

# TEMPORAL ENTITIES

## *Types, Tokens and Qualifications*

B. O. Akinkunmi

*Department of Computer Science, University of Ibadan, Ibadan, Nigeria*

**Keywords:** Reified Logic, Knowledge Representation, Event types, Event tokens, Ontology.

**Abstract:** Reified logics have been a major subject of interest in the knowledge representation community for well over twenty years, since over the years, the need to quantify and reason about propositional entities such as events and states among other temporal entities has grown. Galton had made it clear that one may either refer to types or tokens (instances) of such entities in the ontology. A clear tendency in the literature is to derive event tokens from event types by instantiating types with their times of occurrence. That tendency is exemplified by earlier token-reified logic. The problem with this approach is that it makes it difficult to distinguish between two different events of the same type happening at the same time. This is a major price that earlier logic paid for being a full-fledged logical theory. This paper presents an alternative way of deriving event tokens from event types which uses the concept of qualifications rather than use times of occurrence. A clear distinction is made between qualifications and the actual event tokens they help derive from event types. A qualification captures the peculiarities of an actual event token that are not part of the event type definitions. Our logic maintains both the advantage of being a full-fledged logic as well being able to add many qualifications to an event token.

## 1 INTRODUCTION

The main objective behind the invention of reified logics is to make it possible to reason about and quantify over certain propositions (referred to in this paper as entities), the way one would do with any other objects in domain of the logic. This objective becomes imperative in view of the fact that such entities are not timelessly true, and their truths must be associated with various time units. Such entities may be states such as “the light is on” or events such as “John danced with Mary”.

According to (Galton, 2006) there are a number of unresolved issues with reified logics. One such issue is what it really means for a logic to be reified. According to Galton, one view of reification is for propositional terms to be arguments to a truth predicate. A less stringent view is for such propositional term to be used as arguments to any relation. The key property for all reified logics is for the logic to enable individual propositions to be quantified over.

With respect to associating entities with time units, there are two major syntactic options. The first option used in (Allen, 1984)’s reified logic

syntactically assigns the status of terms to what ordinarily should be propositions. They can then be associated with time using predicates such as Occurs or Holds. As such in order to assert that “John is in London, Monday”, one would write:

Holds (in (john, london), monday)

The other option due to (Galton, 1991) and (Davidson, 1967) would represent the same assertion by introducing a new variable into the proposition, so that new facts about the entity can be added by making assertions about the variable. For this option the above example will be rendered:

$\exists e$ . In (john, london, e)  $\wedge$  Holds (e, monday)

According to (Galton, 1991), in the second example above, e is an instance or token of the property John is in London. As such an association was established that suggested that entity types can only be expressed by using Allen’s syntax, while entity tokens can only be expressed by Galton’s syntax.

An entity token is a particular instance of an entity, which takes place once, while a type is an intensional reference to the class of all entities, which by definition share the same basic attributes.

According to (Galton, 2008) a temporal entity “type is an abstract entity corresponding to a description under which may fall any number of distinct instances called tokens: types are universals whereas tokens are particulars”. For example consider the proposition “John is London”. A particular instance of the state of John being London is an entity token, while a reference to the type talks about the class of states of John being in London (without referring to the class membership, hence the use of the term intensional reference). In another example, an event of the type “John danced with Mary” refers to class of all events in which John dances with Mary, while a token of the event is a reference to a particular dance event involving John and Mary.

(Galton, 1991) had noted that entity tokens are needed in order to express causation, because a causation relation between two events is a relation between two event instances or tokens and not event types. On the other hand event types are needed when one needs to talk about the repetition of an event, as it is not possible to talk about the repetition of an event token (Akinkunmi, 2000, Akinkunmi and Osofisan, 2004). One therefore needs both entity tokens and entity types in the ontology. It has been argued (Akinkunmi, 2000) that nothing about both syntactic options suggested any commitments about the nature of entities being reified. Thus the author uses Galton’s syntax for a theory reifying both entity types and entity tokens.

(Vila and Reichgelt, 1996) had argued that (Galton, 1991) did not present a “full-fledged” token reified logic, but rather a set of schema for deriving a full-fledged token reified logic. In other words, in Galton’s theory axioms are treated as schemas such as  $\exists e. P(x, e) \wedge \text{Holds}(e, \text{time})$  in which P must be regarded as place-holders for actual state/event predicates such as *kill* or *kiss*, and x by appropriate objects from the world. *It is in this sense that Galton’s theory is not a full-fledged theory.* This is not the case for Allen’s theory.

Consequently, they had proposed a token reified logic, which instantiated entity types by adding time units to them. A major drawback of their proposal is the fact that their approach to instantiation threw away the major advantage of Galton’s Davidson inspired approach to reifying entity tokens, which is the possibility of adding a boundless number of qualifications to entity tokens. As a matter of fact, they were able to derive a full-fledged logical theory because of their syntactic choice and not because of their approach to instantiating entities.

In this paper we present a full-fledged reified theory that allows both entity tokens and types in its

ontology. We achieve this goal while maintaining the ability of entity tokens to have potentially boundless qualifications asserted about it. In doing this, we introduce the notion of qualification formally into the logical theory, such that the identity of an entity type and a qualification is enough to determine the identity of an entity token.

It is important to note here that the event types we refer to in this paper are basic temporal entity types. Basic entity types are minimal classes of entities to which a particular token may belong. We do not deal with super-classes. For example we are interested in the event type “John danced with Mary”, rather than “John danced with *somebody*” or for that matter “*Somebody* danced with Mary”. This way we rule out having to consider all the types to which a token may belong.

The major goal of this paper is to introduce the concept of qualification as a means of instantiating entity types, as opposed to the approach of instantiating entity (event/state) types with time as done by (Vila and Reichgelt, 1996) as well as (Bennett and Galton, 2004). Qualifications are needed in order to express the idea that two event tokens of the same type can be different in certain respects. One key question is this: how are qualifications different from temporal entity tokens?

The rest of the paper is organized as follows. Section 2 presents an overview of the various reified logics that have appeared in the literature. Subsequently, our reified logic is presented in section 3. We demonstrate the advantages of the logic over other reified logics by the use of examples.

## 2 TYPE AND TOKEN REIFICATION

(Galton, 1991) concluded from McDermott’s set theoretic semantics that both Allen and McDermott reified event and state types and not event and state tokens. This he criticized as being Platonist. He also criticized Allen’s representation of causation as not carrying the exact information that an event is caused by another. In the place of Allen’s reification, he proposed a representation that is based on (Davidson, 1967)’s approach to instantiating events.

Davidson had pointed out that the description of an actual event will have potentially “unbounded qualifications”. In this context qualifications refer to the many different facts about aspects of the

occurrence that may be included in the description. For example, if we knew that John killed some particular snake in an actual event, then one qualification of that event is the weapon used by John, which may be a stick or a gun. Since many such qualifications may arise for actual events, Davidson suggested reifying the event in such a way that any other qualifier for the event may be added. For example, the event John killed the snake may be represented by:

$$\exists e. \text{Kill}(\text{john}, \text{snake}, e)$$

As such a qualifier that asserts that he used a weapon like a gun, may be added with a function weapon applied to the event  $e$  thus:

$$\exists e. \text{Kill}(\text{john}, \text{snake}, e) \wedge \text{weapon}(e) = \text{gun}$$

Galton likened, Davidson's  $e$  term to Situational Calculus' term  $s$ . We believe this to be a more accurate comparison than Vila's likening of situation terms to time terms in the method of temporal arguments (MTA) (Haugh, 1987). This is because both situational terms and event terms are acted upon by potentially many functions in the original theory, which is not necessarily the case for time terms in MTA (In the case of situational terms the functions are fluents returning boolean values).

(Galton, 1991) reckoned that instantiation of events can be achieved by introducing Davidson style event variables. Thus, by Galton's proposal, to assert that Mary kissed John at noon, one would write:

$$\exists e. \text{Kiss}(\text{mary}, \text{john}, e) \wedge \text{Occurs}(e, \text{noon})$$

In the above formula,  $e$  is to be regarded as an event token. Galton claims that this might be viewed as a means of syntactically "unreifying" Allen's reified logic i.e. doing away with the need to treat formulae like  $\text{kiss}(\text{mary}, \text{john})$  as terms, as Allen did. He also notes that causation is easier to express in this new way. He claims that there is no loss of expressive power as a result of unreifying Allen's formulae in this way. Interestingly, (Allen, 1991), Allen and (Fergusson, 1994) and (Fergusson, 1995) have since used Davidson's instantiation technique in representing actual actions in a planning system. However the need to retain action types is realized, since it enables one to express the fact of an agent trying to carry out an action.

Galton also criticized the reification of what he referred to as "state types" in (Kowalski and Sergot, 1986)'s Event Calculus, EC. Kowalski and Sergot did reify event tokens and state types. For example the fact that person  $x$  travelled to place  $y$  is an event

token that initiates the state type of  $x$  being in place  $y$  is rendered in EC as:

$$\text{Travel}(x, y, e) \Rightarrow \text{Initiates}(e, \text{in}(x, y))$$

Galton would rather have the consequent part of the above rendered:

$$\exists s. (\text{Initiates}(e, s) \wedge \text{In}(x, y, s))$$

where  $s$  is a state token.

(Vila and Reichgelt, 1996) while agreeing with the need to admit event/state tokens as objects into a theory instead of types, criticized Galton's work on the basis of the fact that Galton did not actually define a full-fledged theory, but rather gave a set of schemas for generating a theory. This is particularly obvious in Galton's definition of event causation which goes thus:

$$\begin{aligned} \forall e_1, e_2. \text{Ecause}(e_1, e_2) &\Leftrightarrow \\ E(e_1) \wedge E'(e_2) \wedge \text{Occurs}(e_1, i_1) \wedge \text{Occurs}(e_2, \text{succ}(i_1)) \wedge \forall i, e. (E(e) \wedge \text{Occurs}(e, i) &\Rightarrow \exists e'. (E'(e') \\ \wedge \text{Occurs}(e, \text{succ}(i))) \end{aligned}$$

In this definition,  $E$  and  $E'$  are not actual predicates but placeholders for actual predicates. As such the above definition is some sort of schema and not an actual axiom. We note here that  $\text{succ}$  is a function returning time intervals, and that what is referred to as  $\text{succ}(i)$  is actually referred to as  $i+1$  by Galton, but the basic ideas are the same.

We believe this same accusation by Vila and Reichgelt, may be made against the result of Bacchus et al's work in unreifying Shoham's theory into MTA (i.e. Method of Temporal Arguments) formulae (Haugh, 1987). They observed rightly that nothing in Galton's theory prevents an event token from occurring at two different times. The reified theory presented in (Akinkunmi, 2000) demonstrates this oversight in Galton's proposal by using Davidson's syntax for reifying both event types and event tokens, and then using a specific logical axiom which rules out duplicated occurrences of individual tokens in order to clearly define the difference between types and tokens.

(Vila and Reichgelt, 1996) thus presented a full-fledged reified theory first order theory, with formally defined semantics. The formulae reified are assumed to be from a first-order. In the new theory, each  $n$ -place predicate of the initial logic becomes an  $n+2$  place function in the reified logic, the 2 additional sorts being time sorts. Hence a function  $f(x, y, t_1, t_2)$  returns a token of type  $f(x, y)$  which starts at time point  $t_1$  and ends at time point  $t_2$ . They also had 1-place predicates  $\text{HOLDS}$  and  $\text{OCCURS}$  which are similar in usage to Allen's  $\text{Holds}$  and

Occurs respectively. To state that John went swimming between 1500 hours and 1530 hours in this theory one writes:

OCCURS (swim (john, 1500, 1530))

In the above “swim” is a function, and 1500 and 1530 are time points.

There is a problem with this approach. For example, if we know that there are two raining events that took place between 2 and 3 pm, we would have no way of differentiating one from the other. In Vila and Reichgelt’s language, they are both the same: Rain (2, 3). If we know later that one took place in Lagos, Nigeria and the other took place over the Wimbledon centre court, we would simply have no way of differentiating one from the other in Vila and Reichgelt’s logic.

The logical theory presented in this paper, is a full-fledged reified theory with both entity tokens and types in its ontology. We have noted that Vila and Reichgelt were able to attain a full-fledged reified theory because of their adoption of a syntax that is akin to Allen’s own presented in section 1, over Davidson/Galton’s syntax. However, the theory presented here achieves instantiation of entity types by introducing qualifications. These qualifications are similar to the instantiation variables used by Galton. This makes it possible to add other qualifications to an entity. That is something precluded in Vila and Reichgelt’s theory. As such this theory finds a way of combining the advantages of a full-fledged theory made possible by adopting Allen’s syntax, with those of a theory in which one can add new information about a reified entity, made possible by adopting Davidson style individuation of entities. It must be stressed however that qualifications are completely different from event tokens, in the sense that qualifications only capture the peculiarities of each event token that are not part of event type definition. This will become clear from in the next section.

### 3 THE LOGICAL THEORY

Now we present our expressive reified theory, which uses Allen’s syntax and allows both tokens and types in its ontology. Our theory contrasts Vila and (Reichgelt, 1996) and (Bennett and Galton, 2004) in that while those authors take event tokens to derived from event types and the time of occurrence, while we take our temporal entity tokens to be derived from temporal entity types and qualifications.

### 3.1 Language and Notation

The logic presented is a many sorted first order logic, with the sorts entity types  $E_T$ , entity tokens  $E_{TK}$ , time intervals  $Int$  domain entities  $D$ , and qualifications  $Q$ . We define as  $n$  place functions all  $n$ - place predicates that define events or states in the initial logic to be reified. These functions return elements of the sort  $E_T$ . In addition to these we have an instantiation function  $f_i$  taking as sorts an entity type and a qualification, and returning an entity token. The functions are formally introduced thus:

$p: D^n \rightarrow E_T$  (where  $p$  is an  $n$ -place predicate in the language to be reified).

$f_i: E_T \times Q \rightarrow E_{TK}$

type:  $E_{TK} \rightarrow E_T$

We need to clarify here that temporal entity types that we deal with here are basic temporal entity types only. In this case, any temporal entity token can only be of one basic entity type. Basic entity types are similar to basic event types in Kautz’s event abstraction hierarchy (Kautz, 1987). Like (Bennett and Galton, 2004), we regard any two tokens happening at the same time as not being necessarily connected in any way.

It is possible to express the idea of the trial of an entity type. We can say that an agent tried to achieve an event type (and not an event token). For this purpose, we introduce a function *try* which maps a pair of entity token and qualification into an event token.

$try: E_T \times Q \rightarrow E_{TK}$

A qualification is the means by which one may know things that are peculiar about an event token which are not necessarily shared by event tokens of the same type. These peculiarities may be so many that they cannot all be captured by event type definitions. However, a qualification is entirely different from the token it defines, as one cannot rule out the possibility of two different event tokens sharing the same qualifications. Examples of this are presented in section 3.1.1.

Assertions about peculiarities of an event token can be made by propositional assertions about its qualification. For example if we knew of an event token of the type “john killed the snake”. Some assertion can be made about the qualification regarding the *place of event* and *weapon* by introducing predicates *Weapon* and *Place*.

Weapon (q, stick)

Place (q, under-the-oak)

We also introduce the function *succ* which maps the time of an event (a cause) to the time of its effect thus:

$$\text{succ}: E_{TK} \times Int \rightarrow Int$$

With this we are assuming that an event can only have one effect. In order to allow multiple effects, *succ* must return subsets of the Cartesian product  $I \times E_{TK}$ .

$$\begin{aligned} T: E_T \times Int &\rightarrow \text{Boolean} \\ T_K: E_{TK} \times Int &\rightarrow \text{Boolean} \end{aligned}$$

T denotes the truth of an entity type over a time interval, while  $T_K$  denotes the truth of an entity token over an interval. We are also using some of Allen's interval relations:

$$\text{After, Overlaps, Meets etc.}: Int \times Int \rightarrow \text{Boolean}$$

In addition to these we introduce the Cause relation, which is the causal relation between an event token and its effect, which is also a token.

$$\text{Cause}: E_{TK} \times E_{TK} \rightarrow \text{Boolean}$$

As a notation, we assume that the symbols, i, j, k, l, m, n, p, q with or without suffixes are time intervals, while e with or without suffixes, refer to entity tokens. The symbols x, y, z with or without suffixes, are used for entity types.

### 3.1.1 Examples

We now present some examples that demonstrate the expressiveness of our logic, as well as its advantage over some existing reified theories, particularly over the logic of (Vila and Reichgelt, 1996). For the sake of distinguishing between predicates and functions for these examples, we write functions in italics. These examples among others demonstrate the usefulness of having qualification variables.

**Example 1:** *Osuofia danced with Adaobi for an hour at noon.*

$$\exists e, q. e = f_i(\text{dance\_with}(\text{Osuofia}, \text{Adaobi}), q) \wedge T_K(e, 1200-1300)$$

We note here that *dance\_with* is treated here as a function, because as explained in section 3.1, we treat the predicate to be reified as a function. In Vila and Reichgelt's logic this was also the case. However, in Vila and Reichgelt's logic this will be rendered:

$$\text{OCCURS}(\text{dance\_with}(\text{Osuofia}, \text{Adaobi}), 1200, 1300))$$

For this example, our new logic is no less expressive than Vila and Reichgelt's.

**Example 1' :** *Osuofia danced with Adaobi for an hour at noon. The kind of dance was polka. It happened at the Ritz.*

$$\exists e, q. e = f_i(\text{dance\_with}(\text{Osuofia}, \text{Adaobi}), q) \wedge T_K(e, 1200-1300) \wedge \text{Kind}(q, \text{polka}) \wedge \text{Place}(q, \text{ritz})$$

The fact that qualifications are to be added would pose a challenge for Vila and Reichgelt's logic. The best one can do to say that the kind of dance is polka and that the dance took place at the Ritz in Vila and Reichgelt's logic would be to write:

$$\text{Kind}(\text{dance\_with}(\text{Osuofia}, \text{Adaobi}), 1200, 1300)) = \text{polka} \wedge \text{place}(\text{dance\_with}(\text{Osuofia}, \text{Adaobi}), 1200, 1300)) = \text{ritz}$$

However, there would be nothing in the token to distinguish it from another token involving the same persons at the same time, but if the kind of dance had been bata and not polka. Although it is not likely that Osuofia is and Adaobi are engaged in another dance at the same time, but in general it is possible to distinguish between one instance of an entity type and another that takes place at the same time. The next example demonstrates this.

It is important to note here that *place* and *kind* are both qualification functions giving the kind of dance and place of dance.

**Example 2:** *Osuofia tried for five minutes to get Adaobi to dance at noon.*

$$\exists e, q. e = \text{Try}(\text{dance}(\text{Adaobi}), q) \wedge \text{agent}(q) = \text{Osuofia} \wedge T_K(e, 1200-1205)$$

We note here that the function *agent* a qualification function like  $f_{Q1}, f_{Q2}...$ etc.

The best one can do in Vila and Reichgelt's logic is to express the trial incident and the fact that Osuofia was the agent as:

$$\begin{aligned} &\text{OCCURS}(\text{try}(\text{dance}(\text{Adaobi})), 1200, 1205) \\ &\wedge \text{Agent}(\text{try}(\text{dance}(\text{Adaobi})), 1200, 1205) = \text{Osuofia} \end{aligned}$$

However, there would be nothing in the event token that would make it different from another trial event involving Adaobi and happening at the same time whose agent is someone else. In other words if another person Adaeze, was trying to make Adaobi dance at the same time as Osuofia, the same token  $\text{try}(\text{dance}(\text{Adaobi}), 1200, 1205)$  would refer to the two different event tokens in Vila and Reichgelt's language.

The next two examples are adaptations of examples from (Vila and Reichgelt, 1996). This demonstrates that our language is no less expressive.

**Example 3:** *When a cause occurs, its effect holds.*

$$\forall e_1, e_2. T_K(e_1, j) \wedge \text{Cause}(e_1, e_2) \Rightarrow T_K(e_2, \text{succ}(e_1, j))$$

**Example 4:** When Lagbaja dances with someone it makes them tired.

$$\forall x, e_1, q_1. \exists e_2, q_2. e_1 = f_1(\text{dance-with}(\text{lagbaja}, x), q_1) \wedge e_2 = f_1(\text{tired}(x), q_2) \wedge \text{Cause}(e_1, e_2)$$

However when it becomes necessary to add a qualifier to the kind of dance that makes a person dancing with Lagbaja tired, Vila and Reichgelt's logic fails as the following example demonstrates.

**Example 4':** Dancing with Lagbaja gets one tired. This happens if the dance is bata.

$$\forall x, e_1, q_1. \exists e_2, q_2. e_1 = f_1(\text{dance-with}(\text{lagbaja}, x), q_1) \wedge e_2 = f_1(\text{tired}(x), q_2) \wedge \text{Cause}(e_1, e_2) \wedge \text{Kind}(q_1, \text{bata})$$

In Vila and Reichgelt's language the best one can do to achieve such a qualification is to have such qualification functions such as *kind* and then write:

$$\text{Kind}(\text{dance-with}(\text{lagbaja}, x, t_1, t_2), \text{bata})$$

However as we have argued before in the example involving the raining example, there would be nothing in the event token to distinguish it from a dance involving the same individuals at the same time, if the kind of dance was salsa and not bata.

**Example 5:** Causes precede their effects.

$$\forall e_1, e_2. \text{Cause}(e_1, e_2) \Rightarrow \forall j. T_K(e_1, j) \Rightarrow T_K(e_2, \text{succ}(e_1, j)) \wedge (\text{After}(\text{succ}(e_1, j), j) \vee \text{Overlaps}(j, \text{succ}(e_1, j)) \vee \text{Meets}(j, \text{succ}(e_1, j)))$$

From the above examples it should be clear that our logical theory is a full-fledged one unlike (Galton, 1991). It follows from the pairs of examples 1, 1' and 4, 4' that it supports incremental knowledge representation on the account of allowing unbounded qualifications for entities. We note that the ease with which one states that "causes precede their effects" in example 5 is the same as in (Vila and Reichgelt, 1996).

Finally examples 6 and 7 illustrate the idea that qualifications are not in any way attached to event types. As such event tokens of different types may share the same qualifications.

**Example 6:** Tarzan killed the lion in exactly the same way in which he killed the leopard.

$$\exists q_1, q_2. f_1(\text{kill}(\text{tarzan}, \text{lion}), q_1) \wedge f_1(\text{kill}(\text{tarzan}, \text{leopard}), q_2) \wedge q_1 = q_2$$

**Example 7:** Lola did her laundry and washed the car on Saturday. She did everything in the same sluggishly manner.

$$\exists q. e_1 = f_1(\text{laundry}(\text{lola}), q) \wedge e_2 = f_1(\text{wash}(\text{lola}, \text{car11}), q) \wedge T_K(e_1, \text{Saturday}) \wedge T_K(e_2, \text{Saturday}) \wedge \text{Manner}(q, \text{sluggish})$$

## 4 SUMMARY AND CONCLUSIONS

In this paper we have presented a new approach to deriving event tokens from event types by introducing the concept of qualification. This is different from the approach in the literature that derives event tokens from event types and times of occurrence. We have stressed that the latter approach has the disadvantage of making it difficult to distinguish between two events of the same type happening at the same time. This becomes more evident when one needs to add new information about the peculiarities of one of the two events.

One must stress again that the only similarity between the event tokens in Galton's logic (Galton, 1991) and the qualifications introduced here is that they are both variables. However from the examples qualifications are clearly different from tokens in the sense that qualifications only capture the peculiarities of an actual event token.

What is left is perhaps to present a clear formal semantics for this logic.

## ACKNOWLEDGEMENTS

The author is grateful to the referees for their helpful comments and to Solomon Akinola for his editorial assistance.

## REFERENCES

- Akinkunmi, B. O. (2000). On the expressive limits of Reified logics. *Journal of Logic and Computation*, 10(2), pp. 297- 313, April 2000.
- Allen, J. F. (1991), Planning as Temporal Reasoning. In J.F. Allen, R. Fikes and E. Sandewall (eds.) *Proceedings of the 2<sup>nd</sup> International Conference on Principles of Knowledge Representation and Reasoning (KRR 91)*, pp. 3-14, Morgan Kaufmann, San Francisco, May 1991.
- Allen, J. F. (1984), Towards a general theory of action and time. *Artificial Intelligence*, 23: 123-154, 1984.
- Allen, J. F. (1983), Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26: 832-843, 1983.

- Allen, J. F. & Fergussion, G. M. (1994). Actions and Events in Interval Temporal Logic. *Journal of Logic and Computation* 4(5): 531-579, December 1994.
- Allen, J. F. & Hayes, P. J. (1985). A common sense theory of time. In R. Bajcsy (ed.), *Proceedings of the 9<sup>th</sup> IJCAI, 1985*, pp 528-531 Morgan Kaufmann Publishers, 1985.
- Bacchus F., Tenenbergs J. & Koomen, J. A. (1991). A non-reified temporal logic. *Artificial Intelligence* 52 (1) pp. 87-101, 1991.
- Davidson, D. (1967). The logical form of action sentences. In Rescher, N. (ed.) *The logic of decision and action*. pp 81-95, University of Pittsburgh Press, 1967. Also in Davidson, D. (ed.) *Essays on Actions and Events* pp 105-221, Oxford University Press, Oxford 2001.
- Fergussion, G. M. (1995). *Knowledge Representation and Reasoning for Mixed-Initiative Planning*. PhD thesis, University of Rochester, Rochester, New York, 1995.
- Galton A. (2008) Experience and History: Processes and their Relation to Events. *Journal of Logic and Computation* 18(3): 323-340, December 2008.
- Galton A. (2006) Operators vs Arguments: The Ins and Outs of Reification. *Synthese* 150: 415-441
- Bennett, B and A. P. Galton (2004) A unifying semantics for time and events. *Artificial Intelligence* 153(1-2): 13-48
- Galton, A. (1991). Reified Theories and how to unreify them. In Mylopoulos, J. and Reiter, R (eds.) *Proceedings of the IJCAI, 1991*, pp.1177-1182, Morgan Kaufmann Publishers, San Francisco, 1991.
- Goody, J. (1994). *A transition based approach to reasoning about action and change*. PhD thesis, Department of Computer Science, University of Exeter, September, 1994
- Haugh, B. A. (1987). Non-standard semantics for the method of temporal arguments In J. Mylopoulos (ed.) *Proceedings of the 10<sup>th</sup> International Joint Conference on Artificial Intelligence*, pp.449-455, Morgan Kaufmann, San Francisco, 1987.
- Kautz, H. A. (1987). *A Formal Theory of Plan Recognition*. PhD thesis, University of Rochester, Rochester, New York.
- Koomen, J. A. (1989). *Reasoning about Recurrence* PhD thesis, University of Rochester, Rochester, New York.
- Kowalski, R. & Sergot, M. (1986). A logic based calculus of events. *New Generation Computing* 4(1): 67-96, April 1986.
- Shoham, Y. (1987). Temporal logics: semantical and ontological considerations. *Artificial Intelligence*, 33: 89-104, 1987.
- Vila, Ll. & Reichgelt, H. (1996). The token reification approach to temporal reasoning *Artificial Intelligence*, 83(1): 59-74, 1996.