency. *Nordic J. Philos. Logic*, 2(2), 1997.

[15] M. Finger and D. M. Gabbay. Adding a temporal dimension to a logic system. *Journal of Logic, Language and Information*, 1:203–233, 1992.

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

[17] P. Garbacz. The four dimensions of artifacts. In *Principles of Knowledge Representation and Reasoning: Proc. of the Ninth International Conference (KR2004)*, pages 289–299, 2004.

[18] G. Governatori and A. Rotolo. On the Axiomatisation of Elgesem's Logic of Agency and Ability. *Journal of Philosophical Logic*, 34:403–431, 2005.

[19] D. Grossi, J.-J. Ch. Meyer, and F. Dignum. The many faces of counts-as: A formal analysis of constitutive rules. *J. Applied Logic*, 6(2):192–217, 2008.

[20] N. Guarino. Artefactual Systems, Missing Components and Replaceability. In *Artefact Kinds: Ontology and the Human-Made World*, volume 365 of *Studies in Epistemology, Logic, Methodology, and Philosophy of Science*, pages 1101–1117. Synthese Library, 2013.

[21] A. Herzig, E. Lorini, and N. Troquard. A dynamic logic of institutional actions. In *Computational Logic in Multi-Agent Systems*, volume 6814 of *Lecture Notes in Computer Science*, pages 295–311. Springer Berlin Heidelberg, 2011.

[22] R. Hilpinen. Artifact. In E. N. Zalta, editor, *Stanford Encyclopedia of Philosophy*. CSLI, 1999–2011.

[23] W. Houkes and P. E. Vermaas. *Technical Functions (On the Use and Design of Artefacts)*, volume 1 of *Philosophy of Engineering and Technology*. Springer, 2010.

[24] J. A. W. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, 1968.

[25] S. Kanger and H. Kanger. Rights and Parliamentarism. *Theoria*, 32:85–115, 1966.

[26] Y. Kitamura, Y. Koji, and R. Mizoguchi. An ontological model of device function: Industrial deployment and lessons learned. *Applied Ontology*, 1(3–4):237–262, 2006.

[27] P. Kroes. *Technical Artefacts: Creations of Mind and Matter (A Philosophy of Engineering Design)*, volume 6 of *Philosophy of Engineering and Technology*. Springer, 2012.

[28] S. Levy. *In the Plex: How Google Thinks, Works and Shapes Our Lives*. Simon & Schuster, 2011.

[29] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari. Wonderweb deliverable d18. Technical report, ISTC-CNR, 2003.

[30] C. Masolo, L. Vieu, E. Bottazzi, C. Catenacci, R. Ferrario, A. Gangemi, and N. Guarino. Social roles and their descriptions. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004)*, pages 267–277, 2004.

[31] D. Porello and N. Troquard. A resource-sensitive logic of agency. In *Proc. of 21st European Conference on Artificial Intellitence (ECAI'14)*, 2014.

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

[33] F. Santos, A. Jones, and J. Carmo. Responsibility for Action in Organisations: a Formal Model. In G. Holmström-Hintikka and R. Tuomela, editors, *Contemporary Action Theory*, volume 1, pages 333–348. Kluwer, 1997.

[34] J. R. Searle. *The Construction of Social Reality*. Free Press, 1995.

[35] R. B. Stone and K. Wood. Development of a Functional Basis for Design. *Journal of Mechanical Design*, 122(4):359–380, 2000.

[36] N. Troquard. Reasoning about coalitional agency and ability in the logics of "bringing-it-about". *Autonomous Agents and Multi-Agent Systems*, 28(3):381–407, 2014.

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