Why event processing needs a Rules Engine

aviata
www.AviataInc.com

Industry:

Products and Services:

Challenge:

Why does event processing need a rules engine? Before we tackle that directly, let's take a look at all of the elements at play and examine them one by one. Event processing and rules engines are a natural fit for one another. Let's first talk about Event Processing.

To learn what Complex Event Processing is - read our article titled What is Complex Event Processing?

What is event processing?

In the most general sense, event processing involves the reading of events, followed by the evaluation of data, and the taking of some kind of action in response. The following table outlines some of the more general and simple event processing use cases:

Category	Evaluation	Action
routing data	read routing criteria data	route event to proper recipient
data validation	compare data to data schema	reject if invalid
data aggregation	identify if data matches current time series	aggregate with time series

Event processing can also be incredibly complicated and complex with multistep data processing and multi-action/result outcomes.

Cogswell.io is Aviata's Complex Event Processing solution.

What is a rules engine?

Generically, a rules engine consists of defined rules of which questions are asked and a result is produced in response to the question. Typically the outcome is a boolean result (yes | no). This result can either be sent back to the sender or can be used to drive other results.

While there are several kinds of rules engines, we are going to focus on reactive rules engines. Reactive rule engines detect and react to incoming events and process event patterns. These rule engines use an event driven architecture approach.

Rules engines usually provide a complete management solution. They allow the creation, deletion, and update of rules. This is often done through a UI.

The primary goal of event processing is to evaluate some set of conditions and ultimately derive an actionable result.

What are the advantages of using a rules engine?

Service Abstraction

The rules engine itself can be a separate, distinct part of the overall solution.

Isolation of rules

The use of a rules engine allows logical separation of rules from code. Rules engines are frequently used to manage business rules. Business rules frequently change. Contrast the isolation of rules within a complete rules engine solution vs hard-coded rules which reside in code which may be spread out in different locations in the codebase. In the latter, the code containing hard-coded business rules may have to be frequently changed. Isolation of rules from the rest of your code base helps protect the integrity of your application and reduce the potential for various problems that may arise due to frequent code changes.

Performance / Scalability

Purpose built rule engines use tried and battle tested algorithms and various optimizations to achieve high performance. They are often built to be highly scalable to handle the work required of them. Performing tens to hundreds of thousands rule evaluations a second are within the realm of possibility. Building a rules engine from from scratch is a very time consuming, complex endeavour that takes extraordinary expertise to create one that is bug free and highly performant.

Knowledge Centralization

Business rules are often one of the most critical parts of a software solution. By using a rules engine with clean isolation of rules, that collection of rules becomes a wealth of information and a form of self-documentation. Rules engine's which have an interface to the rules in a clean, easy to read, easy to understand format - have an advantage here. Contrast this with hard-coded rules spread throughout a code base.

Time to market

Using an existing solution can greatly speed your time to market for your software productions.

When should you use a rules engine?

- The rules you are trying to implement are complex and would compromise the code and design of your solution.
- You have a large number of rules. Maintenance and future expansion is important to you.
- Subject Matter Experts within your organization regarding the rules you need for your solution are not technical. Often businesses are faced with the crossroads where the staff responsible for product conceptualization, and design are not developers. They understand the particular space they are trying to bring a solution to market in. A rule engine which has a human readable rule creation tool would provide them the ability to create and maintain the rules themselves (without coding knowledge).
- The logic of the rules themselves would change often.
- High throughput and performance are desirable characteristics.

When should you not use a rules engine?

- Small number of rules
- Rule complexity is generally low
- Rule changes will be infrequent after initial implementation
- The effort to integrate an existing rules engine would be take more time than it would take to roll a custom solution (consider future changes when making this decision!)

Conclusion

The basic concept of event processing is reading data, evaluating that data, deriving a result, and taking an action based on that result. Software solutions which involve event processing at their core are typically architected with an event driven architecture. Event driven design is a reactive model where processing occurs as a result of external events received as input to the system. Reactive rules engines are designed on the same principle, where rule evaluation is triggered in response to an event received as input.

The primary goal of event processing is to evaluate some set of conditions and ultimately derive an actionable result. Rules engines pair well in all aspects of this goal. They encapsulate the evaluation process, specific logic and conditions behind the evaluation, as well as producing a result. In addition to this primary service, they have rule management capabilities to further enhance maintainability and speed development. To top it off, they are designed to be a separate, isolated system with high performance characteristics.

If your current or planned software development effort involves processing event data and making decisions on that data, take a strong look at whether it makes sense to bring a reactive rules engine into your solution.