Event-centric Policy Specification for E-commerce Applications

Alan S. Abrahams and Jean M. Bacon
University of Cambridge Computer Laboratory
{asa28, jmb}@cl.cam.ac.uk

1 Introduction

This paper introduces an event-centric paradigm for policy specification in electronic commerce applications. We first introduce the problem we are trying to solve and describe why events may help to address the problem. Events are defined and the event paradigm is compared briefly to object-orientation. We show the relationship between events and policy. A brief mapping from natural language to events is illustrated; we describe how some core and domain-specific events can be exposed in natural language specifications and provide some simple examples. Finally we explain how we intend to develop these ideas.

2 Background

The objective of our work is to allow for the executable specification of electronic commerce applications in a natural, English-like form. Current e-commerce applications are written in languages such as Java, Perl, and VBScript. These system-level languages make the application difficult for business users to read and modify, and bury the business procedures (policy) in code. Controlled, English-like, languages such as ACE [7] have been proposed for the executable specification of applications. ACE cannot be used to specify executable e-commerce applications. Aside from its lack of bindings to internet technologies such as HTTP and SMTP, ACE's controlled language explicitly forbids the usage of modal verbs (such as 'can' and 'must') which would be used in English to concisely specify norms: that is, authorizations and obligations. A thorough treatment of norms and norm violation is critically necessary in e-commerce for the definition of contracts between parties and also for the specification of internal procedures, practices, and workflows. Support for the definition of norms (authorization and obligation policies) is provided by policy notations like Ponder [6]. However, Ponder is an implementation-phase focussed technology – it omits analysis and design considerations and provides few pointers for mapping from a natural language specification to events or policies. Recently some work towards translating controlled English specifications into policy been undertaken at Cambridge [4]. However, this has been confined to the functionality provided by OASIS access control policy. Traditional event frameworks like CEA [3] and GEM [11] also omit detailed mappings to English.

We believe that translating an e-commerce application specification into events will eliminate the separation between (English) specification and (programming language) code: the specification, once mapped to descriptive and normative events, can be stored as data in an event store and can be used to control the behaviour of the application. This should bring benefits of ease of maintenance and understanding, and improve the credibility of systems by allowing the business practices implemented to be clear and unobscured by programming language syntax.

In *Section 3* we explain the major concepts of the event-centric paradigm. An explanation of what is meant by event subordination follows. *Section 5* lists the core types of events. *Section 6* explains how events may be found in natural language specifications, while *Section 7* gives sample mappings from real specifications.

3 The Event-Centric Paradigm: Events, Participants, and Parameters

Ponder, OASIS, CEA, and GEM are intended to be integrated with object-oriented technologies; The approach we use is event-centric rather than object-centric. The event-centric paradigm treats events as the primary abstraction. Uniquely identified referents may be participants in events such as classification events (specifying the type or class of the referent), normative events (specifying what the referent can and must do), and other (e.g. domain-specific) events. **Referents** [9] are entities denoted (or denotable) by a unique identifier, and may be things, places, concepts, roles, or other events. Referents are **participants** in the event and are bound to the event in **roles**. Roles may be:

- *event-specific roles*, which depend on the nature of the event itself. For example, 'writer' and 'written' are roles of participants in a 'write' event.
- *semantic or thematic roles* [1, 13], such as actor, patient, instrument, input (resource), or output (product / result) of the event, which pertain to many event types. For example, a pen may be the instrument of a 'write' event, the writer would be the actor, and item written would be the output of the event.

Events are regarded as any occurrence, process, wilful action or activity, or state, in which referents participate and can be identified, inter alia, from *verbs*. Events may have **parameters** such as time (instant or period) and place.

Though it is unusual to treat the notion of events as incorporating the notion of states we have opted for this treatment as we believe both events and states denote a time-delimited relatedness between entities and are commonly denoted using verbs in English. Notably, Bach [2] treats 'eventualities' as covering both 'events' and 'states'; we have chosen to follow Bach here. For instance 'The unit trust manager *approves* the share sale' is typically considered an event whereas 'The share sale is *approved*' would be considered a follow-on state. In this case, we believe that the state 'approved' is merely a retrospective view of a past, successfully completed 'approve' event. Our contention is that both 'approving' and 'approved' refers to the self-same relatedness as was initially indicated by the 'approve' event - a relatedness between a 'unit trust manager' entity and a 'share sale' entity via an 'approves' event viewed from different perspectives. Barring any reverse action (e.g. cancelling the approval), the 'approved' state will hold. Forever in the future we could contend that the 'share sale was approved' though this may be somewhat misleading as it may imply, perhaps falsely, that the sale is still approved.

It is worthwhile to point out here that the event-centric approach does not make use of the traditional object-oriented notion of objects having attributes and methods. Instead **state** is determined by the currently applicable event bindings to a referent, and **behaviour** of a referent is determined by the ability (authorization) or obligation to:

- insert new events bound to the referent into actionable queues when composite events[1] are detected, or
- invoke external operations.

Referents have rights and responsibilities: responsibilities are obligations to which the referent is subjected, rights are authorizations and obligations in favour of the referent.

4 Subordination of Events

Traditional event frameworks, such as GEM and CEA mentioned earlier, do not provide for event subordination – a notion common in English grammar [12] – where events are relevant only in the context of other events. For instance, in 'administrator *authorizes* John to *read* file', it would seem that 'authorize' is an actual event (in the real world), whereas 'read' is an event that exists in a world of norms created by the 'authorization'. The sentence does not imply that any reading has occurred in the real world. Instead the 'read' event is subordinated to an 'authorize' event and therefore exists only in a sub-world; when interpreting this sentence it would be improper to take any action based on the 'read' event since it has not really occurred - it has only been mentioned in relation to a superordinate 'authorization'.

5 Types of Events

The core event types we define are:

- **Factual** events: These can be user-interface events such as 'selects', 'clicks', and 'displays', or business events. Business events may be:
  - <u>contractual</u> events such as 'buys', 'leases', 'subscribes', or 'insures' which entail the acceptance of terms and definitions and the incurrence of obligations; or
  - <u>workflow</u> events such as 'charges', 'pays', 'approves' which typically entail the discharge of obligations or the exercising of rights (including the uptake of opportunities).
- **Descriptive and definitional policy** events: these include 'naming', 'classifying'[2], and 'quantifying' (e.g. 'counting') events. Depending on the mood of the specification utterance, naming and classification events may:
  - Be operational data: That is, they may be deemed to have occurred, indicating that a referents *is* so named or classified.
  - Indicate definitional policy: That is, they may be sub-ordinated to an *obligation* event, implying that any matching referents *must be* so named or classified.

  Referents may be multiply typed (classified) and types (classification events) may be mutually exclusive. Definitions ('define' events) are obligations to associate a name or classification with items that comply with a description (criteria or constraints). For example:

  <u>Name/Classification</u>: Wealthy Londoners
  are *defined* as (i.e. is a classification that must be given to):
  <u>Description/Criteria</u>: People with yearly income > £100k per year and telephone number beginning with 0207 or 0208.[3]

  There may then be certain prescriptive policies – e.g. related to web-page content personalization – which pertain to referents *classified* as 'wealthy Londoners'.

  <u>*Type-determination*</u> in the event-centric paradigm is supported through the triggering of a *classification* event when a relevant composite event is detected. In the event-centric development paradigm, the type

[1] Refer to [3] for a discussion of composite events and the operators available for composite event detection.

[2] All event occurrences are typed: typing of events is achieved by the participation of an event in one or more classification events.

[3] This definition would more appropriately be atomized into separate definitions for 'wealthy' and 'Londoner'.

of a concept is determined by events which the (extensionally or intensionally) identified referents have enacted or could enact (as *willing* actors or *controlled* instruments) or to which they have been, or could be, subject (as *voluntary or involuntary* patients). So, the event-centric paradigm supports at least two modes of type determination:

1. Type determination based on *factual event-history*.
2. Type determination based on *pattern of permissible future invocations* (derivable from *normative event history*).

Again, the **behaviour** of referents that are of a particular type – their reaction to events – is determined by what events *must* or *can* be inserted into the event store (or what external operations must or can be invoked) for referents of that type.

Unlike object-orientation, the event-centric paradigm makes direct provision for subjectivism via optionally specified actors of classification, naming, and definition events.

In the event paradigm, both specifications (normative events) and data (factual events) are stored in the event store. Code and data are thus uniformly treated and can manipulated and queried using a common language.

- **Prescriptive policy (normative)** events: these include 'authorize', 'oblige', and 'forbid'. Informally, we can say that prescriptive policy allows us to define what can or must (or cannot or must not) do what to what and when.

Box 1 illustrates some verb frame templates that may be used for the input of descriptive and prescriptive policy events. The syntax of factual events can, in part, be based on verb frames from WordNet [15].

Box 1: Some core events useful across varied application domains

```
{[somebody]} authorizes {[somebody] to} [qualified event]
{[somebody]} obliges {[somebody] to} [qualified event]
{[somebody]} forbids {[somebody] to} [qualified event]
{[somebody]} classifies [something] as [concept] denoted by [symbol]
{[somebody]} names [something][symbol]
{[somebody]} defines [some-classification] as [criteria]
{[somebody]} quantifies [something] as [measure]
```

'Symbol' is any word or sequence of letters from the language. 'Somebody' is any referent classified as an agent. 'Something' is any referent. 'Concept' is an abstract referent with unique semantics denotable by one or more symbols (synonyms). Where the subject is omitted, the verb is stated in the passive form.

In all cases, referents can be extensionally or intensionally defined: that is, they must be either one or more unique identifiers, or they can be '*'+*criteria* indicating any referent that conforms to specified criteria. 'Qualified event' is any event conforming to stated criteria. Criteria are constraints on the values or types of participants or parameters to the event.

6 Exposing Events in an English Language Specification

A variety of mechanisms are available for exposing events. These can be briefly summarized as follows:

- Contractual and workflow events can typically be identified from verbs and their morphological forms in specifications. For instance a 'subscribe' event may appear as 'subscribes', 'subscribing', or as the deverbative noun 'subscription'. It is important that all these be referenced to a single canonical form (e.g. 'subscribe') if they co-refer to a given verb sense. Possessive forms (e.g. "'s" and "of") may imply the existence of authorizations and obligations (rights and responsibilities) that surround ownership. Roles — which often end in –er, –or, –ant, or –ent in English[4] — frequently indicate underlying domain-specific events. For example, the 'manager' role implies a 'manages' event. Underlying normative events (e.g. oblige, authorise) that describe the *responsibilities* and *privileges* of the role may also be implied.
- Modals such as 'can', 'must', 'should', 'have to' and the suffices '-able' and '-ible' may imply 'authorize', 'forbid', or 'oblige' events.
- Nouns and adjectives imply 'classifying' events; proper nouns imply 'naming' events. Adverbs may imply 'classify' events upon referents that are events.
- Determiners, anaphora, and modifiers or qualifiers (such as 'the', 'that', and 'which') may imply 'selecting' (find and filter) events, as may conjunctions ('and') and disjunctions ('or').
- Quantifiers (such as the cardinals '1', '2', 'one', etc. and words such as 'all', 'none', 'a', and 'some'), and singular, plural, and collective forms may imply 'counting' events.
- Ordinals (such as 'first', '3rd', and 'last'), comparatives and superlatives (e.g. ending in '-er' and '-est'), and prefixes like pre-, post-, and suc- may imply 'sorting' events.
- Negatives may imply negation, failure, forbiddance, or non-occurrence of events, rejection (non-conformance) events, or reversal of the effects of a previous event (i.e. retraction of assertions).
- Tenses and temporal function words (like 'before', 'after', 'within' and 'during') may denote temporal relations [1, 8, 10] between events.
- Speech acts (moods) [5, 14] may imply events: imperative and commisive speech acts imply 'oblige' events; declarative speech acts imply 'naming' or 'classification' events; interrogative speech acts may imply 'selection' (output of finding and filtering according to criteria) events.

[4] E.g. Farm<u>er</u>, Act<u>or</u>, Particip<u>ant</u>

7 Sample mappings from English to events

Following are two examples which demonstrate simple policies common in e-commerce websites and their implementation in an event-centric paradigm:

7.1 E-commerce Example 1

In the example:

"Customers can buy goods if they are registered".

we can see that:

- The modal 'can' implies an 'authorize' event.
- The nouns 'customers' and 'goods' imply 'classification' events. Specifically they imply that we are looking to match the events: 'referents classified as customer' and 'referents classified as goods'.
- The anaphor 'they' is ambiguous here; being plural, it could imply the selection of either the referents classified as 'customers' or the referents classified as 'goods'. In this case it is the former, and we can substitute 'the customers' for 'they' to disambiguate the sentence. Alternatively, a reordering of the sentence as "Customers, if they are registered, can buy goods" may resolve the ambiguity. Identifying and catering for ambiguity in original phrasings is useful though, as alternative, ambiguity-controlled, phrasings may have a more stilted reading and so would often not be the first-choice of system analysts who write specifications.
- The past tense 'registered' implies we are looking for referents – customers – that have in the past (relative to the policy usage event), participated in a 'register' event[5].
- The function word 'if' implies an event (in this case an 'authorize' event) must be added to the event store when a certain condition is met.

The full translation to an event-centric implementation would be:

"If there is a referent in the event store that is *classified* as a customer as a result of its participation as an actor in a *register* event (i.e. an event occurrence classified as 'register'[6] and temporally before the time of policy usage), then add to the event store an *authorize* event, with the afore-mentioned referent as a participant (beneficiary). This authorization event authorizes 'buy' events to be added to the event store with the afore-mentioned referent (i.e. customer) as an agent of the buy event – i.e. as the buy*er*. A *violation* event may be triggered if there is an attempt by a non-registered customer (i.e. referent classified as customer but not classified as registered) or by a non-customer to buy goods."

7.2 E-commerce Example 2

In the example:

"Purchase orders need three management signatures before they may be approved".

we can see that:

- The modal 'need' implies an 'oblige' event.
- The deverbative noun 'signature' implies a 'sign' event.
- The modal 'may' implies an 'authorize' event.
- The cardinal 'three' implies that we need to trigger a 'counting' event.
- 'Before' implies that the authorization ('authorize' event) may only be triggered after the obligation to count three management signatures for that purchase order has been fulfilled. Furthermore, it implies that there should be a default 'forbid' in the absence of the necessary explicit authorization.

So here, where we encounter a referent *classified* as a purchase order, the system is obliged to, by default, forbid approval events on said purchase order. The system is also obliged to count *sign* (signature) events with actors *classified* as managers and of which that referent (i.e. purchase order) is a patient. If this count yields an output of three, then we can *authorize* approval events which take that referent as a patient (by adding a relevant 'authorize' event to the event store, which overrides the default 'forbid'). In the absence of explicit authorization events, attempts to add any unauthorized approval to the event store are a violation of the default forbiddance on such events.

8 Future Work

Work on the current framework and architecture is ongoing. We are currently beginning implementation of the event-centric development and execution environment and thereafter will attempt to execute a set of real-life specifications which we have gathered. Insights gained will be used to adapt and refine the framework. Events in the current implementation will be stored in a relational database and transferred as text over TCP/IP, SMTP, or HTTP. Supporting tools used will be WordNet, the Cambridge International Dictionary of English electronic edition, as well as a part-of-speech tagger and lemmatizer, the latter being used to identify the canonical form of verbs.

[5] Technically we would need to specify that the register event was one that resulted in the referent being classified as a customer, since it would not be sufficient if the referent was classified as 'registered' – e.g. as a user – and then subsequently, in an unrelated event, classified as a 'customer'.

[6] Event occurrences themselves are actually stored as referents and then *classified* as being events of event types. Note that further classifications of the event occurrence may be necessary in order to disambiguate polysemous verbs and arrive at a single sense or meaning for the verb.

9 References

[1] Allen J. *Natural Language Understanding: Second Edition*. Benjamin/Cummings. 1995. pp 1-41, 248

[2] Bach E. The Algebra of Events. *Linguistics and Philosophy*. 9 (1986). pp. 5-16

[3] Bacon J, Moody K, Bates J, Hayton R, Ma C, McNeil A, Seidel O, Spiteri M, Generic Support for Distributed Applications. *IEEE Computer*. March 2000, pp 68-76.

[4] Bacon J, Lloyd M, Moody K, Translating Role Based Access Control Policy within Context. *Policy Workshop 2001*. Bristol, January 29-31, 2001.

[5] Cruse A. *Meaning in Language: An Introduction to Semantics and Pragmatics*. Oxford University Press, Oxford, UK. 2000. Chapter 16, pp 331-346.

[6] Damianou N, Dulay N, Lupu M, Sloman M. *Ponder: A Language for Specifying Security and Management Policies for Distributed Systems. The Language Specification. Version 1.11*. Imperial Collect Research Report DoC 2000/1. 18th January 2000.

[7] Fuchs NE, Schwertel U, Schitter R. Attempto Controlled English (ACE) Language Manual, Version 3.0. Insitut fur Informatik der Universitat Zurich. August 1999. Available at: http://www.ifi.unizh.ch/groups/req/projects/attempto/rerg_projects_vtclp.html

[8] Hayes PJ. *A Catalog of Temporal Theories*, Tech report UIUC-BI-AI-96-01, Beckman Institute and Departments of Philosophy and Computer Science, University of Illinois 1995. Available at: http://www.coginst.uwf.edu/~phayes/TimeCatalog1.ps
Accessed on: 1 May 2000

[9] Kamp H, and Reyle U. *From Discourse to Logic*. Kluwer Academic Publishers. 1993.

[10] Ladkin PB. *The Logic of Time Representation*. PhD Thesis. Group in Logic and Methodology of Science, University of California at Berkeley. November 1987.

[11] Mansouri-Samani M and Sloman M. GEM: A Generalised Event Monitoring Language for Distributed Systems. *IEE/IOP/BCS Distributed Systems Engineering Journal*, Vol. 4 No. 2, June 1997. Extended version of a paper presented at ICODP/ICDP '97 conference, Toronto, May, 1997.

[12] Palmer F. *Grammar*. Penguin Books. 1984.

[13] Sowa JF. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks/Cole. Pacific Grove, California, USA. 2000. Revisions to the chapter on *Thematic Roles* available at: http://www.bestweb.net/~sowa/ontology/thematic.htm

[14] Thomas J. *Meaning in Interaction: An Introduction to Pragmatics*. Addison Wesley Longman Limited. 1998. pp 1-54

[15] *WordNet – A Lexical Database for English*. Available from: http://www.cogsci.princeton.edu/~wn/online/