Event-centric business rules in e-commerce applications

Alan S. Abrahams and Jean M. Bacon

University of Cambridge Computer Laboratory
Cambridge, United Kingdom
Facsimile: +44 1223 334678
{asa28, jmb}@cl.cam.ac.uk

ABSTRACT: *In this position paper we propose an event-based decomposition of business rules, providing guidance and examples of how rules from e-commerce applications may be decomposed into events and stored in an event store. Rules may be grouped into packets or contracts for manageability.*

KEY WORDS: business rules, policy, events

1 Introduction

We have been investigating the specification of business rules (policy) in a variety of e-commerce applications. We are developing a purely event-based architecture to facilitate the executable specification of these business rules. Rules are stored as typed events, which have participants that occupy various thematic roles. We describe the notion of events and discuss the core event types useful for the definition of *descriptive* and *prescriptive* business rules. These core events include those for classification, authorization, forbiddance, and obligation. The *business rules (policy) life cycle* is described; we list the events that can happen to business rules, which in turn determine the state of business rules during their lifetime. We also discuss the grouping of business rules into *packets* and *contracts* for manageability, and set out the events that can apply to these packets and contracts during their life cycle. Real world e-commerce examples, categorized into various rule types, are provided.

2 Events

The event-centric paradigm treats events as the primary abstraction. **Events** are regarded as any occurrence, process, wilful action or activity, or state, and can be identified, inter alia, from *verbs*[1]. Events may have **parameters** such as time and place, and events must relate to *referents*.

Referents are simply concepts denoted (or denotable) by a unique identifier, and may be entities (e.g. things, places, or roles) or may be other events. Referents are **participants** in the event and can be bound to the event in a finite set of *semantic or thematic roles*. Common thematic roles (participant types) include:

- *Agent*: *undertakes* the event
- *Patient*: *undergoes* the event
- *Instrument*: The means *used* to achieve the event.
- *Beneficiary*: directly benefits from the event

Spatial and temporal information for the events may be defined in *parameters* to the event:

- *Location*: the place of occurrence the event
- *Time* or *Period*: the time or interval of occurrence of the event. Events are durational and potentially interruptible: they may have a start and end -times or -events, and may be suspended or resumed.

Events may take other events as participants, in which case one event is said to *subordinate* another. For example:

- '*authorisation* to *read* the file' comprises an `authorise` event which subordinates a `read` event

[1] Although events are referred to by verb name for convenience throughout this document, events are more accurately referred to by verb sense identifiers as any given verb (i.e. word or symbol) may have multiple senses (i.e. meanings). Verb senses can often be identified via consultation of a lexicon such as WordNet [WN2000] - an approach demonstrated in COLOR-X [BvdR95]. Event *occurrences* (i.e. referents that are events) are actually instances of *event types* (*verb senses*), just as activity-occurences are instances of activities as in the *Process Specification Language (PSL)* specification [PSL2000].

- 'the *order will* be *despatched*' comprises a commissive speech act (i.e. an `oblige` event) which subordinates a `dispatch` event relating to an `order` event

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. It is for this reason that we use the linguistic term *referents*, rather than the term *objects*, to avoid association with object-oriented notions. Likewise we refer to *classifications* (or sets/named-groups) of referents, instead of to object-oriented *classes*. Uniquely identified *referents* may be role-players in events such as classification events (specifying the *type* of the referent), normative events (specifying what the referent *can* and *must* do), and others. **State** is determined by the currently applicable event bindings to a referent. **Behaviour** of a referent is determined by the *ability to* or *obligation to*:

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

These *abilities* and *obligations* are the *capabilities* and *responsibilities* of the referent, respectively.

Sections 3 and *4* describe how *descriptive* and *prescriptive* business rules may be specified using various core event types.

3 *Descriptive* Business Rules: Classification and Naming Events

Descriptive rules (descriptions) are declarative in nature and are implemented via `classification` and `naming` events.

3.1 Naming events

`Naming` is a common event which recurs frequently in specifications. Proper nouns, which name a specific referent, imply `naming` events.
The basic form of a `naming` event is:

```
[referent] is/must-be named [symbol/name]
```

3.2 Classification events

Referents which are similar in some respect may be grouped by `classification` (categorization) events. Classification events allow us to:

- create *classification hierarchies*. For example, "A customer is a person" means "A referent `classified` as a customer **`must`** be `classified` as a person".
- create *sets* or *named groups* of referents. These classifications associate a **name** with items that comply with a **description** or **set of criteria**. For example:

  Name: Wealthy Londoners
  are `classified` as
  Description: People with yearly income > £100k per year and telephone number beginning with 0207 or 0208.[2]

The party making the classification is implicit but may become relevant when diverse schemas need to be federated.

The basic form of a `classification` event is:

```
[referent(s)] classified as [type/kind/class/classification/category[3]]
```

Referents may be multiply typed (classified) and types (classification events) may be mutually exclusive.

More complex forms of `classification` event also exist. For example, classification of something as 'able', 'ready', 'the same as', 'different from', or 'a member of' is often with respect to some other (extensionally or intensionally defined) referents. The form of such complex classification events is:

```
[referent(s)] classified as
  able-to/ready-to/the-same-as/different-from/a-member-of/etc.
[referent(s)]
```

[2] This definition would more appropriately be atomized into separate definitions for 'wealthy' and 'Londoner'.
[3] Type (synonyms: kind/class/category) is a semantic role peculiar to classification events.

Events themselves are typed: typing of events is achieved by sub-ordination of an event to one or more `classification` events.

Type-determination 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 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 *operational event-history*.
2. Type determination based on *pattern of permissible future invocations* (derivable from *normative*[4] *event history*).

The **behaviour** of the type – its 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.

4 *Prescriptive* Business Rules: Constraints and Prescriptions

Prescriptive rules (constraints and prescriptions) are implemented using **normative events** – which are imperative - and **modal events** – which are declarative.

4.1 Normative Events

The closed set of *normative events* we provide includes `authorise`, `forbid`, and `oblige`[5]. `Authorisations` (permissions) and `forbiddances` (prohibitions) associated with a role are the **privileges** of the role – e.g. what the user *can* input or do. `Obligations` (`oblige` events) associated with a named role are the **responsibilities** of the role – e.g. what the user *must* input or do. Obligations may be immediate or bounded by a start and/or end -time or -event. **Rights** are taken in the broad business sense to include not only `authorisations` (*powers*), but also `obligations` in the referents favour (i.e. *entitlements*). For example, a buyer has the *right* to receive the product bought because the seller is `obliged` to deliver the product bought to the buyer (i.e. the `obligation` is in favour of the buyer – the buyer is the *beneficiary* of the `obligation`).

Obligations are imperative in nature. Examples of obligations in e-commerce include:

- interest and capital repayments on loans: obligation to pay
- sale: obligation of seller to deliver, obligation of buyer to pay (commitment to expenditure)
- lease payments: obligation to pay
- rebates: obligation to give discount when rebate ticket presented
- warranties and guarantees: A warranty (or warrant or guarantee) can be generalized as a *contingent obligation*: that is, an obligation that arises (i.e. is triggered) upon occurrence of certain events (contingencies). Warranties and guarantees generally have a lifespan. Examples of warranties are:
    - A 5-year product warranty may trigger an obligation to repair at no charge if the product is damaged during normal use within 5-years from the date of purchase (i.e. while it is 'under guarantee').
    - A profit warranty, as part of a share sale transaction, warrants specified profits during a certain financial period and the amount of the purchase consideration is contingent on those profits being achieved.
- pension obligations: obligation to pay regularly after retirement
- accrual of entitlements by employees, customers, or suppliers: for example, obligation to pay employees outstanding leave pay within 30 days of date of resignation from the firm
- obtaining qualifications: A referent may be obliged to fulfill specific requirements in order to obtain a qualification.

[4] Normative events are defined in *Section 4.1*.
[5] 'Authorise' is synonymous with 'permit' and 'allow'; 'forbid' is synonymous with 'prohibit' and 'disallow', and 'oblige' is synonymous with 'command'.

Obligations, like other events, are often *contingent*: they (i.e. `oblige` events) may be triggered by some uncertain future events. *Section 9.3* below describes the treatment of contingent obligations in business practice rules for constructing financial statements.

In all cases of permission, obligation, and forbiddance, the system may *prevent* violations, or may *detect* and act upon violations.

4.2 Modal Events

Modal events allow the formulation of descriptive (declarative) rules which describe the capabilities of referents as well as what is possible in the current domain or environment. The closed set of *modal events* provided includes `classify as able to` and `classify as possible`.

5 Rule Packets and Contracts

ILOG Rules [ILOG2000] allows rules to be gathered (i.e. `classified`) into *packets* for manageability. Packets can be *activated* or *deactivated*. Rules can only be violated if their packet is active. We extend the notion of packets by allowing descriptive and prescriptive events to be grouped into both regular packets, and into special packets to form e-commerce **contracts**. E-commerce contracts incorporate definitions (descriptive rules) and rights and responsibilities (prescriptive rules), and set forth the obligations and authorizations of the parties to the contract. Contracts come into force when the terms (i.e. rules) of the contract are agreed to (upon `contract` or `legal agreement`[6] events).

Table 1 below summarizes the common events that occur in business rule specifications and operational data. *Section 6* explains that both business rule events and operational events can exploit a common event infrastructure. *Section 7* describes how events may be exposed in English specifications of business rules.

Table 1: Common events in business rules

	Event (Active verb and synonyms)	Passive Form	Name (Deverbative noun)	
Events in Business Rules	**Descriptive (Describes)**			
	Names	is named	Naming	
	Classifies	is classified	Classification	
	Prescriptive (Prescribes)			
	Permits, Allows, Authorizes	is permitted, is allowed, is authorized	Permission, Authorization (*Powers*)	**Rights and responsibilities**
	Obliges	is obliged	Obligation	
	Forbids, Prohibits, Denies	is forbidden	Forbiddance (Negative Authorization)	
	Events Pertaining to Rule Packets			
	Contracts / Legally Agrees	is contracted to	Contract, Legal Agreement	
Operational Events	**Factual**			
	Occurs, happens	-	Occurrence	
	Captures, records	is captured, recorded	Capture	

6 Implementation: Benefits of Uniform Treatment of Business Rules and Operational Events in a Common Event Infrastructure

As was described in *Sections 3* and *4* above, descriptive and prescriptive business rules can be modelled as various types of event. These events can in turn be stored in an event store. Operational data is also stored in the event store, in the form of **factual events**. For the purpose of distinguishing between when an event *occurs* and when it is *captured*, we provide two *factual events*: `occurs` and `captures`.

[6] Note that agreements are not necessarily legal agreements, so we have explicitly qualified 'agreements' as 'legal agreements'.

So, instead of storing business rules in database stored procedures or triggers, or in application code, we propose to store rules as events in an event store. Aside from basic operations for creation, modification, removal, copying, and movement of events, the event store will also provide composite event detection, event firing, and event correlation services, as well as generic services for sorting and counting and other aggregate functions.

This uniform treatment of business rules and operational data allows the common event infrastructure to be used for processing both business rules and operational data. This means that processes (process information), data, and metadata can be seamlessly queried, altered, and managed, using a uniform infrastructure. Storing all policy and operations as events makes the software highly reflective. We therefore believe that the uniform treatment of business rule events and operational events may have significant advantages over storing business rules in triggers, stored procedures, or dispersed through application logic.

7 Exposing events in English business rule specifications

Events may be exposed from verbs in English business rule specifications. However, verbs are not the only lexicalisations of events, and events may be lexicalised, inter alia, as deverbative nouns, modals, adjectives, nouns, and cardinals.

7.1 Events exposed by deverbative nouns

Deverbative nouns are noun forms of verbs, and imply the existence of the event corresponding with that verb. Deverbative nouns can be detected in English specifications through indicative suffixes (such as -ion, -ment, -ent, -ure, -ance, -ence, -ancy, -ency, -ing, -ness, -al, or -y). For example: `subscription` implies a `subscribe` event, `government` may imply a `govern` event, `failure` implies a `fails` event, and `approval` implies an `approves` event.

7.2 Events exposed by modals

Modals (e.g. modal auxiliaries and suffixes) indicate the existence of modal or normative events. In English, modal auxiliaries include lexical items like 'can', 'must', 'have to', 'should', and 'will'. Certain suffixes (e.g. -able and -ible) are also normative or modal. For example, one (normative) sense of 'readable' implies '[actor] `authorises` to read'; for 'interruptible' there is an implied '[actor] `authorises` to interrupt'. Another (modal classification) sense of 'readable' is '[actor] is `classified as able` to be read [patient]' or 'reading of [patient] by [actor] is `classified as possible`'. 'Payable' may imply that a referent is `obliged` to pay; e.g. 'the account is payable' means 'the account holder is `obliged` to pay the account'.

Table 2 below illustrates how the common semantics of the English modals can be defined in terms of the prescriptive (*normative* or *modal classification* events) events defined in Section 4, or in terms of *detection* and *induction*. Note that, in any given sentence, a modal can have one or more semantics: that is, it is possible for a modal to simultaneously have multiple meanings. This is why English modals are considered vague.

Table 2: Event Semantics of Modal Auxiliaries and Suffixes in English

Modal Auxiliary or Suffix	Possible Event Semantics
Can, May	▪ … `permits` … to [event*] ▪ … `classifies` … as able to [event] ▪ … `classifies` [event] as possible
Cannot, May not	▪ … `forbids` … from [event] (i.e. *prevention*) ▪ … `classifies` … as unable to [event] ▪ … `classifies` [event] as impossible. Usually this implies that the system must also `verify` that [event] has not happened
Could, Might	▪ … `permits` … to [event] **if** … (i.e. *conditional/contingent* permission) ▪ … `classifies` as able to [event] **if** … (i.e. *conditional* ability) ▪ … `classifies` [event] as possible **if** … (i.e. *conditional* possibility) ▪ past tense of 'can' (i.e. denotes `permit` or `classify` events that have occurred before the utterance or usage of the policy)
Must, Shall, Should, Ought, Will, Has-to / Have-to, Ensure that, Needs to	▪ … `obliges` … to [event] ▪ … `classifies` [event] as necessary ▪ The system must `verify` … that [event] has happened (i.e. obliged *detection*) ▪ The system must `deduce` … that [event] has happened (i.e. obliged *induction*[7])
Should have	▪ past tense of 'must'
Must not, May not, Shall not, Should not, Ought not, Will not	▪ … `forbids` … from [event] ▪ The system must `verify` that [event] has not happened
Should, Would	▪ … `obliges` … to [event] **if** … (i.e. conditional obligation)
-able, -ible	▪ … `permits` … to [event] ▪ … `obliges` … to [event]
Does, is (factual modality)	▪ … `happens / occurs` …

* Note that all events/actions here are possibly parameterised (i.e. complex events/actions)

7.3 Events exposed by nouns and modifiers

Nouns and *modifiers* - such as adjectives, adverbs, adjectival, and adverbial phrases - may imply `classification` or `naming` events.

7.4 Events exposed by cardinals and measures

Cardinals numbers (e.g. '3') and *measures* (e.g. '3 metres') often imply `quantification` events.

[AB2000] provides a more complete description of techniques for exposing events in English specifications of e-commerce applications.

8 Business Rules (Policy) Life Cycle

The term 'business rules' in synonymous with the term 'policy'. [Wies95] explains that the policy life-cycle consists of: policy definition (creation), policy transformation (high-level policies transformed into low-level policies), policy activation, policy monitoring, policy pausing and resumption, policy change or adaptation, policy enforcement, policy deactivation, and policy deletion. We supplement these notions with the following additions:

- We distinguish between policy *approval* and policy *application* as approval and activation are separate events. The window of applicability of a policy (i.e. the interval between the policy 'applying' and 'ceasing to apply') may have an upper and lower bound, each of which may be delimited by a date-time stamp or an event. For example, a policy may be applied directly after the approval event of that policy, or application of the policy may be deferred to a later time or event.
- We term policy transformation as policy *elaboration* as we feel this is more descriptive. Policy elaboration involves linking high-level policies (i.e. the 'what') to the lower-level policies that enforce that policy (i.e. the 'how'). This may occur at multiple levels.
- During monitoring, a policy may be *violated* or *complied with*. Actions may *violate* or *comply with* policy, or their status may be *unknown* if they have not yet been assessed against policy. When a policy is violated a

[7] Obliged induction relates to **necessity** (i.e. something that is **necessarily true**). This is the 'NEC' relationship in COLOR-X [BvdR95].

violation event is signalled, whose participant is the policy being violated. A violation may be *respond*ed to by the system or by users. The violation may be *resolved* or *outstanding* (unresolved).

- We distinguish a number of ways in which a policy may *cease* to apply (i.e. be deactivated). For example, policies may *lapse* or be *revoked* by an empowered 3rd party.
- We allow for policies to be *grouped* into (`classified as being a member of`) **packets** for manageability, as described earlier, in *Section 5*.
- Support for conflict. If several rules meet their conditions at the same time, we provide for a determination of the rule with the highest precedence, which may *suppress* (i.e. temporarily *forbid*) other conflicting or non-conflicting rules from firing (i.e. being enforced). In ILOG Rules [ILOG2000], rules can be assigned a static or dynamic priority; the rule with the highest priority is fired first.

Table 3 below demonstrates the events and states that pertain to the business rules (policy and packet) life cycle.

Table 3: Business Rule (Policy and Policy Packet) Events Through the Business Rule Life Cycle

Policies or packet events	Policy or packet states
Policy or packet `created` / `defined`	Policy or packet `exists` (but not yet active)
Policy `elaborated`	
Policy `classified as member of` (grouped into / assigned to) packet	
Policy or packet `changed` e.g. permission `transferred`	
Policy or packet `approved`	
Policy or packet `applied / activated`. In the case of permissions this is referred to as `granting`.	Policy or packet `active` / `enforced` / `in force`
Policy or packet `paused`	Policy or packet `inactive`
Policy `violated` (Creates a `violation` event)	
Policy or packet `resumed`	
Policy or packet `de-applied` / `deactivated`	
Policy or packet `deleted`	

Table 4 below shows events relating to policy `violations`.

Table 4: Policy Violation Events

Violation events	Violation states
`Trigger` violation (Policy `violated`)	Violation outstanding
`Respond to` violation	
`Resolve` violation	Violation resolved

Obligations are a special type of policy and certain peculiarities apply to the life cycle of obligations. These supplementary events, which are summarized in *Table 5* below, are:

- Obligations may *arise*: that is, be *imposed* or *incurred*.
- Obligations may *fall due*. An obligation that has arisen (as a result of events) and fallen due (as a result of the passage of time or occurrence of events) must be *satisfied* within a certain time or before the 'due by' date-time or event, otherwise the obligation is *violated*.
- An obligation may be *settled* (i.e. *fulfilled*, *discharged*, or *complied-with*). The International Accounting Standards Committee Framework [IASC99] lists a number of business events that may result in the settlement of an obligation, including: payment of cash, transfer of assets, provision of services, replacement of the obligation with another obligation, or conversion of the obligation to equity. An obligation may cease if it is completely fulfilled or may *continue* if it has been partially fulfilled.
- The obligation may *cease* to apply in a number of ways:
    - The obligation may be *completely fulfilled* (fulfilled in its entirety)
    - The obligation may be *extinquished*. Again following the terminology of [IASC99], extinguishment of an obligation may be via the waiver or forfeiture of rights (obligation *waived*) by the obliging party, or by the removal of rights by an empowered 3rd party (obligation *cancelled*).
    - The obligation may *lapse* or *expire* as a result of the failure of another party to fulfill mutual obligations within a given time: for example, an obligation of a buyer to pay may lapse if the seller has not fulfilled their obligation to deliver within 15 days of the placement of the order.

- The obligation may be *over-ridden* by another obligation that has precedence.

Table 5: Supplementary Policy Events Through the Life Cycle of Obligations

Obligation events
(Contingent) Obligation `arises`: `incurred` or `imposed`
Rights `ceded`[8] (i.e. *beneficiary* of obligation `changed`)
(Contingent) Obligation `falls due`. (Previously-incurred obligation enforced, often conditional upon circumstances).
Obligation `settled` / `fulfilled` / `discharged` The obligation will cease if completely fulfilled, or may continue if partially fulfilled.
Obligation `ceases` ▪ Obligation `settled` / `fulfilled` / `discharged` ▪ Obligation `extinguished` o Obligation `waived` / Rights `forfeited`, `surrendered`, `relinquished` or `ceded` by beneficiary o Obligation `cancelled` by empowered 3rd party ▪ Obligation `lapses` / `expires` ▪ Obligation `over-ridden`

As previously mentioned, contracts are named packets of descriptive and prescriptive events. Certain additional events may apply to contracts which determine their state and the state of the rules contained within them. *Table* 6 below summarizes events pertinent to contracts.

Table 6: Supplementary Policy Packet Events Through the Life Cycle of Contracts

Contract events	**Contract states**
Contract `created`	Contract `exists`
Contract `signed`	
Contract `legally agreed to`	Contract `in force` / `entered`
Contract `nullified`	Contract `void`

9 Business rules in e-commerce, categorized by rule type

This section provides some examples of various types of business rules that can be decomposed according to the event-centric paradigm. Rules can be divided into two major rule categories:

- **Business** rules (management rules, business practices, access rights, knowledge/ontology, etc.). Business rules are taken to include **organizational** rules, policies, and practices, as well as or **statutory**, **regulatory, communal**, or **inter-organizational** laws, standards, practices, policies, or contracts.
- **System** rules (e.g. monitoring and control, transactions, consistency of distributed data, resource-utilization, reliability/availability/failover, recovery, access control, etc)

A business user should be shielded as much as possible from system rules such as when to move data from memory to disk or memory to archive, or when to put locks on data to prevent concurrent access.

The business and system rules demonstrated in this section are categorized into various sub-types of rule which, again, makes management and administration easier. These sub-types are partially overlapping. Domain-specific events are highlighted in `Courier` type-face. Modals, which expose prescriptive events, are highlighted in **`Courier Bold`** type-face.

9.1 Management Rules.

Management involves planning, leading, organizing, monitoring and control[9].

- Planning, Leading, and Organizing (*proactive*):

[8] 'Cede' has two verb senses: the first involves transferring rights, the second involves only surrendering or waiving rights.
[9] As before, these categories are partially overlapping: rules may be classified into more than one of these categories.

- Management by delegation: *Delegation* involves the allocation of *resources* to *tasks* - *tasks* are events that referents (delegates) are `authorised` to and `obliged` to perform on managed items.
- Routing (data flow; distribution): e.g. "Forward credit applications over $10,000 to the branch manager"
- Control flow. Control flow can be achieved through dynamically triggering `authorisation`, `obligation`, and `forbiddance` events.
- Workflow is achieved through a combination of control flow and data flow.

- Monitoring and Control (*reactive*): Monitoring involves detecting both activity and inactivity often by comparing *actual* behaviour to *desired* parameters. Control involves the monitoring service responding to detected events or non-events. Monitoring and control rules include:
  - Business risk management and control rules: Discussed in Section 9.3.
  - Monitoring for inactivity (non-occurrence of events): E.g. "The user **`must`** be `logged out` after 10 minutes of inactivity and their shopping basket `cleared`".
  - Monitoring for failures or exceptions: Detecting `failures` or `exceptions` is important as rules may need to be triggered to, for instance, back out of transactions or initiate alternative courses of action.

9.2 Access Control Rules (Security and Permissions)

Access control rules describe who can access what and under what circumstances, and who can grant `permissions`. Access control rules may be defined extensionally in terms of *individually specified users* or intensionally in terms of *user roles* or *users fulfilling certain criteria* (i.e. named `classifications` or sets of users). E.g. "Before the customer **`can`** `access` the Investment Advice page they must `register`" which can alternatively be stated as "Only `registered` customers **`can`** access the Investment Advice page".

9.3 Business Practice Rules

Business practice rules include:

- Business risk management and control rules (which may involve monitoring and control), such as: "The `purchase order` **`must`** be `approved` by 2 managers and a project manager", "All orders over $100 **`must`** be `credit-checked`". Often, to limit data consumption companies may institute policies like "After 3 `requests` for financial broker information within 24 hours, `deny` the user further `requests`". A unit trust company with a large independent broker network could use this policy to prevent competitors from culling large amounts of broker information, whilst allowing clients to conveniently find brokers close to them.
- Pricing rules. For example, fare rules in the airline industry are quite complex and describe how customers `qualify` for special fares. An example of complex fare rules from *travelocity.com* [Tra2000] is: "`Reservations` must be made no later than 7 days before `departure` from origin. Return `travel` **`must`** `commence` no earlier than 12:01am on the first Sunday after `arrival` at the last stopover point and no later than 4 months `measured` from `departure` from origin to `departure` from last stopover point. 67% of the adult fare is `charged` for a child passenger, 2 through 11 years of age; 33% is `charged` for an infant younger than 2 years, when `accompanied` by an adult. No enroute `stopovers` are permitted. A `stopover` `occurs` when the passenger does not `depart` an intermediate point within 24 hours of `arrival` at that point". Notice how 'child passenger' means 'a passenger `classified` as a child', and that the apposition '… ,2 though 11 years of age, …' implies that 'a passenger between 2 and 11 years of age **`must`** be `classified` as a child'.
- Terms of service rules. E.g. "[**`Must`**] `Despatch` all products in `order` within 24 hours of `order`".
- Formulae, Algorithms, Measurement bases, and Standards. Formulae and algorithms are common in financial calculations. E.g. "Yearly salary = Weekly wage rate x 52 weeks" or "Leave days accrued for human resource year = 1.2 x months since start of human resource year". In the case of financial statements, measurement bases (rules for measuring past economic events) are defined by the relevant International Accounting Standards. For example, financial statements must account for contingent obligations, which are `obligation` events that may arise (i.e. be triggered) as a result of some previous `contracting` event that was entered into. It is therefore possible to quantify both present obligations and potential obligations of the firm. As the occurrence of the future events on which the potential obligations is uncertain, often probabilistic measures are used as the basis for these calculations.

Examples of the use of obligation events during the preparation of financial statements can be regularly found in the International Accounting Standards:

- IAS33 (1997) [IASC99] specifies that Basic Earnings Per Share is calculated as "Net profit/loss attributable to ordinary shareholders, divided by the weighted average number of ordinary shares outstanding during the period". In contrast, Diluted Earnings Per Share is calculated based on the number of (dilutive) potential ordinary shares. A potential ordinary shares is defined as "a financial instrument that may *entitle* the holder to ordinary shares" [emphasis added]. This `entitlement` is a contingent `obligation` that triggers upon the occurrence of certain events.
- IAS33 (1997) [IASC99] defines a warrants or option as "a financial instrument that give the holder the *right* to purchase ordinary shares" [emphasis added]. This right is a contingent `obligation` (of the company to issue ordinary shares to the option-holder) that triggers upon the exercising of the option.

9.4 Data integrity rules.

Data integrity rules include:

- Concurrent access rules: "An employee **`cannot`** `access` a customer record while another employee is `modify`ing it. However, if the second employee has made no `change`s in the past 10 minutes, then `permission` to `modify` is `transferred` from the second employee to the first."
- Input validation rules: Checking that inputs are *syntactically* and *semantically* correct (valid) and `classify`ing data as valid or invalid. So, **validation rules** describe complex *conditions that* **`must`** *be fulfilled* before an input can be `classified as valid`. e.g. Dates may be syntactically correct if they conform to a certain mask such as "dd/mm/yyyy", whereas they may be semantically invalid (e.g. "30 Feb 1999"). Configuration programs often require complex validation where fields are only valid in certain combinations: for example 70Hz frequency is valid with a 1024x768 resolution, but invalid with a 2000x1500 resolution.
- Dependency rules: Ensuring that information is always accompanied by dependent information. E.g. "`Orders` **`must`** have a valid customer."
- Transactions rules: "Before a transaction is `confirmed`, **`ensure`** that a correct `copy` of it has been made on a secondary server".

9.5 Display rules

Display rules may pertain to:

- Pages: e.g. page format, standard page and document headers and footers, standard document background image and colour, etc.
- Fields: required field format, field type (e.g. drop-down list, radio-button, check-box, text field), colour, font, sort order, and display conditions which specify under what circumstances fields **`must`** and **`mustn't`** be displayed

Currently, default display rules (standards/policies) are often incorporated in Cascading Style Sheets (CSSs) [W3C98] in e-commerce applications. These defaults are applied to (i.e. enforced on) all documents that incorporate links to the style sheet. Defaults can be specified in order of preference; this prioritisation of rules allows, for instance, a second choice (e.g. 'Arial' font) to be enforced when the first preference (e.g. 'Alburtus Extra Bold' font) is not possible (i.e. not available).

9.6 Ontological rules

Ontological rules relate to what was typically considered as the static object model (classes, relationships, attributes, and methods), but are now modelled entirely in terms of events. Ontological rules relate to:

- Categorisation into sets or classes: e.g. "Employees and customers are [`classified` as] people". Traditionally this was modelled as a 'is-a-type-of' or 'is-a-kind-of' relationship in entity relationship diagrams, but is now modelled as a classification event. Mutually exclusive (disjoint / non-overlapping) sub-types can be enforced by preventing conflicting `classification` events.
- Definitions: "A working week `starts` on a Saturday and `ends` on a Friday", "The financial year `starts` on 1st October and `ends` on 30th September", "25 December is [`named/called`] Christmas".
- Qualification criteria: Referents may need to meet criteria – i.e. have been participants in certain events or be in a certain state – in order to be awarded a `qualification`, which may bring with it certain

rights and responsibilities. In an online campus, students are obliged to complete (i.e. fulfil the requirements of) all courses in the 'prescribed curriculum', in order to obtain a qualification, which may give them the right to practice in a specific field.

- Mereology (parts and wholes): e.g. "All people have names and addresses" Traditionally this was modelled as a 'is-a-part-of' relationship in entity relationship diagrams. The event centric paradigm uses stative events like `has`, `contains`, and `includes` and their inverse 'is `classified` as part of' to model parts and whole
- Possession (ownership and power): e.g. "Marketing administrators own marketing data"
- *Capabilities* of referents or sets (classes) of referents: E.g. "[referent(s)] `classified as able to`"
- *Rights* or *powers* of referents or sets (classes) of referents. E.g. "[referent(s)] `is allowed to / forbidden from`"
- *Responsibilities* of referents or sets (classes) of referents: E.g. "[referent(s)] `is obliged to`"

Ontological rules often involve **stative events** (**relationships** or **relations**) like `have` and `contain` and `own` which do not involve action. Stative events are the result of active events[10], for example:

- an '`is [classified as] a type of`' stative event is the result of a related '`classify`' event.
- an '`is [classified as] a part of`', '`contains`', '`has`', or '`includes`' stative event is a result of a related '`joins`', 'is `allocated`', or 'is `added` to' event.
- an '`owns`' stative event is the result of a related '`acquires`' (e.g. '`buys`', 'is `given`', '`makes`', or '`creates`') event.

9.7 Filtering rules

Filtering is usually based on selecting referents that have been subject to `classification` events. For example, in "Show a list of all `expired` products", the filter is dependent on the definition: "A product `expires` [i.e. and **`must`** then be `classified` as 'expired'] when it has passed its sell-by date". Other examples of filtering rules are: "Show a list of all `pending` tasks", "Show a list of all tasks `completed` in the past 10 days", etc.

Filters are useful for defining named sets of referents (so-called *intensionally* defined sets), where the contents of the set may be changing at run-time, but the developer wishes to be able to refer to the individuals in the set by a stable name or set of criteria. The ability to define such named sets allows policy definitions to remain stable even as the managed entities change [DDLS2000].

9.8 Content personalization rules

Many popular e-commerce application servers offer *content personalization* rules. These products generally incorporate:

- Functionality or guidelines for manually tagging content (manual `classification`)
- User segmentation rules for automatically `classifying` users (e.g. 'high earner = user.income > $100,000')
- Rules for automatically `classifying` content
- Rules for content `selection`: these rules allow particular content to be displayed to specific users during defined times and under defined circumstances. For example "display previous week's business highlights to users classified as high earners on Monday mornings".

Commercial products that include support for content personalization rules include: ATG Dynamo [ATG2000a, ATG2000b], BEA Weblogic [BEA200a, BEA200b], Microsoft Site Server 3.0 and Commerce Server 2000 [Mic2000a, Mic2000b]. These products generally implement content personalization rules via a combination of XML (for tagging) and hooks to an existing object model.

9.9 Reliability, performance, scalability, and distribution (replication) rules.

Rules are often used to improve reliability, performance, and scalability. For example:

- Intelligent caching to speed up page availability. Rules can, for instance, be applied to regenerate data-based pages that are based on infrequently updated or firm (unchanging) data, on a scheduled basis. This

[10] These implicit active events, that are the cause of (or reason for) the related stative events, may or may not be relevant. For example, we may not care how a referent came to `own` something – so we may need to know only of the stative '`owns`' event and not of the preceding '`buys`' or 'was `given`' events which caused the ownership.

could apply in an online news application where it is more efficient to generate the day's headlines from the news database early each morning, rather than requerying the database with each access. Similarly, in the unit trust industry, static pages containing monthly unit trust pricing graphs or share holding statements – data which is firm at month end and not subject to change – can be generated at the end of every month, rather than repeatedly querying the database with each user access, which would impact negatively upon database performance.

- "If a server `fails` or is `overloaded`, `redirect` the user to another available server".
- "`Replicate` the customer information nightly to a second server in London so that management reports can be run off that server"

9.10 Resource utilization and cleanup rules.

- Pre-utilization rules: "`Lock` data before update".
- Post-utilization rules: "`Archive` customers who have not `logged in` for 6 months".

10 Related Work

Our approach relies heavily on linguistics and bears many conceptual similarities to the COLOR-X requirements definition approach [BvdR95] in this respect[11]. COLOR-X's VERP demonstration program allows the definition and graphical display of a CASE model. VERP is supported by interaction with the WordNet [WN2000] lexicon and a supplementary, custom-written linguistic database. The COLOR-X notion of *relations* maps to our concept of *events*, COLOR-X *classes* and *objects* map to our concept of *classifications* and *referents* respectively, and COLOR-X incorporates *thematic roles* (which they term *semantic functions*), *parameters* (which they term *satellites*), and *modalities*. However, the end-goal of COLOR-X is the generation of *object-oriented code* (classes, attributes, and methods) from a conceptual schema or functional specification, whereas we propose the generation of various types of *events* to be stored and animated in an event store. COLOR-X is not a web-centric technology and provides no inbuilt support for HTTP and SMTP, which we plan to provide to support the rapid development of e-commerce applications.

11 Conclusion

This position paper discusses work-in-progress. We plan to begin an implementation of an event-centric business rule definition and execution environment in November 2000. We have gathered functional specifications for a variety of e-commerce applications. Examples are listed in Table 7 below.

Table 7: E-commerce application specifications gathered

Online betting and gaming	Household insurance
Retail	E-currency
Sales incentive system	Personalization for online investors
Skills management system (intranet), and job-candidate matching service (internet and extranet)	Online personalized broadcast news and news dissemination
E-procurement (RFQs and reverse auction)	Delegated regional content management
Online campus	Lunch ordering (intranet)
Time/project/task management	Billing system for medical aid message switching
Fleet management	Airline flight booking
Online unit trusts	Demographic database for subscriptions

We plan to develop an application server for this definition and execution environment that incorporates a rules engine with HTTP (web) and SMTP (email) capabilities to demonstrate the viability of the our approach for specifying executable e-commerce applications using event-based decompositions of business rules.

12 Acknowledgements

This research has been made possible by generous grants from the *Cambridge Commonwealth Trust*, the *Overseas Research Student Scheme*, and the *University of Cape Town Postgraduate Scholarships Office*. Acknowledgement is also due to the *Dimension Data iCommerce Internet Services* for their provision of numerous real-world electronic commerce specifications that are being used for validation of the proposed architecture.

[11] COLOR-X is based on Functional Grammar.

13 References

[AB2000]
Abrahams AS, Bacon JM. *Event-centric Policy Specification for E-commerce Applications*. Under review for Policy Workshop 2001, Bristol, 2001.

[ATG2000a]
Art Technology Group. *Personalization Server: Personalization Control Center*.
Available at: http://www.atg.com/products/dps/pcc.html
Accessed on: 10 July 2000

[ATG2000b]
Art Technology Group. *Dynamo Personalization Server Version 4.5.1 Overview*.
Available at: http://www.atg.com/support/documentation/ (restricted access)
Accessed on: 10 July 2000

[BEA2000a]
BEA. Creating and Managing Rules. Available from http://edocs.bea.com/wlcs/p13n/rules.htm.
Accessed on: 15 Aug 2000.

[BEA2000b]
BEA. Working with Rules. Available from http://edoc.bea.com/wlcs/p13ntour/mbbrules.htm. Accessed on: 15 Aug 2000.

[BvdR95]
Burg JFM and van de Riet RP. *COLOR-X: Object Modeling Profits from Linguistics*. In Proceedings of the 2nd International Conference on Building and Sharing of Very Large-Scale Knowledge Bases (KB&KS'95), Enschede, The Netherlands, 1995

[DDLS2000]
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.

[IASC99]
International Accounting Standards Committee. *International Accounting Standards 1999*. London, United Kingdom. 1999.

[ILOG2000]
ILOG Rules White Paper. Available from http://www.ilog.com/
Accessed on: 15 August 2000

[Mic2000a]
Microsoft. *Microsoft SiteServer 3.0*.
Available at: http://www.microsoft.com/siteserver/
Accessed on: 1 July 2000

[Mic2000b]
Microsoft. *Microsoft Commerce Server 2000*.
Available at: http://www.microsoft.com/commerceserver/prodinfo/features.asp
Accessed on: 25 July 2000

[PSL2000]
Schlenoff C, Gruniger M, Tissot F, Valois J, Lubell J, and Lee J. *The Process Specification Language (PSL) Overview and Version 1.0 Specification*. National Institute of Standards and Technology. Available from: http://www.mel.nist.gov/psl/

[Tra2000]
Travelocity.com. http://www.travelocity.com/
Accessed on: 6 September 2000

[W3C98]
World Wide Web Consortium (W3C). *Cascading Style Sheets, level 2 CSS2 Specification*. 12 May 1998.
Available from: http://www.w3.org/TR/REC-CSS2/
Accessed on: 7 September 2000

[Wies95]
Wies R. *Using a Classification of Management Policies for Policy Specification and Policy Transformation*. Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management. Santa Barbara, California, USA, 1-5 May 1995.

[WN2000]
WordNet. *WordNet: A Lexical Database For English*. Cognitive Science Laboratory, Princeton University.
Available from: http://www.cogsci.princeton.edu/~wn/
Accessed on: 20 June 2000