

eng.event_driven

Alexandr Kirilov (<https://github.com/alexandrkirilov>)

Event driven

1. Event driven philosophy

In accordance with meaning of the term "event", based on Collins Cobuild Dictionary:

An event is something that happens, especially when it is unusual or important. You can use events to describe all the things that are happening in a particular situation.

For a software, this means we have an ability to use this definition for the basis of developing any kind of event driven system. Let's look at this definition in more detail. The most valuable part of this definition is: "... use events to describe all the things that are happening ..."

When we are evolving or creating any kind of event system we are always answering 3 questions primarily:

- **What?** - what kind of situation happened, what the nature of the event described by any kind of properties that we defined like valuable. For example:
 - Buy 1 litre of milk costs 1\$ (what properties: kind = milk, amount = 1 liter and price = 1\$)
 - Heavy accident on road with 4 casualties (what properties are: seriousness = heavy, casualties = 4)
 - Patient John Smith arrived with headache (what properties: first_name = John, last_name = Smith, symptoms = headache, diagnosis = none)
 - Switch on light in the house (what properties: house_equipment = light, mode = on)
 - etc
- **Where?** - where "What?" happened, what type of location: geographical location (Longitude and Latitude), relative location (something such as: "in the office", "on the road" or any other). For example:
 - We found sandbank at point 21.072069N, -86.389749W. This means location of event = 21.072069N, -86.389749W
 - Buy milk at shop at the point 37°46'52.0"N 122°23'57.5"W, means location description is totally the same like for the sandbank example. It might not be so comfortable for human everyday usage what are you referring to.
 - Buy milk at shop on 399 4th St, San-Francisco CA 94107 the location is the post address that might be associated to geographic location.

- Patient arrived at the doctor's office in room 56, the location is: the geographical coordinates associated with the medical facility, post address, and combined to the structure of the facility.
- etc
- **When?** - the position on timeline, not only "time". For now the physical meaning of the term "time" is based on rotation of Earth. We are only counting rotation around the Sun and the Earth self-rotation. For everyday language, we have divided one rotation around the Sun for months and one around itself in rotation on hours, seconds etc. Briefly: when we are saying something like this: ".. at 13:00AM 1 April 2017" we imply " ... when the Earth completes 2017 full rounds of Sun and 110 degree and 190 degree of self rotation ..." Because all of time is always a positive number that might be formatted by human and periodically be rendered null by human again. This includes some kind of vector within positive values from 0 to infinity.

For the case of developing Data Base (DB), answering the following 3W questions might be composed in an opposite order: When? Where? What?

2. Event's architecture, the 3W (When, Where, What) schema

After a brief explanation of event driven philosophy, lets talk about the technical approach that might be suited for any kind of purpose of event system. Based on 3W, it's obvious to use it like 3 parts architecture for any kind of event:

- When: describe the time following conventional definitions, standards or any other regulations.
- Where: describe the location in terms of information usage requirements you have to choose type of data;
- What: most difficult part of event because of innumerable list of type of events;

This is description of the part related to the event system core developing. There is not any intention in this article to describe client side solution. This definition is low-level definition for developing core of any event system.

2.1. Event's architecture part: when

The time description is most simple part for this case. In most variations of event it's better to use UNIX timestamp ([UNIX timestamp in collinsdictionary.com](http://collinsdictionary.com) or in unixtimestamp.com or in wikipedia.org) because technically it's the integer that is fastest for making analysis by time related request and uses the smallest size for

storing than any kind of string within time formatted for any standard. Besides all of it – you'll always be able to convert integer to any format and this operation will be cheaper than transforming one formatted time to another.

Technically this is a positive integer.

2.2. Event's architecture part: where

The location description is dependent upon usage of data. We need to define the level of detail and precision that is fitted for system usage. Some possible cases:

- you are handling any system that is using GEO related data - use geographical coordinates (longitude, latitude), for example:
 - Storing data of oil drilling that is directly related to location of drilling
 - Storing data of forest fire
 - Sea accidents or collisions reports
- You are developing a system that is suited to serve any facility or group of facilities within or without inner structure - develop the tree-like structure definition and use the ID of tree's nodes like identifiers within parent-child relationship description. Possible mostly used solution. In case of adding GEO functionality there is always ability to add through additional reference GEO-position for every node. For example:
 - Any holding that is operating couple or network of supermarkets within departments
 - Any medical facility within rooms divided on groups by the type of treatment
- You are developing system that has dependency on data from outside location definition system, for example Post delivery system, where every street, house or any other has their own ID. Use the same schema of location definition in your system. For example:
 - pizza delivery company
 - transport company
- you are developing complex system that has an inclusion of different kinds of locations, then taking care of operational usage of this data, and then define system location identifier that might be used and will be suitable for any cases.

Technically it might be only two types: - ID from location reference table within parent-child relation (ID might be: integer, string, binary - depend on kind of DB engine you are using) - GEO position by longitude and latitude, this is only pair of numbers that might be stored like any complex data or separately like numbers/integer/float.

2.3. Event's architecture part: what

Previous 2 parts was only "general". All of this 2 parts inhere to any kind of any events. Lets concentrate on the most difficult part of any event system - What?

This part is describing the nature of the event and therefore might be contained with any kind of data and might have any kind of structure. It means somehow for unification, we have to store the structure of event description within data validation schema. This is from each type of event that is operated by the system.

This is the most difficult point for understanding. These challenges are all inherent to current DB engines. For now we don't have any DB that is event oriented. We just started to use a collection in DB engines that is the first approach to event driven engines. Because of that we have to emulate event driven engines based on what we have right now.

Look below for one of possible approach that been using in couple of developed system before.

There is a main table that named "Timeline". In this table is stored data "When" and "Where" supplied by Event Unique ID (EUID) and Event Kind (EK). By EK we are selecting Events Kind Table (EKT) and Event Structure (ES), by EUID we are selecting event from the EKT that is interpreted based on ES.

After separating onto these tables we are able to independently create and perform a search procedure.

All of this means that we have to use a couple of tables instead of one and the structure of every table is unique. There is not any DB suited now for this use case "in-box". All of this has to be developed.

Follow author updates on [Linkedin](#)

Follow AR|BO|RE|US updates on [Twitter](#) and [Linkedin](#)