A Versatile Representation for Time and Events

Brandon Bennett*
School of Computing
University of Leeds, Leeds LS2 9JT, UK
brandon@comp.leeds.ac.uk

Antony P. Galton
Department of Computer Science
University of Exeter, Exeter EX4 4PT, UK
antony@dcs.exeter.ac.uk

April 20, 2001

Abstract

The paper gives an overview of a highly expressive language for representing temporal relationships and events. This language, which we call *Versatile Event Logic* (VEL), provides a general temporal ontology encompassing many other representations.

1 Foreword

This paper is an abbreviation of a much longer technical report (Bennett and Galton 2000). This shorter version gives the motivation, the syntax and an informal account of the semantics of the language VEL. The full report gives a fully formal semantics and a set of axioms for reasoning with the language.

2 Introduction

Many researchers in the field of AI have recognised the need for formal representation languages capable of expressing high-level information in a naturalistic form — e.g. (Hayes 1979, Davis 1990). Of central importance to such a representation is a precise analysis of the semantics of actions, events and temporal relations. Among the most influential of the large number of formalisms proposed to deal with these are: the *Situation Calculus* (McCarthy and Hayes 1969), Allen's theory of action and time (Allen 1984) and the *Event Calculus* (Kowalski and Sergot 1986, Shanahan 1999). Philosophers have also examined various logical aspects of events and we aim to take into account in particular the analyses given by Davidson (1980) and Galton (1984). We present a unifying language that we call *Versatile Event Logic* (VEL) embodying a range of insights into the logic of time and events.

Representations of time within formal languages can be readily divided into three main approaches:

- Explicit reference to time points and a temporal ordering relation.
- Explicit reference to temporal intervals and the temporal relationships between them (which implicitly constrain the ordering of the end-points of the intervals).
- Use of propositional tenses to convey temporal relationships among facts (without explicit reference to any temporal entities).

Whereas most temporal logics incorporate only one of these ways of referring to temporal structure, in our formalism all three of them happily coexist. A single VEL formula can contain time point and interval variables as well as tenses (though it turns out, unsurprisingly, that all of these constructs are in fact reducible to logically equivalent expressions involving just time points and the temporal ordering relation).

Our formalism incorporates each of the following analyses:

*Supported by the EPSRC under grant GR/M56807.

Events as Transitions: A view of events that can be found in some of the earliest AI systems is that an event is a transition between states. This idea is the basis of STRIPS (Fikes and Nilsson 1971) and also the *Situation Calculus* (McCarthy and Hayes 1969). The transition account of events can be handled very nicely within the general framework of modal logics, where a class of state transitions can be regarded as an accessibility relation. This idea has been used to provide the semantics for *dynamic logic* (Harel 1984).

Events as Occurrences Over Intervals: An approach widely adopted in AI is to correlate events with the time intervals over which they occur (Allen 1984, Galton 1993, Galton 1997). This is done by means of a quasi-logical predicate $\mathsf{Occurs}(\mathbf{e}, \delta)$, which says that an event of type $\mathbf{e}$ occurs during the interval δ.

Davidson's Existential Analysis: Davidson (1967) gave an analysis of events according to which every action verb is associated with an implicit existentially quantified event token variable. For example the sentence 'John saw Mary in London on Tuesday' would have the logical form

$$\exists\varepsilon[\mathsf{Saw}(\varepsilon, \mathsf{j}, \mathsf{m}) \wedge \mathsf{In}(\varepsilon, \mathsf{l}) \wedge \mathsf{On}(\varepsilon, \mathsf{t})] \ .$$

This explains how a large number of modifying phrases may be attached to verbs and also accounts for a wide range of inferences concerning verbs and their modifiers. For instance from the above example we can infer $\exists\varepsilon[\mathsf{Saw}(\varepsilon, \mathsf{j}, \mathsf{m})]$ by straightforward 1st-order reasoning.

Although Davidson's work is not very well known to AI researchers, the logical analysis it provides is very close to that employed by a wide variety of AI representations based on *reified* events. In particular, semantic networks (Lehmann 1992) typically treat an event as a special kind of entity. Other entities are linked to the event by arcs describing their relationship to the event. The main difference between this approach and Davidson's analysis is that the fundamental rôles of agent and patient (and perhaps also indirect object) are separated out from the verb rather than treated as integral. Thus, many AI representations would exhibit a logical form such as

$$\exists\varepsilon[\mathsf{Saw}(\varepsilon) \wedge \mathsf{Agent}(\varepsilon, \mathsf{j}) \wedge \mathsf{Patient}(\varepsilon, \mathsf{m}) \wedge \\ \mathsf{In}(\varepsilon, \mathsf{l}) \wedge \mathsf{On}(\varepsilon, \mathsf{t})] \ .$$

The difference between this and Davidson's formulation does not seem to have any concrete semantic significance and in developing the semantics we have found it more intuitive to use treat verbs as relation-like with a small fixed arity, as does Davidson. However, as will be explained below, we do not treat the event variable as a hidden argument of the verb but as bound to the verb by means of a kind of quantifier.

The ability to refer directly to individual event instances is undoubtably useful for representing certain kinds of knowledge. For instance it enables one to represent adverbs straightforwardly as predicates over events and to describe causality as a binary relation between events.

Event Radicals: Galton (1984, 1987) analyses events in terms of *event radicals*, syntactic units which refer to event types and combine with temporal *aspect* operators to form propositions. For example, if the event type $\mathsf{John\text{-}tie\text{-}his\text{-}shoelace}$ were operated on by the *progressive* aspect, we would obtain the proposition $\mathit{Prog}(\mathsf{John\text{-}tie\text{-}his\text{-}shoelace})$, meaning that en event of type '$\mathsf{John\text{-}tie\text{-}his\text{-}shoelace}$' is in progress. We would read this as 'John is tying his shoelace'.

We want to encompass the essential insights of all these different approaches within a single semantic framework. Our paper may be regarded as an attempt to produce a detailed *ontology* (as advocated by e.g. (Guarino 1998)) of time and events accounting for their various linguistic manifestations. Whereas much recent work on ontology for AI systems has been carried out at the level of axioms (e.g. (Lenat 1995)), ours is founded at the semantic level. We do give axioms governing the behaviour of our operators; but formulating a complete proof system is a subject for further research.

It must be emphasised that the VEL formalism is *not* intended to contribute directly to solving the problem of how to *reason* about actions and change. This problem involves many factors that are completely beyond the scope of this work. In particular we do not address the much studied *frame problem* or any of its sub-problems, which have been investigated in considerable detail within the AI literature (see e.g. (Hayes 1973, Hanks and McDermott 1986, Brown 1987, Baker 1991, Sandewall 1994, Shanahan 1997)). VEL gives us a very general language for talking about time and events but is completely agnostic about the workings of phenomena such as causality and inertia that determine what sequences of events are be possible or likely.[1]

It must be further acknowledged that even in terms of expressive power there is little that can be said within VEL that could not be said by means of some other existing formalism. However, the contribution of VEL is to put everything together.

3 Temporal Ontology

The semantics of our event logic must include a structure that determines relationships between different times. The simplest model treats time as a set of linearly ordered points. This set may be further constrained to be either discrete, dense or continuous and either bounded or unbounded in each direction (or might satisfy more esoteric conditions). If we want to be able to employ modal operators to distinguish between analytic and contingent propositions, we shall need a richer structure in which there are possible states of the world which do not occur at any point in the actual history of the world. We can think of the set of all possible histories of the world as forming a branching tree structure. A number of possible semantics and axiom sets based on such structures are discussed in (Thomason 1984). Branching structures have also been proposed as a basis for AI reasoning systems (McDermott 1982, Shoham 1988) but this approach does not seem to have been widely adopted. However, in the theory of computation, branching time logics, such as CTL* (Emerson and Halpern 1986) have been extensively studied (Moller and Birtwistle 1996, Moller and Rabinovich 1999).

3.1 History Trees

To model the temporal order of states of an evolving universe and the relationship between possible alternatives to the actual history of the world we employ a structure that we call a *History Tree*. This is a tuple $\mathcal{T} = \langle S, T, \prec, H \rangle$, where:

- S is a set of world states,
- T is a set of times,
- $\prec$ is an irreflexive linear order on T (which may be further constrained to be dense, continuous etc.),
- H is a set $\{h_i\}$ of histories, each of which is a function from T to S.

The history functions in H are further constrained relative to the time structure by the following conditions:

SconfP $(\forall h_1, h_2 \in H)[(h_1 = h_2) \vee (\exists t \in T)(\forall t' \in T)[t' \preceq t \leftrightarrow h_1(t') = h_2(t')]\ ]$

Gclock $(\forall h_1, h_2 \in H)(\forall t_1, t_2 \in T)[h_1(t_1) = h_2(t_2) \rightarrow (t_1 = t_2)]$

A pair $\langle h_i, t_i \rangle$ (where $h_i \in H$ and $t_i \in T$) is the *index* of a possible world in the tree, and $h_i(t_i)$ denotes the world state that holds at that index. Each state is associated with a set of *entities* N_s. An *interval* $[t_1 ... t_2]$ is the set $\{t \mid t \in T \wedge t_1 \preceq t \preceq t_2\}$ of time points.

SconfP gives strict confluence in the past: every two distinct histories agree up to and including some time point and after that point they never coincide. **Gclock** tells us that we cannot have the same state at two distinct time points (there is a 'global clock'). The confluence property induces an equivalence relation on histories with respect to any given time point. We define

$$h_1 \overset{t}{\approx} h_2 \quad \equiv_{def} \quad (\forall t' \in T)[(t' \preceq t) \rightarrow h_1(t') = h_2(t')] \ ,$$

which can be read as 'h_1 and h_2 are similar up to t'.

This model of branching time has significant advantages over other approaches. In particular: a) on each history the time structure is linear and can be described by a relatively simple logic; b) reasoning about alternative histories can be handled by an $S5$ modality.

3.2 Entities and Individuals

In standard 1st-order logic names are taken as basic expressions denoting *individuals*, which are taken as satisfying a fixed set of predicates. However, in a temporal language in which one can describe events and changes one must be able to refer to individuals which persist through time although their properties may change. In the context of our history tree model of time, we stipulate that an individual is *realised* in each world state (in which it exists) by an *en-*

[1] It can be argued that any language which is sufficiently expressive to describe a variety of different possible models of causation must itself be to some degree neutral about the nature of causation.

tity. These entities correspond to 'time slices' of individuals located within the branching history structure. Thus an individual is modelled by a function from states to entities. At those worlds where an individual does not exists it denotes the value $\emptyset$. If at some state s two individuals denote the same entity we say they are *co-realised* at s.

Formally, the time structure together with the individuals which inhabit it is modelled by a VEL *frame*. This is a structure: $\mathcal{F} = \langle \mathcal{T}, N, I \rangle$ where

- $\mathcal{T} = \langle S, T, \prec, H \rangle$ is a history tree.
- $N = \bigcup_{s \in S} N_s$ is a set of *entities*, partitioned into subsets N_s existing in each state.
- I is a set of individuals ι_i, each of which is a function from S to $N \cup \{\emptyset\}$, such that $(\forall s \in S)[\iota_i(s) \in N_s \cup \{\emptyset\}]$.

Let $I(\mathcal{F})$ denote the set of individuals in $\mathcal{F}$.

3.3 Episodes

In specifying the semantics for events we shall need to refer to the interval in which an event occurs; however, an interval $[s...f]$ alone is not enough to locate the event within the branches of the structure: we must also take into account the history on which it occurs. This leads us to consider a pair $\langle h, [s...f] \rangle$. We now observe that for any history h' similar to h up to f, the same sequence of states, and hence the same event, occurs. Therefore we define the *episode* $[s...f]_h^{\approx}$ (where $s \preceq f$) by:

$$[s...f]_h^{\approx} = \{\langle h', [s...f] \rangle \mid h' \in H \wedge h' \overset{f}{\approx} h\} .$$

$\mathbf{Ep}(\mathcal{F})$ denotes the set of all episodes on frame $\mathcal{F}$.

Episodes are not themselves sets of times so we do not distinguish whether they are open or closed. However, the open/closed distinction becomes important when we consider the time points over which a proposition is true. A proposition might be true/false at all points within an episode, or at either of the end points of the episode, or any combination of these.

3.4 States and Events

A world state $h(t) \in S$ determines all those properties of the world at time t in history h which are invariant across all the histories similar to h up to t. We call these the properties which are *settled* at t in h. $h_1(t_1) = h_2(t_2)$ does not mean that the same propositions are true at $\langle h_1, t_1 \rangle$ and $\langle h_2, t_2 \rangle$ because the truth of some propositions depends upon the future, which depends on the actual history.

A *partial state* is a subset $S' \subseteq S$. It can be used to characterise a property shared across a number of states. This corresponds to what we often mean by a "state", e.g., "John is asleep". Being sets, partial states are closed under the standard Boolean operations, e.g., union, intersection and complement.

An *event type* is a set of episodes. This may be specified in various ways; one important way of specifying event types is in terms of partial states. For, example, given the partial state $S' \subseteq S$, the event type

$$\begin{aligned}\{[t..t]_h^{\approx} \mid (\exists t_1, t_2)[(t_1 \prec t \prec t_2) \wedge \\ (\forall t')[((t_1 \prec t' \prec t) \rightarrow h(t') \notin S') \wedge \\ ((t \prec t' \prec t_2) \rightarrow h(t') \in S')]]\}\end{aligned}$$

picks out *inceptions* of partial state S', i.e., transitions between an interval over which S' does not hold and an interval over which it does.

Event types can be classified in many different ways. Popular categories include:

- *punctual* event types are those all of whose constituent episodes are of the form $[t..t]_h^{\approx}$
- *durative* event types have all episodes of the form $[s..f]$ where $s \prec f$
- *unitary* (or *solid*) event types have the property that they never contain two episodes such that one is a proper sub-episode of the other.

3.5 Event Tokens

Event tokens are occurrences of event types. This being so it is tempting to regard event types as predicates over a domain of event types. Indeed this is the essence of Davidson's account. However, a naïve development of this idea can easily lead to contradictions. These are most apparent when we try to formulate the conditions under which an event token can be an occurrence of more than one event type.

Consider the types: $\mathbf{l}$ = *John-stand-on-one-leg*, $\mathbf{a}$ = *John-eat-an-apple*, $\mathbf{la}$ = *John-stand-on-one-leg-while-eat-apple*. On the one hand it seems that $\mathbf{l}$ and $\mathbf{a}$ are different kinds of event, since a token of **l** might cause John's leg to become stiff, whereas a token of **a** could not. so there can be no token that is a token of both these types. If the two event types apply to the same sequence of world states then this just means that two different tokens occur simultaneously. However, on the other hand, it seems reasonable that a token of type **la** is also a token of **l** and a token of **a**, since these are just less specific descriptions of a particular occurrence described by **la**.

We believe that these intuitions about the token/type relationship cannot be reconciled as long as we divorce event tokens from their types. However, by treating an event token ε as a pair $\langle \eta, \mathbf{e} \rangle$, where $\mathbf{e}$ is an event type and η a specific episode of that type, much mystery can be diffused. An event token can be regarded an episode seen from a certain perspective.

3.6 Participation of Individuals in Events

When an individual takes part in an event described by some verb, it goes through a series of changes that satisfy those conditions that constitute the meaning of the verb. In our semantics the changing properties of individuals are modelled by them denoting different entities at different states along histories in a VEL frame. We can thus associate an occurrence of a verb with the sequences of entities corresponding to participating individuals along the episode over which the event takes place. Formally, a verb occurrence is then a pair

$$\langle\, [t_s...t_f]_h^{\approx},\ \nu : [t_s...t_f] \rightarrow N^n \rangle\ ,$$

where ν satisfies

$$(\forall t \in [t_s...t_f])[\nu(t) \in (N_{h(t)})^n]\ ,$$

i.e. $\nu(t)$ must be a tuple drawn from those entities that exist in the state $h(t)$.

We identify each verb with a set of verb occurrences. The set of possible denotations for an n-ary verb in an VEL frame $\mathcal{F}$ will be denoted by $\mathbf{V}_n(\mathcal{F})$.

To relate verb occurrences to individuals, we define an operator which projects a tuple of individuals onto an episode, so as to obtain a mapping from the points on that interval to tuples of entities:

$$(\langle \iota_1, \ldots, \iota_n \rangle {\downarrow} [t_s...t_f]_h^{\approx}) \ \equiv_{def}\ \nu : [t_s...t_f] \rightarrow N^n$$

such that

$$(\forall t \in [t_s...t_f])[\nu(t) = \langle \iota_1(h(t)), \ldots, \iota_n(h(t)) \rangle]\ .$$

We can now say that a tuple of individuals $\langle \iota_1, \ldots, \iota_n \rangle$ participates in a verb occurrence $\langle \eta, \nu \rangle$ just in case

$$(\langle \iota_1, \ldots, \iota_n \rangle {\downarrow} \eta) = \nu\ .$$

Finally we can model the operation of a verb with occurrence set V upon a tuple of individuals $\langle \iota_1, \ldots, \iota_n \rangle$ to produce an event type $\mathbf{e}$ by the following semantic equation:

$$\mathbf{e} = \{\eta \mid \langle \eta, \nu \rangle \in V \wedge (\langle \iota_1, \ldots, \iota_n \rangle {\downarrow} \eta) = \nu\}\ .$$

4 Syntax

We now specify the syntax of VEL. The language has variables of six *nominal types*, $\sigma_i, \sigma_t, \sigma_\delta,\ \sigma_\varepsilon,\ \sigma_s, \sigma_o$, which can be quantified over. Any variable that is not bound by a quantifier behaves as a constant. The intended meanings and vocabulary sets of these types of variables are respectively:

- Individuals, $\mathcal{V}_i = \{a, b, c, \ldots\}$
- Times, $\mathcal{V}_t = \{t_i\}$
- Intervals, $\mathcal{V}_\delta = \{\delta_i\}$
- Event Tokens, $\mathcal{V}_\varepsilon = \{\varepsilon_i\}$
- States, $\mathcal{V}_s = \{s_i\}$
- Observable Values, $\mathcal{V}_o = \{o_i\}$

In many cases quantification over times, intervals, events and world states can be avoided by the use of tense and aspect operators, modal operators and various other constructs. For practical applications one might employ sublanguages of VEL that allow little or no quantification.

Four other types of non-logical constant are not quantified over:[2]

- Count nouns, $\mathcal{V}_c = \{\mathrm{C}_i\}$
- Relations, $\mathcal{V}_R = \{A, B, C, \ldots\}$
- Functions, $\mathcal{V}_f = \{f_i\}$
- Verbs, $\mathcal{V}_v = \{\mathbf{v}_i\}$

A VEL vocabulary is thus a 10-tuple $\mathcal{V} = \langle \mathcal{V}_i, \mathcal{V}_t, \mathcal{V}_\delta, \mathcal{V}_\varepsilon, \mathcal{V}_s, \mathcal{V}_o, \mathcal{V}_c, \mathcal{V}_R, \mathcal{V}_f, \mathcal{V}_v \rangle$. The sets $\mathcal{V}_R$, $\mathcal{V}_o$ and $\mathcal{V}_v$ are partitioned into sub-classes according to the numbers and types of arguments that these expressions can combine with.

[2] In principle we could also allow quantification over these; but we would then immediately get a second order language which would be highly intractable and most probably unaxiomatisable.

It may seem that the diversity of syntactic types in VEL is excessive; but the idea of VEL is to be inclusive and to incorporate as many as possible of the kinds of expression that have been employed in terporal representations. VEL is not intended to provide a practical logic but rather a comprehensive umbrella language into which many other formalisms can be translated.

Terms: A term is either: a variable, or a functional term $f(\beta_1, \ldots, \beta_n)$, where $\beta_1, \ldots, \beta_n$ are terms; or the logical symbol null (of type σ_i). Functions come in a variety of different types according to the nominal types they accept as arguments and the nominal type of the resulting functional term. A function type is described by a pair $\langle \sigma_0, \langle \sigma_1, \ldots, \sigma_n \rangle \rangle$ specifying the result and argument types respectively. As well as the arbitrary function symbols in $\mathcal{V}_f$, VEL includes the unary logical functions: b and e (both type $\langle \sigma_t, \langle \sigma_\delta \rangle \rangle$), state (type $\langle \sigma_s, \langle \sigma_t \rangle \rangle$), time (type $\langle \sigma_t, \langle \sigma_s \rangle \rangle$) and dur (type $\langle \sigma_\delta, \langle \sigma_\varepsilon \rangle \rangle$).

Event Type Expressions: An event type, or $\sigma_{\mathbf{e}}$, expression is either:

- a 0-ary verb $\in \mathcal{V}_v$,
- an expression $\mathbf{v}(\alpha_1, \ldots, \alpha_n)$, where $\mathbf{v}$ is an n-ary verb $\in \mathcal{V}_v$ and the αs are of type σ_i,
- an expression $\langle \delta_i : \varphi \rangle$, where φ is a *proposition* (as defined below),
- an expression $\{\theta, \tau\}$, where θ is of type $\sigma_{\mathbf{e}}$ and τ is of type σ_t.

Count Nouns: As well as the atomic count nouns in $\mathcal{V}_c$ the type σ_c also includes all expressions of the form $\{\kappa, \tau\}$, where κ is of type σ_c and τ of type σ_t.

Atomic Propositions: An atomic proposition has one of the following forms:

- $\alpha_1 = \alpha_2$, where the α's are terms of the same type.
- $\tau_1 \leq \tau_2$, where the τ's are terms of type σ_t.
- $R(\beta_1, \ldots, \beta_n)$, where $R \in \mathcal{V}_R$ and $\beta_1, \ldots, \beta_n$ are terms.

Each argument of a relation must be of a specific nominal type. The type of an n-ary relation is described by a tuple $\langle \sigma_1, \ldots, \sigma_n \rangle$. Apart from the arbitrary relations in $\mathcal{V}_R$ there are two unary logical predicates AT and AS of types $\langle \sigma_t \rangle$ and $\langle \sigma_s \rangle$ respectively.

Propositions: A proposition of VEL is either atomic or has one of the forms:

- one of: $\neg\varphi$, $\varphi_1 \wedge \varphi_2$, $\mathbf{F}\varphi$, $\mathbf{P}\varphi$, $\Box\varphi$, $\boxast\varphi$, $\triangleright\varphi$, $\triangleleft\varphi$,
- $\mathsf{Holds\text{-}at}(\varphi, \tau)$, where τ is of type σ_t,
- $\mathsf{Holds\text{-}in}(\varphi, \varsigma)$, where ς is of type σ_s,
- $\mathsf{Occurs}(\mathbf{e}, \delta)$, where $\mathbf{e}$ is of type $\sigma_{\mathbf{e}}$ and δ is of type σ_δ,
- $[\mathbf{e}]\varphi$, where $\mathbf{e}$ is of type $\sigma_{\mathbf{e}}$,
- $(\forall\chi)\varphi$, where χ is a variable of one of the types $\sigma_t, \sigma_\delta, \sigma_s, \sigma_o$,
- $(\forall\kappa, \upsilon)\varphi$, where κ is of type σ_c and υ is a variable of type σ_i,
- $\langle\forall\theta, \varepsilon_i\rangle\varphi$, where θ is of type $\sigma_{\mathbf{e}}$,
- one of: $\mathit{Perf}\theta$, $\mathit{Prog}\theta$, $\mathit{Pros}\theta$, where θ is type $\sigma_{\mathbf{e}}$.

where in each case φ is any VEL proposition.

5 Informal Interpretation

We now give an informal account of the semantic interpretation of VEL (a fully formal semantics can be found in (Bennett and Galton 2000). One aspect of the formal semantics one does need to be aware of is that each expression will be given a denotation relative to an index point $\langle h, t \rangle$ within a history tree structure. At this point h is the *actual history*, t is the *actual time* and $h(t)$ is the *actual state.*

The Logical Functions and Predicates:

The logical functions are used to convert between different semantic types: $\mathsf{b}(\delta)$ and $\mathsf{e}(\delta)$ return times respectively corresponding to the beginning and end points of the interval δ. $\mathsf{state}(\tau)$ gives the world state that obtains at time denoted by τ in the actual history. $\mathsf{time}(\varsigma)$, where ς denotes the state s, gives the (unique) time t for which there is some history h' such that $h'(t) = s$. $\mathsf{dur}(\varepsilon)$ gives the interval of occurrence of the event token ε. $\mathsf{AT}(\tau)$ is true just in case τ denotes the actual state. $\mathsf{AS}(\varsigma)$ is true just in case ς denotes the actual state.

Count Nouns and Quantifiers over Individuals:

A component of VEL that is probably unfamiliar to most AI researchers is the use of Gupta's (1980) theory of count nouns. At each index $\langle h, t\rangle$, a count noun picks out a set of individuals. This means that the entities picked out by a count noun are doubly relative to the possible world at which a count noun expression is evaluated: the count noun is evaluated at some possible world to give a set of individual concepts which are instances of the count noun. These individual concepts must in turn be evaluated relative to a possible world to yield an entity existing at that world. The reason for this double relativity is that we very often employ count nouns to pick out individuals at some particular world and then consider properties of these individuals at another world. E.g. in the sentence "Some girl will become president of the USA", the phrase "some girl" quantifies over individuals who are girls at the present time. The individual concepts corresponding to these girls will denote different entities at different possible worlds but in many of these worlds these individuals will not be girls. In particular, in some future world, one of these individuals may be a woman President.

Quantification over individuals takes the form

$$(\forall\textsc{c}, x)[\varphi(x)].$$

The count noun operates like a *sort* but it also fixes the trans-world identity conditions of quantified variables. We also define:

D1) $(\exists\textsc{c}, x)\varphi(x) \ \equiv_{def}\ \neg(\forall\textsc{c}, x)\neg\varphi(x)$,
D2) $\textsc{c}(\alpha) \ \equiv_{def}\ (\exists\textsc{c}, x)[x = \alpha]$,
D3) $\textsc{c}[\alpha] \ \equiv_{def}\ (\exists\textsc{c}, x)[\Box(x = \alpha)]$.

$\textsc{c}[\alpha]$ means that α denotes one of the individuals that is (at the actual index) a $\textsc{c}$; whereas, $\textsc{c}(\alpha)$ means that α denotes an individual that is merely co-realised by some individual that is a $\textsc{c}$. Thus we might write: $(\exists\textsc{president}, a)[\textsc{woman}[a]]$, since presidents and women are both individuals of the same kind and share the same trans-world identity criteria; but $(\exists\textsc{quantity-of-wood}, a)[\textsc{ship}(a)]$, since, although ships and quantities of wood may be materially identical (co-realised) in some world state, they do not have the same trans-world identity criteria (e.g. consider the ship of Theseus).

Propositional Operators:

VEL includes a variety of well-known temporal operators and modalities. **P** and **F** are past and future tense operators defined in the normal way. $\Box$ is just an $S5$ modality operator — $\Box\varphi$ is true if φ is true at every index point. The $\boxast$ operator is a bit more subtle: $\boxast\varphi$ is true at $\langle h, t\rangle$ if φ is true at t in all histories which share the same initial segment as h up to time t. Thus, $\boxast\mathbf{F}\varphi$ means that φ is 'inevitable'. Dual operators $\Diamond$ and $\diamondast$ are defined in terms of $\Box$ and $\boxast$ in the usual way. ($\diamondast$ can be used to model the E operator of CTL*, which has a similar interpretation.)

We also have operators $\triangleleft\varphi$ and $\triangleright\varphi$, meaning respectively that φ 'has just been true' or 'is just about to become true'. $\triangleright\varphi$ is rather like the standard 'next time' operator, except that it makes sense within a dense of continuous time series. It means that φ is true over some open set of time points, whose greatest lower bound is the actual time. $\triangleleft$ is defined analogously. Such operators were proposed by (Barringer, Kuiper and Pnueli 1986).

Relations and Verbs:

Relations represent arbitrary intensional constraints over the universe of possible denotations of nominal terms. At each index point an n-ary relation is associated with the set of n-tuples drawn from this universe for which the relation holds. These tuples must respect the type of the relation (full details are given in (Bennett and Galton 2000)). Verbs combine with one or more expressions denoting individuals to produce an expression that denotes an event type. At each index a verb is associated with a mapping from n-tuples of individuals to event types.

Syntactic Rôles of Event Types:

Expressions of the category of event type can combine with other expressions in various different ways. Corresponding to the four theories of events we have:

Event Modalities: In the style of dynamic logics we employ event types within a propositional operation $[\mathbf{e}]\varphi$, which says that φ holds in all states accessible from the current state *via* an occurrence of an event of type $\mathbf{e}$. The dual operator $<\!\mathbf{e}\!>\varphi$ is defined in the usual way.

The Occurs **Relation:** The Occurs relation provides a flexible way of talking about types of events in terms of conditions which hold before and after (and perhaps also during) the time period over which an event of that type takes place. For example, following Galton (Galton 1997), we could define the event type where a body moves between two positions by:

$$\Box(\forall \text{BODY}, b)(\forall r_1, r_2)[\mathsf{Occurs}(\mathbf{move}(b, r_1, r_2), i) \\ \leftrightarrow \\ (\ \mathsf{Holds\text{-}at}((\mathsf{pos}(b) = r_1), \mathsf{begin}(i)) \wedge \\ \mathsf{Holds\text{-}at}((\mathsf{pos}(b) = r_2), \mathsf{end}(i)) \wedge \\ \mathsf{Holds}((\mathsf{pos}(b) \neq r_1), i) \wedge \\ \mathsf{Holds}((\mathsf{pos}(b) \neq r_2), i)\ )]\ .$$

Here r_i are observable variables and pos is a function returning an observable value giving the region occupied by an individual.

Quantifying Over Singular Events: Allowing event types to act as sortals in quantification over events enables us to maintain syntactic homogeneity unifying a Davidson like analysis with the others. It also provides a mechanism for associating event types with event token variables. A quantification over event tokens is a formula of the form $\langle \forall \mathbf{e}, \varepsilon \rangle [\varphi(\varepsilon)]$. Because the event token variable carries with it an event type as well as an episode the truth conditions of properties of event tokens (and also the values of functions involving them) are not determined solely by the time period over which they occur.

Aspect Operators: The operators *Perf*, *Pros* and *Prog* operate on event type expressions to produce propositions. *Perf* **e**, *Pros* **e** and *Prog* **e** mean respectively that event **e** has happened, will happen, and is happening. If we want to say that an event is in progress without asserting that it will definitely be completed we can use the form $\diamondsuit\, \mathit{Prog}\, \mathbf{e}$. We can say that an event is just about to happen using the form $\neg \mathit{Prog}\, \mathbf{e} \wedge \triangleright \mathit{Prog}\, \mathbf{e}$ (for just past use $\triangleleft$).

Event Type Abstraction:

The construct $\langle \delta : \varphi(\delta) \rangle$ serves as a powerful means of defining event types in terms of conditions on the time intervals over which it occurs. When evaluated at an index $\langle h, t \rangle$, the expression $\langle \delta : \varphi(\delta) \rangle$ refers to the event type corresponding to the set of all episodes $[s...f]^{\approx}_{h'}$ such that $\varphi([s...f])$ is true at index $\langle h', t \rangle$.[3] To illustrate the use of event abstraction, let us define a verb $\mathbf{collide}(a, b)$ describing (punctual) collision events between two individuals:

$$\Box(\forall \text{ BODY}, a, b)[\,\mathbf{collide}(a, b) \ \doteq \\ \langle i : ((\mathsf{b}(i) = \mathsf{e}(i) = t) \wedge \\ \mathsf{Holds\text{-}At}(\mathsf{C}(a, b) \wedge \triangleleft \neg \mathsf{C}(a, b), t)) \rangle]$$

Here C is the connection relation of the RCC spatial theory (Randell, Cui and Cohn 1992).

Holds-at **and** Holds-in**:**

VEL includes the construct $\mathsf{Holds\text{-}at}(\varphi, \tau)$, which has been employed in many AI representations. It is true just in case φ is true (in the actual history) at the time point denoted by τ. In order that the logic of count nouns and events interacts flexibly with the logic of the Holds-at predicate we also need constructs for shifting the point of evaluation of these expressions to a given time point. Thus, $\{\text{C}, \tau\}$ refers to the class of individuals which is picked out by {C at time point τ (on the actual history). E.g. $(\forall \{\text{BOY}, 1960\}, x)[\{\text{MAN}, 1999\}[x]]$. The construct $\{\mathbf{e}, \tau\}$ is required for technical reasons but is seldom used because the denotation of most event type expressions is constant over time.

$\mathsf{Holds\text{-}in}(\varphi, \varsigma)$ asserts that φ is 'settled' as true at the state s denoted by ς. This means that φ is true at every index $\langle h, t \rangle$ such that $h(t) = s$.

6 Conclusion

We have given an overview of a very expressive temporal language (VEL), which is capable of describing events in a variety of different ways. In particular it incorporates within a single formalism four of the most influential approaches to representing events. The framework provides a general temporal ontology within which more practically oriented representations can be interpreted. It also supports the design of modular reasoning systems combining decision procedures for tractable sublanguages of VEL. For a full account of the formal semantics of the language the reader should consult (Bennett and Galton 2000).

[3] Thus, in general, the abstracted event type depends on the time t of the point of evaluation; but in most useful cases $\varphi(\delta)$ will actually be invariant over time, unless φ contains some embedded indexical.

In further work we hope to establish a complete axiom system for VEL and to relate our system to formalisms such as CTL* whose computational properties are better understood. In the unpublished paper (Bennett n.d.), completeness is proved for an axiom system for a significant sub-language of VEL incorporating the operators $\mathsf{Holds\text{-}at}(\varphi, t)$, $\mathsf{AT}(t)$ and $\circledast\varphi$ together with quantification over time variables. (Bennett and Galton 2000) shows how these are sufficient to define many of the other operators of VEL.

Preliminary work on describing physical processes suggest that VEL is well suited to modelling causality. We would like to further explore how VEL might take account of issues such as the *frame problem* and how it relates to non-monotonic formalisms that deal with this. We are also working on a spatial ontology/semantics inspired by (Randell et al. 1992), which we intend to combine with VEL.

References

Allen, J. F.: 1984, Towards a general theory of action and time, *Artificial Intelligence* **23**(2), 123–154.

Baker, A. B.: 1991, Nonmonotonic reasoning in the framework of the situation calculus, *Artificial Intelligence* **49**, 5–23. Special issue on knowledge representation.

Barringer, H., Kuiper, R. and Pnueli, A.: 1986, A really abstract concurrent model and its fully abstract semantics, *Proceedings of the 13th ACM symposium on principles of programming languages (POPL)*, pp. 173–183.

Bennett, B.: n.d., A logic of branching histories with a shared linear time series, Available from web at `http://www.scs.leeds.ac.uk/brandon/atcomplete.ps`.

Bennett, B. and Galton, A.: 2000, A unifying semantics for time and events, *Technical report*, QSR Group Technical Report, School of Computing, University of Leeds, Leeds LS2 9JT, UK. Available from web at `http://www.scs.leeds.ac.uk/pub/doc/srg/time_tech_qsr.ps`.

Brown, F. M. (ed.): 1987, *The Frame Problem in Artificial Intelligence: Proceedings of the 1987 Workshop*, Morgan Kaufmann, Los Altos, Ca.

Davidson, D.: 1967, The logical form of action sentences, *in* N. Rescher (ed.), *The Logic of Decision and Action*, University of Pitsburgh Press. Reprinted in *Essays on Actions and Events*, Oxford University Press, 1980.

Davidson, D.: 1980, The logical form of action sentences, *Essays on Actions and Events*, Clarendon Press, Oxford, chapter 6, pp. 196 – 148.

Davis, E.: 1990, *Representations of Commonsense Knowledge*, Morgan Kaufmann, San Mateo.

Emerson, E. and Halpern, J.: 1986, 'sometimes' and 'not never' revisited: on branching time versus linear time temporal logics, *Journal of the ACM* **33**(1), 151–178.

Fikes, R. E. and Nilsson, N. J.: 1971, STRIPS: A new approach to the application of theorem proving to problem solving, *Artificial Intelligence* **2**, 189–208.

Galton, A. P.: 1984, *The Logic of Aspect: an Axiomatic Approach*, Clarendon Press, Oxford.

Galton, A. P.: 1987, The logic of occurrence, *in* A. P. Galton (ed.), *Temporal Logics and their Applications*, Academic Press, London, chapter 5, pp. 169–196.

Galton, A. P.: 1993, Towards an integrated logic of space, time and motion, *Proceedings IJCAI-93*, Chambery, France.

Galton, A. P.: 1997, Space, time and movement, *in* O. Stock (ed.), *Spatial and Temporal Reasoning*, Kluwer, Dordrecht, chapter 10, pp. 321–352.

Guarino, N. (ed.): 1998, *Formal ontology in information systems: Proceedings of the 1st international conference (FOIS-98)*, Vol. 46 of *Frontiers in Artificial Intelligence and Applications*, Ios Press, Trento, Italy.

Gupta, A.: 1980, *The Logic of Common Nouns: an investigation in quantified modal logic*, Yale University Press, New Haven.

Hanks, S. and McDermott, D.: 1986, Default reasoning, nonmonotonic logics, and the frame problem, *Proceeding AAAI-86*, pp. 328–333.

Harel, D.: 1984, Dynamic logic, *in* D. Gabbay and F. Guenthner (eds), *Handbook of Philosophical Logic, Vol. II: extensions of classical logic*, Reidel, pp. 497–604.

Hayes, P. J.: 1973, The frame problem and related problems in artificial intelligence, *in* A. Elithorn and D. Jones (eds), *Artificial and Human Thinking*, Jossey-Bass, San Fransisco, pp. 495–520.

Hayes, P. J.: 1979, The naïve physics manifesto, *in* D. Mitchie (ed.), *Expert systems in the microelectronic age*, Edinburgh University Press.

Kowalski, R. and Sergot, M.: 1986, A logic-based calculus of events, *New Generation Computing* **4**, 67–95.

Lehmann, F. (ed.): 1992, *Semantic Networks in Artificial Intelligence*, Pergamon Press.

Lenat, D.: 1995, CYC: a large-scale investment in knowledge infrastructure, *Communications of the ACM*.

McCarthy, J. and Hayes, P.: 1969, Some philosophical problems from the standpoint of artificial intelligence, *in* B. Meltzer and D. Mitchie (eds), *Machine Intelligenge*, Vol. 4, Edinburgh University Press, pp. 463–502.

McDermott, D.: 1982, A temporal logic for reasoning about processes and plans, *Cognitive Science* **6**, 101–155.

Moller, F. and Birtwistle, G. (eds): 1996, *Logics for Concurrency: structure versus automata*, number 1043 in *LNCS*, Springer-Verlag.

Moller, F. and Rabinovich, A.: 1999, On the expressive power of CTL*, *Proceedings of LICS'99: Logic in Computer Science*.

Randell, D., Cui, Z. and Cohn, A.: 1992, A spatial logic based on regions and connection, *Proc. 3rd Int. Conf. on Knowledge Representation and Reasoning*, Morgan Kaufmann, San Mateo, pp. 165–176.

Sandewall, E.: 1994, *Features and Fluents*, Oxford University Press.

Shanahan, M.: 1997, *Solving the Frame Problem: a mathematical investigation of the common sense law of intertia*, The MIT Press, Cambridge, Massachusetts.

Shanahan, M.: 1999, The event calculus explained, *in* M.J.Wooldridge and M. Veloso (eds), *Artificial Intelligence Today*, Vol. 1600 of *LNCS*, Springer-Verlag, pp. 409–430.

Shoham, Y.: 1988, Time for action: On the relation between time, knowledge and action, *Proceedings 11th IJCAI*, Detroit, USA, pp. 954–959.

Thomason, R.: 1984, Combinations of tense and modality, *in* D. Gabbay and F. Guenthner (eds), *Handbook of Philosophical Logic*, Vol. II, Reidel, chapter 3, pp. 135–165.