BEEO: Semantic Support for Event-Based Data Analytics

Michele Ciavotta, Vincenzo Cutrona, Flavio De Paoli, Matteo Palmonari, and Blerina Spahiu(✉)

University of Milano-Bicocca, Milan, Italy
{michele.ciavotta,vincenzo.cutrona,flavio.depaoli,matteo.palmonari, blerina.spahiu}@unimib.it

Abstract. Recent developments in data analysis and machine learning support novel data-driven operations. Event data provide social and environmental context, thus, such data may become essential for the workflow of data analytic pipelines. In this paper, we introduce our Business Event Exchange Ontology (BEEO), based on Schema.org that enables data integration and analytics for event data. BEEO is available under Apache 2.0 license on GitHub, and is seeing adoption among both its creator companies and other product and service companies. We present and discuss the ontology development drivers and process, its structure, and its usage in different real use cases.

Resource Type: Ontology
License: Apache 2.0
DOI: https://doi.org/10.5281/zenodo.4695281
Repository: https://github.com/UNIMIBInside/Business-Event-Exchange-Ontology

Keywords: Event data · Business events · Custom events · API · Ontology · Data analytics

1 Introduction

Events can be defined as changes happening at a given time and in a given (physical or virtual) environment, where some actors take part showing some action features [13]. Since events and the data traces they generate describe the behavior of humans and machines, event data are becoming essential in everyday applications in multiple domains. Examples of events that are tracked and used by computer applications include clicks on Web pages, changes in product prices, marketing campaigns, log records of software applications, and health check-up records. Leveraging event data to derive insights is crucial to make effective decisions in several contexts, e.g., for advertising, human and computing resource planning, price strategies, therapy prescriptions and so on.

A. Hotho et al. (Eds.): ISWC 2021, LNCS 12922, pp. 580–596, 2021.
https://doi.org/10.1007/978-3-030-88361-4_34

With the increasing uptake of data-driven decision making and automation in the industry, often powered by data analytics and machine learning, event-based analytics is providing a unique opportunity to develop and optimize data-driven business services in a vast variety of business domains. Many examples of these services can be found in the large number of companies operating in the eCommerce, Retail, Customer Relationship Management (CRM) and Digital Marketing industries. These companies collect large amounts of data about customers at different touch points across the so-called consumer journey (from need recognition to purchase and customer support) [9]. All these companies are part of a complex value constellation, where their data-driven business-to-consumer and business-to-business services generate high business value.

Consider, for example, a company running a CRM application, which serves a client company launching a promotional campaign. Upon campaign launch, the number of customers' requests served by the CRM company are likely to peak, requiring supplementary resources to be timely allocated. As another example, consider resource planning at retail. Foot traffic is known to increase after events like paydays, holidays, or promos for specific products. In event-based data analytics, companies that have data tracking their own Key Performance Indicators (KPIs), e.g., served requests in CRM, foot traffic count in retail, need to enrich their records with event data so as to train predictive models to estimate the impact of events on their own KPIs and act accordingly (e.g., improve resource planning). Similar services have been developed in a production environment as part of the EW-Shopp EU project,[1] whose goal was to develop technologies to facilitate, with the help of semantics, the creation of these services, with specific attention to the simplification of the event-based data enrichment processes required to support weather and event-based data analytics [4].

In these contexts, the semantics is a key factor because usually event data are not generated internally, but originate from third parties. In the CRM example, the company launching the campaign would generate the data, while the CRM company would use them. Similar scenarios occur in the retail case. Event data exchange across parties is relevant to the domain of event-based data analytics, which explains why supporting semantic interoperability in this context is compelling to streamline the development of data-driven services. Semantic vocabularies and ontologies to define events have been developed. In particular, event modeling has been significantly investigated in the past [1,8,17], but often proposing event ontologies that provide complex representations (e.g., nested descriptions) [1], which, although useful in specific domain contexts, may result too complicated to support the practical tasks of event-based data enrichment (served well by flatter representations). The representation of event data is also very relevant for representing event data on the web and, in fact, Schema.org provides a vocabulary to describe events[2] that achieve a good trade-off between richness and simplicity. A question is, therefore, whether Schema.org covers business needs addressed in the depicted event-based analytics scenario.

1 https://www.ew-shopp.eu/.
2 https://schema.org/Event.

Stemming from this question, in this paper, we present a Business Event Exchange Ontology (BEEO) that addresses the need of harmonizing the description of events provided or used by EW-Shopp partners to enrich information about proprietary data describing a business phenomenon of interest. As a matter of fact, none of the existing ontologies or vocabularies fully covers the aspects that emerged from the requirements analysis carried on within the project, and Schema.org is the resource covering most of the necessary general concepts. BEEO is, in fact, an extension of the Schema.org vocabulary that covers event representation relevant to support data analytics in several domains such as eCommerce, Retail, CRM and Digital Marketing industries. We publish BEEO in Turtle format under a public license to support further extensions. In addition, as a consequence of the requirements collected from business partners, we have developed JSON-LD-based[3] APIs to support event-data exchange based on the proposed ontology. We found JSON-LD to be appreciated also by practitioners working in the industry who are not familiar with pure RDF-based technology. The ontology as well as the APIs have been agreed upon by software engineers of different companies and tested in real business environments. For example, a data enrichment service has been developed for ASIA,[4] a semantic table annotation application that supports data enrichment at scale [3].

The contributions made in this paper can be summarized as follows: (i) we present the methodology used to design and publish BEEO; (ii) we make the ontology available under an open license; (iii) we present the APIs developed to support event data exchange and usage; and (iv) we show different use cases where this can be applied, one of which is explained more in detail.

This paper is organized as follows: In the next Sect. 2 we discuss the requirements and adopted methodology; Then in Sect. 3 a detailed description of BEEO is provided; Sect. 4 presents the API along with an example of use; Sect. 5 compares BEEO with other event models; Finally, Sect. 6 draws some conclusions and outlines future work.

2 Requirements and Methodology

For the development of the ontology, we applied common techniques recommended by well-established ontology development methods [14]. We used a bottom-up approach by identifying the scope and user group of the ontology, requirements, and ontological and non-ontological resources.

2.1 Requirements

The primary resources used during the development of the ontology were company data provided by the industrial partners of the EW-Shopp project to be harmonized to allow for further processing with the tools developed during the

[3] https://json-ld.org/.
[4] https://github.com/UNIMIBInside/asia-backend.

project. In the following, we provide a brief description of each business case and discuss the general requirements for the ontology. Technical requirements on the API to access and use the ontology will be discussed in Sect. 4. Table 1 reports examples of interactions (queries/responses) that the ontology-mediated API should support for each business case.[5]

Brand Performance Insights. Ceneje[6] provides an ecommerce search engine and a comparison-shopping platform to make users' shopping experience smarter, and to provide their business-to-business partners with deep insights into consumer past behavior and predictions on future behavior. Ceneje collects information about users' searches and clicks on vendors' offers (performance indicators of the advertised products) and analyzes their evolution at different levels of aggregation (SKU, brand, category, vendor, price). Such data need to be enriched with business events data, tagged for different category, brand, and marketing segments, to support a predictive service that estimates how the market reacts to certain business events (e.g., price changes, marketing campaigns, new product introductions) to add real value in the decision-making process. Examples of data challenges concern the representation of concepts around the notion of *price changes* and the possibility to support internal identifiers for products, which may be different from the official ones. Other concepts, such as *SKU*, *EAN code* or *seller* are already present in shared vocabularies like Schema.org.

Weather and Event-Aware Business Intelligence for the Optimization of Campaigns and Human Resources. BigBang[7] is a retailer company in the segment of Consumer Electronics and Home Appliances with on-line and 18 physical stores. The stores importance can be measured by the number of visitors or/and employees on the floor. Running a prediction service to help decision-makers in optimizing sales-force and marketing communication planning with estimations of the daily number of visitors requires company data (e.g., number of on-line and in-person visitors) to be enriched with internal business events (e.g., marketing campaigns grouped by channel), and external events (e.g., suppliers' marketing campaigns). Specific data requirements in this case concern the representation of the *marketing channels* and *price discount* concepts that are not present in shared vocabularies or ontologies.

Workforce and Campaign Management Optimization. Browsetel/CDE[8] provides clients with services for Customer support, CRM and Customer Engagement Management. The primary goal is to optimize the system resources (the number of agents working in a campaign that serves specific topics of the client). In order to predict interaction traffic, historical data recorded by the system and weather and (custom) event data from external sources have to be integrated.

[5] More details are available at https://github.com/UNIMIBInside/Business-Event-Exchange-Ontology.
[6] https://www.ceneje.si/.
[7] https://www.bigbang.si/.
[8] https://www.cde.si/.

Custom events are generated by clients (e.g., the launch of a new product on the market) and used by the support service to optimize the workforce (e.g., predict the number of agents needed on the floor). Examples of data challenges are the representations of the *source* (e.g., the client) and the *size* (e.g., the potential customers involved) of an event.

Event and Weather-Aware Foot Traffic Predictions and Analytics. Measurence[9] provides retailers with devices to sense and count people in and around their physical location and services that exploits such information to support retailers' decision making about the best time for marketing campaigns. The use of business events is crucial to enrich the collected data, understand the past customer flows, and enable for reliable predictions for future marketing events. The data challenge here is to distinguish between the number of people interested (e.g., registered attendees) and attending (e.g., actual participants) an event. Even this simple pair of concepts are surprisingly not represented in generic vocabularies such as Schema.org.

General Requirements. The data provided by the above companies were analyzed to determine the scope and requirements for the ontology. The analysis led to the identification of the major concerns that the ontology should address. The overall requirement was to support the integration of all data provided by at least one data provider modeled in different ways under a single representation schema.

Events were classified by their size, type, and context to enable a more efficient integration. Starting from the general concept of event, the main requirement that emerged was the need to capture the concept of marketing event. For the use in the project business cases, a lightweight specification of the ontology as a vocabulary is sufficient (properties, classification schemes, etc.), even if an OWL specification may be desired but not strictly required. A requirement that was strongly supported by the industrial partners was to reuse existing ontologies as often as possible, to reduce effort and promote the use of integrated data. For the design of BEEO, we considered all the above requirements as MUST[10].

2.2 Methodology

We developed BEEO by following one of the most recent ontology design methodologies based on agile and simplified design [15]; in particular, this methodology proposes a cycle consisting of the following three phases:

P1 Collection of domain information with the help of domain experts, definition of usage scenarios and test cases, definition of a modelet (ontology piece) based on these principles and meeting the usage requirements, definition of test cases, and release of the modelet;

[9] https://www.measurence.com/.
[10] According to the MoSCoW prioritization technique - RFC2119.

Table 1. Intuitive query and response for each business case

Business Case	Intuitive Query & Response
Brand Performance Insights	QUERY: Given a table containing data about users queries for products, retrieve all events (from products price history) that describe the change in price in a selected temporal span (days) - API request: `/events/2021-01-01?query=event.measure.priceChange>10`
	RESPONSE: From the requirements, all data about events that present a change in price greater than 10% for a given product are retrieved and stored inside the `eventArray`. For each event, types and properties from the BEEO ontology are used, e.g., the type `beeo:Measure` is used to represent the change in price for the product on the given date.
Weather and Event-aware Business Intelligence for the Optimization of Campaigns and Human Resources	QUERY: Given a table containing data about user visits in online and physical stores in a selected temporal span (for instance 10 days), retrieve all recorded marketing events for a certain channel that took place in that period - API request: `/events/2021-01-01+9?query=event.channelCode=xcodeA32_3`
	RESPONSE: All event data regarding channel xcodeA32_3 and related to marketing events scheduled between 01-01-2021 and 01-10-2021 are retrieved and stored inside the `eventArray`. For each event, types and properties from the BEEO ontology are used, e.g., the type `beeo:channelCode` is used to represent the code associated with a certain channel.
Workforce & Campaign Management Optimization	QUERY: To build a dataset with user interaction data, retrieve all events about new product launch that occurred in the 30-day time interval - API request: `/events/2021-01-01+29?query=event.category.description=product%20launch`
	RESPONSE: All events that (i) belong to a category (`beeo:Category`) with a description matching the query "product launch" and (i) occurred between 01-01-2021 and 01-30-2021 (29 days) are returned to the caller. Each event returns the "attendingAudience" information of type *beeo:Measure* that will be used to study the resulting interactions with users.
Event and Weather-aware Foot Traffic Predictions and Analytics	QUERY: Given a table with information about user visiting a showroom in Milan (Italy) with postal code 20131, retrieve data about the interested audience for events occurred over a 6-day time span - API request: `/events/2021-05-05+5?query=event.location.addressCountry=ITA&event.location.postalCode=20131`
	RESPONSE: All the events occurred in Milan in the considered time period are returned by the API to the caller. Among the various pieces of information, the one that is used to enrich the caller's data set is "interestedAudience" of type *beeo:Measure*.

P2 Integration of the test cases with the current ontology;

P3 Refactoring of the current ontology. The methodology also includes in its substeps several recommendations: usage of a glossary (terms to be considered) for the definition of the test cases, reuse of ontology design patterns and existing ontologies, keep the modelets and the ontology simple and close to the requirements specified in the test set, best practices for entity names.

The work to design BEEO has been, therefore, organized in the following phases (we include references to the above-mentioned methodology).

1. (P1) State-of-the-art: a comprehensive review of the literature and available tools. This preliminary study allowed us to identify the recurrent patterns for modeling events, and rank ontologies by popularity and completeness. To this we added the analysis of event descriptions in Schema.org. The outcome of this phase is that Schema.org is the most popular event ontology and the one that best covers the collected requirements. This ontology provides, in fact, several patterns for modeling events and related information (a guideline recommended in the adopted methodology).
2. (P1) Sample event data collections: the current definition of BEEO started by collecting event data samples from partners to identify the main concepts and data of interest for each partner.
3. (P1) Schema alignment: sample tables have been compared to identify common concepts (properties for the description of events), and preliminary data type definition.

4. (P1) Use and test cases definition: the usage of ontology-compliant event descriptions, with consequent test cases, is well defined in EW-Shopp: it consists in the enrichment of corporate data with custom event data relevant for their analysis.
5. (P1) Definition of guidelines for the ontology definition: based on the state-of-the-art review and the analysis of samples of partners' event data, we have derived a set of guidelines that have inspired the definition of the ontology.
6. Ontology definition: Schema.org has been adopted as starting ontology to define mappings where possible and add new concepts to comply the EW-Shopp needs. The main advantage is to keep compliance with existing tools and systems that already adopt Schema.org as reference ontology. This definition phase has followed the following sub-steps:
   - 6.1 (P2) definition of the subset of Schema.org of interest based on the vocabulary used in the sample schemas;
   - 6.2 (P2-P3) for each event data source: mapping of each data schemas to Schema.org and extension of the ontology with the source properties not covered by Schema.org;
   - 6.3 (P3) refactoring of the ontology and finalization of the first version.

Based on the initial use cases that support event-based analytics workflows in the industry, and on the previous steps of the adopted methodology, we have drawn the following guidelines to drive the design of the ontology:

- Harmonization reusing shared ontologies. To make the ontology valuable and extensible beyond the specific data preparation and analytics workflows supported in the project, we use the terminology of existing ontologies to harmonize the terminology used to describe events.
- Limited nesting of event descriptions. After the semantic enrichment step, event data appear in the columns of a table that contains the enriched data; as a consequence, when used in the analytical modeling steps, the event descriptions are flattened into a table; the event ontology should, therefore, natively support the enrichment step.
- Intuitive rendering of properties as table attributes in event descriptions. Because of (2), the column headers should intuitively describe the content of the columns; while searching for harmonizing the terminology used to describe the event (i.e., reducing the number of different terms used to describe similar properties of the events) the terminology must make the data still understandable by users who will work with them in the analytical modeling steps of the workflow. As a consequence, some of the terminology used by partners to describe their events should be preserved in the ontology.
- Polymorphic property usage and heuristic specifications of domains and ranges. We found that the reasons that motivated the polymorphic property usage and the heuristic specification of domains and range (i.e., as a recommendation more than as a normative specification) also applies to the context where this event ontology is used. For example, the event ontology is primarily used to specify the meaning of properties used in data exchanged

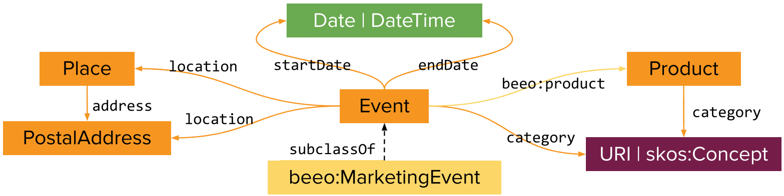

Fig. 1. Main types used in BEEO and their mutual relations.

using the JSON-LD format. Instead, when event data appear in an enriched dataset, events are either modeled in JSON or in a tabular format; in the first case, JSON-LD is fully JSON compliant; in the second case, ontology types do not appear while property names are column headers.

The resulting ontology is property-driven, which means that the primary goal is to harmonize the properties already used to describe events. When data are collected as JSON data, JSON-LD can be used to reuse the ontology properties; when data are collected as or factored into a table, properties can provide a header for each column. For this reason, we mostly specified the properties of the ontology, identifying a minimal number of types that are relevant as they are used as types of subjects or objects (values) for these properties. We found this agile methodology for data-driven vocabulary creation quite useful and applied it *as-is* in other projects as well (e.g., to model the vocabulary of a fantasy football knowledge graph).

3 Ontology Description

The `BEEO` ontology is built upon Schema.org, rather than upon other existing ontologies for two reasons: (i) the uptake of Schema.org in domains and communities related to eCommerce, digital marketing, etc., and (ii) was found convincing for our partners as it covered all the use cases.

Following the adoption of Schema.org as the reference ontology, we identified, among the available properties, those that can be mapped onto the ones in use by data providers. For those that do not represent the concepts of interest, we introduced new properties as specialization of existing Schema.org properties, so to keep the highest possible compliance. Table 2 reports the properties taken from Schema.org (with `schema` prefix and highlighted with orange background), and the ones introduced by EW-Shopp (with `beeo` prefix and highlighted with yellow background). The "notes" column reports the relations between the property specified in a row and Schema.org properties: the phrase "derived from a type" means that the type is specified among the domains of the property, and *subproperty of* is used to specify the superproperty connected to the new property.

The ontology is specified in RDF. Figure 1 reports the main types used in the ontology and their mutual relations. The color convention is consistent with the one introduced in Table 2: dark orange is for types and properties specified in Schema.org, yellow for types and properties introduced in BEEO, green for data types, and purple for a generic URI type (considered equivalent to Thing) and the property from SKOS ontology. We omit the prefixes of all types and properties that are either reused from Schema.org or based on xsd:types (i.e., Time and DateTime). Properties that are not shown in the figure are those that either have data types as ranges (e.g., integers, floats, etc.), with the only exception of time-related information that is central for event representation, or have literal values or describe more detailed information (e.g., postal addresses).

The ontology has the following characteristics:

– It is based on an extension of Schema.org ontology.
– As Schema.org, it uses polymorphic properties and heuristic domain/range specifications (with includesDomain and includesRange); these features make it difficult to depict multiple domain and range specifications in Fig. 1 (we represent multiple range specifications as single nodes with more labels separated by the | symbol and only report main types used as ranges).
– The main types considered in the ontology are derived from Schema.org and are listed among the most frequently used types.[11] These types are:
  • schema:Event, which is the type associated with all events;
  • schema:Product, which is the type associated with products;
  • schema:Place, which is the type associated with locations;
– Additional types used in the ontology are:
  • beeo:MarketingEvent, which is the only new type introduced in the ontology, and is defined as subclass of schema:Event;
  • skos:Concept, which is defined as the possible type for a property schema:category, which is introduced to associate a category with an event; the type schema:CategoryCode is pending in the Schema.org definition and has not been used in domains similar to the ones addressed in EW-Shopp so far; for this reason we reused a type defined in SKOS,[12] a W3C-recommended language to define simple categorization systems;
  • schema:PostalAddress, which is used because it is the recommended value for the schema:address property that is attributed to locations (instances of schema:Place); in practice, a postal address is a placeholder used to aggregate more specific address information specified using a number of properties; leveraging the non-normative specification of domains and ranges in Schema.org, we also consider descriptions where these properties (e.g., schema:postalCode) are directly referred to places without using an instance of postal address as intermediary.
  • Schema.org does not provide properties to describe measures of every event aspect, e.g., it provides properties to capture the number of registered attendees, but not the number of actual attendees that participated in an event; we introduced dedicated properties to describe these measures; in this case, we preferred to keep the terminology as close as possible to the one used by the partners; however, we linked these properties to Schema.org by specifying their superproperties in Schema.org.

[11] https://schema.org/docs/gs.html#schemaorg_types.
[12] https://www.w3.org/TR/2008/WD-skos-reference-20080829/skos.html.

The recommended version of BEEO is based on the Schema.org modeling approach, with heuristic specifications of domains and ranges. However, we also provide an OWL version of the ontology, especially to support visualization and editing with ontology editors. In Schema.org, properties are used in a polymorphic fashion: a property can be used either as ObjectProperty or as DataTypeProperty. In the OWL version, properties are classified as ObjectProperty or DataTypeProperty based on their preferred usage. Domain and range restrictions are introduced only for properties where only one class/datatype was specified as value of the domainIncludes and rangeIncludes properties. Users willing to extend the ontology can look at the recommended types specified in Schema.org in the annotation properties.

All data, and, in particular, textual data are represented using Unicode UTF-8 character encoding to support interoperability across languages at the alphabet level. In total, the Business Event Exchange Ontology has 53 classes (52 from Schema.org and one defined within EW-Shopp (beeo:MarketingEvent)) and 40 properties (27 from Schema.org and 13 defined within EW-Shopp).

4 BEEO API and Use Cases

From the analysis of the use cases described in Sect. 2.1 emerged the need to provide access to the event registries with a shared API that can provide machine-readable descriptions of event properties. The use of the popular format JSON, along with JSON-LD to support identifiers exchanging, was identified as the preferred format to simplify the use of the ontology and event descriptions.

In the remainder of this section, we first describe the API to access a generic event registry designed according to BEEO, and then discuss an example of use from the EW-Shopp partners.

4.1 BEEO API

Implementation of BEEO is realized by offering an Event API to the user. The API will enable the user to manipulate the event data, which are stored in an Event DB, and fetch them according to the proposed model. Within the suggested API implement action, the number of methods is optimized to the ones, (i) reflecting the actual needs of a typical user, and (ii) keeping the manipulations as simple as possible. A typical scenario in event-based data analytics is to fetch all events under certain constraints (e.g., related to one product) in a given time window.

The BEEO API specification is publicly available.[13] This API is build following best practices[14] and is based on a reduced set of calls (`GET event/{date[+N]}`, `POST event`, and `POST events`) and a simple format in JSON-LD. REST APIs are one of the most common web services available as they allow various clients

[13] https://app.swaggerhub.com/apis/EW-Shopp/Business-Event-Exchange-Ontology-API/2.2.0.

[14] https://json-ld.org/spec/latest/json-ld-api-best-practices/.

Table 2. Business event exchange ontology properties

Name	Range	Description	Notes
BEEO definition (properties that describe instances of schema:Event)			
schema:identifier	URI|Text	An identifier of an item	schema:Thing
schema:name	Text	The name of the item.	schema:Thing
schema:description	Text	A description of the item.	schema:Thing
beeo:source	Text	A description of event source	
beeo:channelCode	Text	A code associated with a channel in a marketing event	beeo:MarketingEvent
beeo:channelDescription	Text	A description associated with a channel in a marketing event	
schema:startDate	Date|DateTime	The start date (and time) of the item	schema:Event
schema:endDate	Date|DateTime	The end date (and time) of the item	schema:Event
schema:category	URI	A category for an item	schema:Thing (subproperty of schema:about; rec. range is skos:Concept)
beeo:quantity	xsd:int	A number identifying a generic quantity	Subproperty of beeo:simpleMeasure
beeo:quantyUnitId	URI|Text	The specification of the unit in which a quantity is measured	Subproperty of schema:identifier
beeo:interestedAudience	xsd:int	Number of interested/registered people	Subproperty of beeo:simpleMeasure
beeo:attendingAudience	xsd:int	Number of event attendees	Subproperty of beeo:simpleMeasure
beeo:priceChanged	Boolean	Specify if the product price has been changed	Subproperty of beeo:booleanMeasure
schema:discount	Text|Boolean	Any discount applied (to an Order)	schema:Order
beeo:priceChange	xsd:float	Price change in %	Subproperty of beeo:simpleMeasure
schema:price	xsd:float	The offer price of a product, or of a price component when attached to PriceSpecification and its subtypes.	schema:Offer
beeo:product	URI|Product	The product the event refers to (only for events about products)	Subproperty of schema:about
schema:location	Text|Place|PostalAddress	The location of an event, or where an action takes place.	schema:Event
beeo:simpleMeasure	xsd:float|xsd:int	A generic measure.	Subproperty of schema:value
beeo:booleanMeasure	xsd:float|xsd:int	A measure that assigns a boolean value	Subproperty of schema:value
Classification definition			
schema:description	Text	A description of the item.	schema:Thing
Product definition (properties that describe instances of schema:Product)			
schema:gtin13	Text	The GTIN-13 code of the product, or the product subject of an offer.	schema:Product
schema:description	Text	A description of the item.	schema:Thing
schema:seller	URI	An entity which offers/sells/ leases/lends/loans the services or goods. A seller may be a provider.	schema:BuyAction or schema:Offer or schema:Order
schema:sku	Text	A merchant-specific identifier for a product or service, or the product to which the offer refers.	schema:Product or schema:Offer
beeo:catalogId	Text	Specify the identifier	Subproperty of schema:identifier
schema:description	Text	A description of the item.	schema:Thing
schema:category	URI	Specified as subproperty of schema:about; range is skos:Concept	schema:Product or schema:Thing
Location definition (properties that describe instances of schema:Place and PostalAddress)			
schema:name	Text	The name of the item.	schema:Thing
schema:description	Text	A description of the item.	schema:Thing
schema:addressLocality	Text	The name of the locality	schema:PostalAddress
schema:addressCountry	Country|Text	The country (also formatted as ISO 3166-1 alpha-2)	schema:PostalAddress
schema:latitude	Number|Text	The latitude of a location.	schema:GeoCoordinates
schema:longitude	Number|Text	The longitude of a location.	schema:GeoCoordinates
schema:addressRegion	Text	The region. E.g., CA.	schema:PostalAddress
schema:streetAddress	Text	The street address.	schema:PostalAddress
schema:postalCode	Text	The postal code. E.g., 94043	schema:PostalAddress
schema:address	Text|PostalAddress	The address, possibly specified as a structured PostalAddress specification.	schema:Place or schema:Person or schema:Organization or schema:GeoShape or schema:GeoCoordinates

(browsers, apps, etc.,) to communicate with a server. The GET event purpose is to get a list of events starting from a certain date and spanning over N days. N is given as an optional parameter with the default value equal to “0”. The Path parameter date[+N] represents a date in ISO 8601 format to which is optionally concatenated (by means of the + operator) the information on the number of days ($0 < N \leq 99$) making up the time interval within which the events to be obtained have begun. For example, the following dates are valid: 2016-04-06T10:10:09Z+5, 2016-04-06+9, 2016-04-06, 2016-04-06+10, 2016-04-06+10:01. Successful responses return the retrieved data (Listing 1.1), while specific error responses are implemented to handle standard HTTP error response codes.

Listing 1.1. Successful pull of event data.

```
{
    "@context": {
        "@version": 1.1,
        "@base": "http://inside.disco.unimib.it/BEEO/",
        "schema": "http://schema.org/",
        "beeo": "http://inside.disco.unimib.it/BEEO/ontology/",
        "beed": "http://inside.disco.unimib.it/BEEO/data/rdf",
        "xsd": "http://www.w3.org/2001/XMLSchema#",
        "lang": "@language",
        "text": "@value",
        "identifier": "@id",
        "eventArray": {
            "@id": "beeo:eventArray",
            "@type": "@id",
            "@container": "@set",
            "@context": { "@base": "/rdf/event/" }
        },
        "name": {
            "@id": "schema:name",
            "@language": "en"
        },
                    [...] MISSING CONTENT [...]
        "location": {
            "@type": "beeo:PostalAddress",
            "addressLocality": "Mountain␣View",
            "addressCountry": "USA",
            "addressRegion": "CA",
            "streetAddress": "1600␣Amphitheatre␣Pkwy",
            "postalCode": 94043
        }
    }
}
```

A large number of schemas can optionally be used; among them: IntegerMeasure, AudienceMeasure, PriceMeasure, PostalAddress, LangString, Place, Product, EventsArray, Seller, Category, Context, Event.

4.2 Use Case

We demonstrate the usage of the ontology in one of the business cases described in Sect. 2, where the exchange of third-party event data is more crucial and now operational, making the use of semantics for event data more relevant.

Workforce and Campaign Management Optimization. Browsetel and CDE are developing and selling the COCOS Customer Engagement Platform (COCOS CEP) Omni-channel communication solution to SME clients and large enterprises. Prediction of optimal resources and correct timing for placing campaign calls has always been a challenge within the Contact Center. The optimal number of agents depends on the predicted traffic of inbound/outbound calls and on the success rate of the outbound calls. Plenty of different parameters influence the overall success of marketing campaigns and resource optimization. When the prediction system considers not only internal criteria based on contact center call history, but also external factors, the prediction models can be significantly improved. The Contact Center has been always working in conjunction with clients' departments where their customers are supported by automated chatbots and/or human agents. Activities on the clients' side affect the overall behavior at the Contact Center, therefore, when an event occurs (e.g., the launch of a new product on the market, or a change of a service offered by a client) the Contact Center should be notified in advance, and the workforce management tool should be able to support the managers to define the needed number of active agents. The use of BEEO and of the event API has the objective of simplifying the notification process by the clients, which can exploit event API to provide the Contact Center with their events, and the workforce management tool to consume events with to feed the training process of predictive models.

To optimize its Workforce and Campaign Management process the company used the following inputs: (i) Contact center historical data (recorded interactions between contact center customers and agents), (ii) Weather Data (collected from the ECMWF[15] available Weather Resources), and (iii) events (referred to as Custom Events) collected from the COCOS CEP Business Clients. To enable the management of the Custom Events, the company used the Event API and upgraded its COCOS Campaign Management tool. The Event API has been added to the system in order to enable the COCOS CEP Business Client to directly insert its own events into the system. Thus, enabling the prediction of non-standard events influences the system load. The event ontology supports sharp definitions of events by setting, among others, the source of the event (beeo:source) with a category (schema:category) and associated quantity (beeo:quantity) to size the event (e.g., the number of customers that will be affected by the new version of a service). Such semantic event descriptions are used to enrich the historical Contact Center data, which records the (anonymized) interactions between customers and agents, as well as results of the Contact Center campaigns.

For example, Fig. 2 reports predictions (lines) and actual data (blocks) about the success rate of outbound calls and shows that predictions that also consider events are more accurate than predictions based on other factors (e.g., only

[15] https://www.ecmwf.int/en/forecasts.

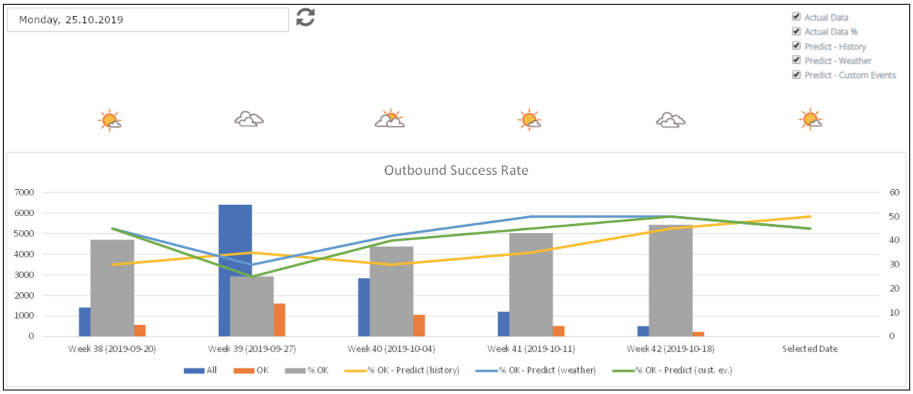

Fig. 2. Use case: predictions vs. actual data.

weather). Events have shown to be key factors to improve the reliability of predictions in this[16] and other similar tasks.[17]

5 Related Work

Event ontology is a shared, formal and explicit specification of an event class system model that exists in real world objectively [13]. Even though there is a number of ontologies suitable for representing events, they were created to serve different purposes. Event ontologies are used in different domains [16], including museums and libraries to describe historical events (e.g. wars, or births) as well as events in the histories of the objects being described (e.g. changes of ownership, or restoration) [5], ABC for modelling archive or digital resources [12], scholarly events for scientific communication channels [6], event logs from databases [2], for biological processes [11], journalism [10], etc.

An overview and a comparison of existing event models is provided in [17]. The main aim is to review different choices that might be used to represent events and build an interlingua model to resolve interoperability problems. This is solved by providing a set of axioms that express mappings between existing event ontologies. The result of this work is LODE,[18] an OWL ontology for representing and Linking Open Descriptions of Events licensed under the Creative Commons License. The ontology defines classes and properties to describe historical events as linked data, mappings to other event-related vocabularies and ontologies (e.g., OWL-Time).

16 https://www.ew-shopp.eu/solution/cocos-cep-worforce-campaign-management-optimization/.
17 https://www.ew-shopp.eu/solutions/.
18 http://linkedevents.org/ontology/.

Event Ontology[19] is centered around events that occur at a certain place and time and that can involve the participation of a number of physical objects both animate and inanimate. It defines one main concept Event that may have a location, a time, factors (e.g., a musical instrument), active agents (e.g., an instrument performer) and products (e.g., the performed sound). Events are considered as a first class object or "token", acting like a hook for additional information pertaining to the event. Such concept might be linked to a particular place through the predicate *event:place* by linking the Event ontology to the Geonames ontology, and to a particular time through event:time, linking the Event ontology to the Timeline ontology. It is possible to represent also information about complex events in a structured way by breaking it into simpler subevents, where each of which can carry part of the information pertaining to the complex whole. Although simple, such an ontology has already been proven useful in a wide range of context, e.g., talks in a conference, descriptions of concerts, or chords being played in a Jazz piece (when used with the Timeline ontology), festivals, etc.

EventsML-G2[20] is a data model and format specified in XML-Schema for collecting and distributing event information. It is part of NewsML-G2, a data model and format to exchange text, images, video, audio news and event or sports data among news agencies. EventsML-G2 is defined as a standard for conveying event information in a news industry environment, but can be used also for publishing all facts about a specific event by a news provider, storing facts about knowledgeable events in archives, adding information regarding the coverage of an event by a news provider. EventsML-G2 is defined by IPTC, a body for developing and publishing Industry Standards for the exchange of news data of all common media types.

LODE and Event Ontology have been adopted by large communities of users and are quite abstract. While the NewsML-G2 standard is adopted by different news agencies, it is not clear if the EventsML-G2 vocabulary is frequently used. In addition this standard is known in the news domain but not adopted in domains addressed by this paper.

GoodRelations[21] is a powerful vocabulary that is used for publishing business-related goods and services. It finds applications in different use cases such as information about products and services exchange, pricing, payment options, other terms and conditions, store locations and their opening hours, and many other aspects of e-commerce, between networks of computer systems [7].

In [8] authors propose to extend an event ontology by firstly, classifying the types of events (e.g., natural events and artificial events). The approach attempts to construct event classifications based on two ontological views: component structures (knowledge representation of events) and semantic functions of events (which imply the logical and ontological semantics of events for reasoning). Secondly, event relations (e.g., causal relations and next-event relations)

[19] http://motools.sourceforge.net/event/event.122.html.
[20] https://iptc.org/standards/eventsml-g2.
[21] http://www.heppnetz.de/ontologies/goodrelations/v1.html.

captures the differences between instances and classes of events. The semantic functions of events were analyzed in expressive logical formulas that would allow to infer logical conclusions from event occurrences.

The Rich Event Ontology goal is to provide a unified representation of events [1]. The main reference ontology encompasses 161 classes and 553 axioms. Including the lexical resource ontologies and the linking models into counts brings the totals to 3,065 classes and 60,531 axioms, as well as 16,005 individuals representing the vocabulary (unique lemmas) of events. Authors provide different use cases, how users could benefit from such ontology. Despite, the authors claim that the ontology will be available, but at the time of writing it is not available and freely used.

6 Conclusions and Future Work

In this paper, we have presented BEEO, an extension of Schema.org that enables data integration and analytics with event data. The ontology is available under a public license and can be freely used, reused, and further extended. Based on BEEO, an API has been developed to support event-data exchange. The ontology and the respective API have been created and tested in real business environments.

Requirements have been collected from the real business cases developed in the EW-Shopp EU project to provide shared terminology and common tools to support tailored services in various contexts with different goals. The development of BEEO was required to overcome the limits of existing vocabularies like Schema.org, and proved to be effective to model key aspects in the marketing domain. Each partner has implemented a private version of the API to upload and retrieve events according to BEEO data model. In this way, partners were able to use the open-source tools provided by the projects to develop, test, and successfully deploy their services. Such a development model can be replicated by new users, either business companies or research institutions, to augment their data with events.

In BEEO we do not yet have classes or properties that cover specific concepts thus a future direction is to increase the expressivity of the ontology to support such specific-level semantics.

Acknowledgements. This research has been supported in part by EU H2020 projects EW-Shopp - Grant n. 732590, and EuBusinessGraph - Grant n. 732003.

References

1. Brown, S.W., Bonial, C., Obrst, L., Palmer, M.: The rich event ontology. In: Proceedings of the Events and Stories in the News Workshop, pp. 87–97 (2017)
2. Calvanese, D., Montali, M., Syamsiyah, A., van der Aalst, W.M.P.: Ontology-driven extraction of event logs from relational databases. In: Reichert, M., Reijers, H.A. (eds.) BPM 2015. LNBIP, vol. 256, pp. 140–153. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42887-1_12

3. Cutrona, V., Ciavotta, M., De Paoli, F., Palmonari, M.: ASIA: a tool for assisted semantic interpretation and annotation of tabular data. In: ISWC Satellites, pp. 209–212 (2019)
4. Cutrona, V., et al.: Semantically-enabled optimization of digital marketing campaigns. In: Ghidini, C., et al. (eds.) ISWC 2019. LNCS, vol. 11779, pp. 345–362. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30796-7_22
5. Doerr, M.: The cidoc conceptual reference module: an ontological approach to semantic interoperability of metadata. AI Mag. **24**(3), 75–75 (2003)
6. Fathalla, S., Vahdati, S., Auer, S., Lange, C.: The scientific events ontology of the openresearch. org curation platform. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp. 2311–2313 (2019)
7. Hepp, M.: GoodRelations: an ontology for describing products and services offers on the web. In: Gangemi, A., Euzenat, J. (eds.) EKAW 2008. LNCS (LNAI), vol. 5268, pp. 329–346. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87696-0_29
8. Kaneiwa, K., Iwazume, M., Fukuda, K.: An upper ontology for event classifications and relations. In: Orgun, M.A., Thornton, J. (eds.) AI 2007. LNCS (LNAI), vol. 4830, pp. 394–403. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76928-6_41
9. Kietzmann, J., Paschen, J., Treen, E.: Artificial intelligence in advertising: How marketers can leverage artificial intelligence along the consumer journey. J. Advert. Res. **58**(3), 263–267 (2018)
10. Kowalczuk, E., Lawrynowicz, A.: The reporting event ontology design pattern and its extension to report news events. Adv. Ontol. Des. Patterns **32**, 105–117 (2017)
11. Kushida, T., Takagi, T., Fukuda, K.I.: Event ontology: a pathway-centric ontology for biological processes. In: Biocomputing 2006, pp. 152–163. World Scientific (2006)
12. Lagoze, C., Hunter, J.: The ABC ontology and model. In: International Conference on Dublin Core and Metadata Applications, pp. 160–176 (2001)
13. Liu, W., Liu, Z., Fu, J., Hu, R., Zhong, Z.: Extending owl for modeling event-oriented ontology. In: 2010 International Conference on Complex, Intelligent and Software Intensive Systems, pp. 581–586. IEEE (2010)
14. Noy, N.F., McGuinness, D.L., et al.: Ontology development 101: a guide to creating your first ontology (2001)
15. Peroni, S.: A simplified agile methodology for ontology development. In: Dragoni, M., Poveda-Villalón, M., Jimenez-Ruiz, E. (eds.) OWLED/ORE -2016. LNCS, vol. 10161, pp. 55–69. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54627-8_5
16. Rodrigues, F.H., Abel, M.: What to consider about events: a survey on the ontology of occurrents. Appl. Ontol. **14**(4), 343–378 (2019)
17. Shaw, R., Troncy, R., Hardman, L.: LODE: linking open descriptions of events. In: Gómez-Pérez, A., Yu, Y., Ding, Y. (eds.) ASWC 2009. LNCS, vol. 5926, pp. 153–167. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10871-6_11