Filtering and transformation in practice

Filters and transformations are expressed in various ways in languages.

This example in Esper implements the BidRequest creation

```
insert into BidRequest(requestId, store, location, pickupTime, deliveryTime,
storeManual)
select d.requestId, d.store, d.location, d.pickupTime, d.deliveryTime, s.manual
from
DeliveryRequest d unidirectional,
GPSLocationW g
//,sql:DomainDB['select ranking from Driver where driver = ${g.driver} and ranking
> ${d.minimumRanking}']
,method:Domain.driverRankLookup(g.driver) r
,method:Domain.isStoreManualLookup(d.store) s
where Geo.distanceKM(g.location, d.location) < 10
and r.ranking >= d.minimumRanking;
```

In this example the derived event is created using the SQL command insert into.

The enrichment done using two methods; the first one performs lookup for drivers that satisfy the ranking requirements and the second one performs lookup for stores to find whether it prefers manual or automatic assignment. In addition there is location check for the drivers. As seen this is a combination of SQL with call to Java methods. For more details see the Esper FFD example.

Another example is in Apama, which shows transformation example using imperative language:

This is a transformation that enriches the Delivery Request with the current location of a driver. For more details see the Apama FFD example .

The next example is a filtering example in Streambase Event Flow UI:

This example is a filter which filter- in drivers that satisfy the store's minimal ranking. This is done as "Range Specification" for the attribute ranking.