**Reactive Data System**

**Using EDA**

***What is it all about?***

Data! One of the most exponentially increasing thing in today’s IT world. At the same time as a side effect, changes to existing data also occur (we will call it as a state change e.g., finance dept may have state change salary/fees pending to paid). So as to keep track of the changes and perform certain actions when some constraints are satisfied, we need to develop our enterprise application which will be highly reactive to state change we called it as a reactive data system which is build over EDA. We will demonstrate our EDA with *universal admission process* as our reactive data application.

***What is EDA?***

Event-driven Architectures is a *Programming without a Call Stack .*It is a science or we can look at it as design methodologies wherein all kinds of rules are defined with the help of two clauses,

**When** <Event Occurred>

**Then** <Action>

Now question arises

***What are events?***

Events represent a change in state, they are self contained (pure and complete representation, loosen coupling), uniquely identifiable, Observable Events may be data driven or time driven or both. We classified events into following types

1. Lifecycle events
   1. Notification related to the life of a state or process
   2. For example, admission group invitation to institute state pending/accepted/rejected, Authorizing payment / payment Authorizing.
2. Management events
   1. Time period elapsed, range or limit exceeded
   2. For example, Account activation expires, Registration full / closed, etc.
3. Business Events
   1. Map to real world events that occur in business
   2. For example, test score added, seat allocated, etc.

***What are Actions?***

Here action may be a user defined function call or SQL query which will be fired on some database.

***Why to bother about EDA for all this if all can be done as a regular call stack process?***

*Key features:*

1. Well suited for asynchronous, unpredictable environments.
2. Maintainable and extensible for complex, distributed applications
3. Reduce Coupling
4. Self governing, controlling and contained
5. Individual components can be developed and testable in isolation
6. Thorough logging

***What we did?***

We’ve developed a generic EDA framework which can support event and action registration through GUI. It has listener and caller unit which detect events and execute action respectively. It also provides API for run time event registration.

We are supporting multiple actions per event, for demonstrating working of our EDA we build Universal Admission Process APP using EDA framework.

Universal Admission App is a unique of its type, wherein any institute of any stream can automate their admission process individually or forming a group of institutes, enrolling their courses. A candidate can register for admission, enter his preferences, pays fee, get admission status. System will auto assign seats to eligible candidate.