# XSB code walkthrough,

The code is divided into 7 files, out of these 4 files are case-study specific (2 case studies). The remaining 3 files are the core files that are shared by both the case studies. I will describe these 3 files first then move to the case studies.

The core files,

1. creation\_script\_db.P
2. nested\_list\_approach.P
3. context\_filter.P

## creation\_script\_db.P

The predicate below is used to create a new environment, it creates both an environment and a component so that the environment can be nested in another environment, it also asserts the context variables of an environment.

create\_environment(Environment\_name, Context\_var\_list):-

assert(environment\_instance(Environment\_name)),

assert(component\_instance(Environment\_name)),

add\_context\_variable\_list\_environment(Environment\_name,Context\_var\_list).

The predicate below is used to create a component, the predicate also calls another predicate to save the context variables of a component.

create\_component(Component\_name, Context\_variable\_list):-

assert(component\_instance(Component\_name)),

add\_component\_context\_variable\_list(Component\_name,Context\_variable\_list).

The below predicate is used to save the definition of a new event schema,

create\_event\_schema(Event\_id,Event\_schema\_list):-

assert(event\_schema(Event\_id,Event\_schema\_list)).

The below predicate is used to declare an event advertisement for a component.

create\_advertisement(Component\_name,Environment\_name,Event\_id,Context\_filter\_expression):-

component\_instance(Component\_name),

environment\_instance(Environment\_name),

environment\_component(Environment\_name,Component\_name),

event\_schema(Event\_id,\_), assert(event\_advertisement\_entry(Component\_name,Environment\_name,Event\_id, Context\_filter\_expression)).

The below predicate is used to create advertisement for a environment,

create\_advertisement(Environment\_name,Event\_id,Context\_filter\_expression):-

environment\_instance(Environment\_name),

event\_schema(Event\_id,\_), assert(event\_advertisement\_entry(Environment\_name,Environment\_name,Event\_id,Context\_filter\_expression)).