Improvements in Data-driven Analysis

Naser Ezzati

Michel Dagenais

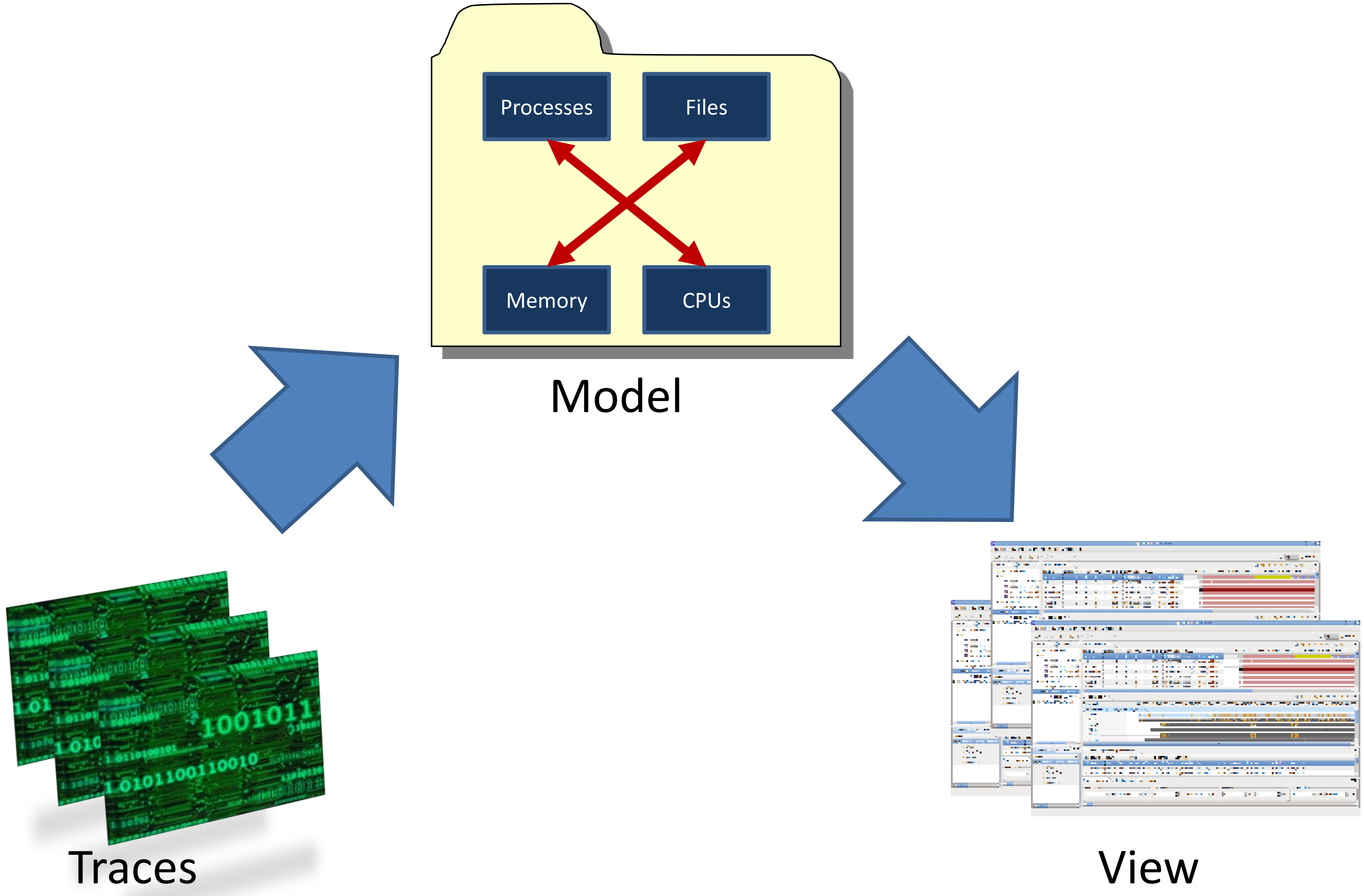
Progress Report Meeting

Dec 2014

POLYTECHNIQUE MONTRÉAL

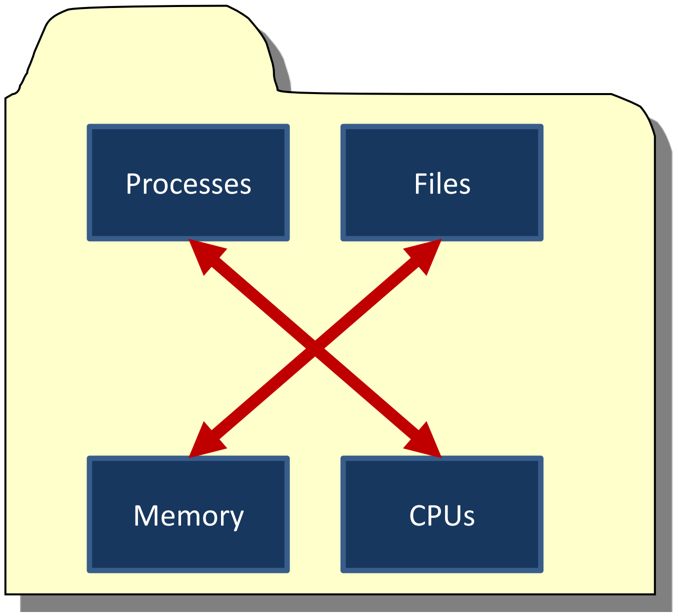
LE GÉNIE EN PREMIÈRE CLASSE

Trace-Model-View



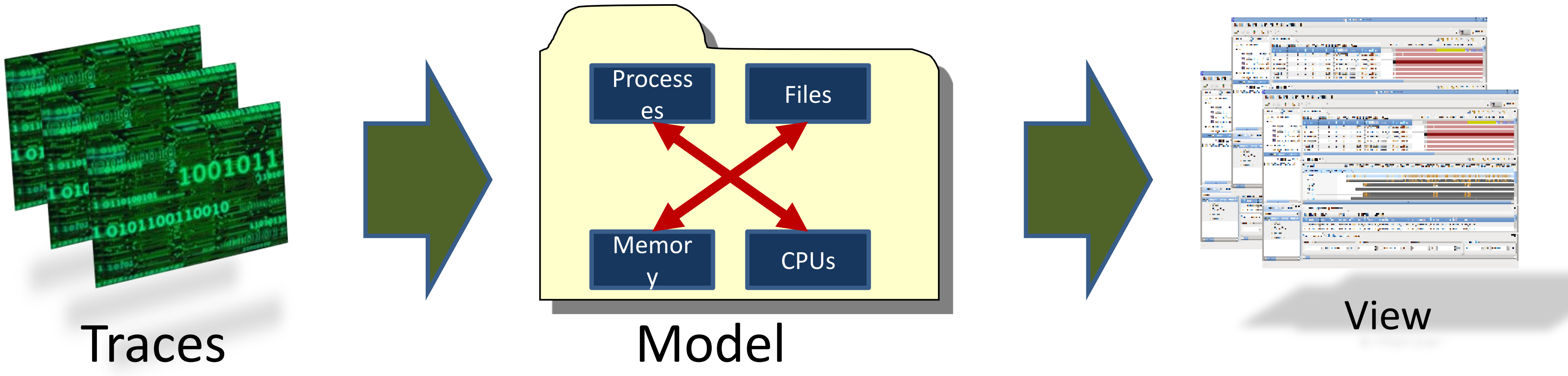
Model

- Model:
  - A specific organization of:
    - Aggregation of trace data
    - Snapshots
    - Summaries (& statistics)
      - Multiple levels
  - Database Systems
  - Custom Structures
    - State System
      - State History Tree

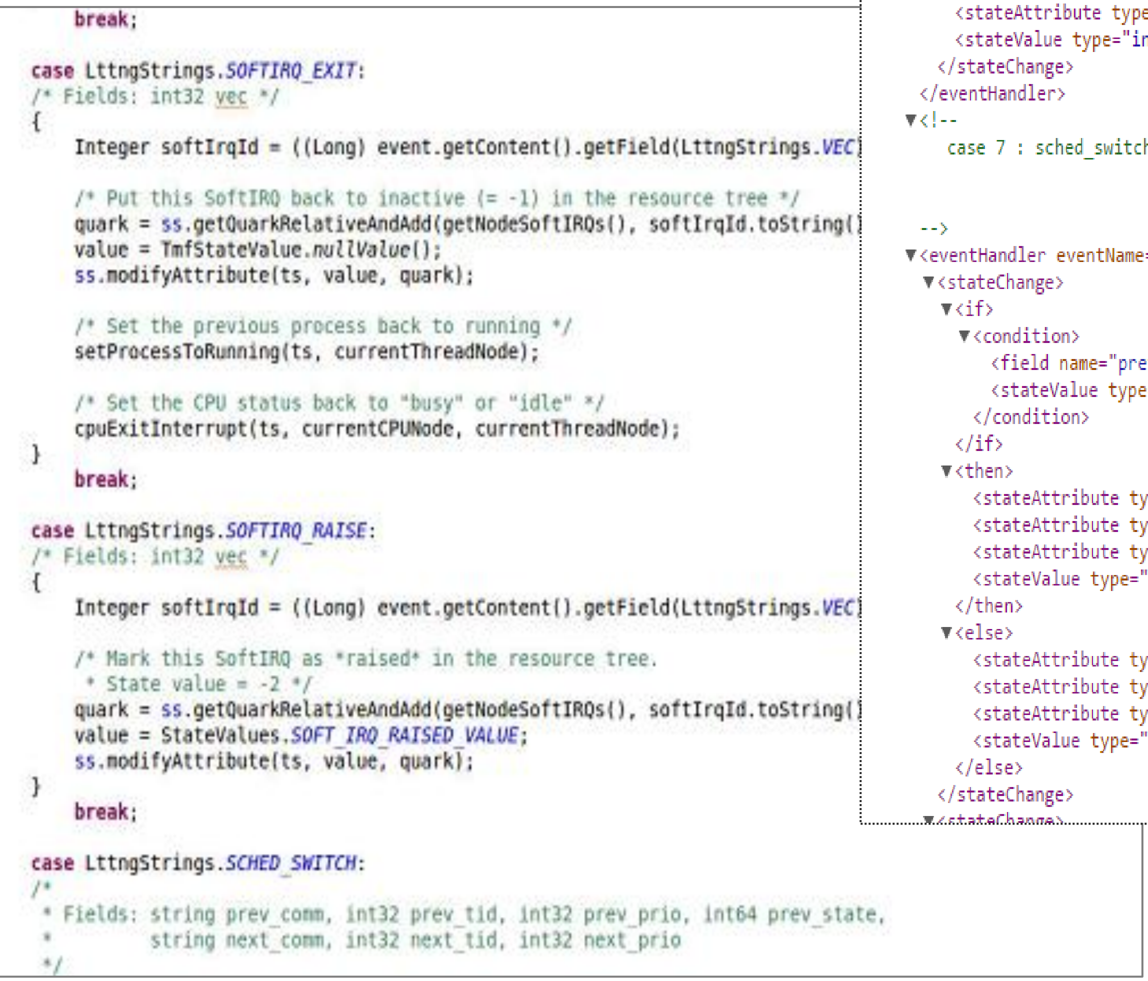


Data Driven Analysis

- Define custom models from (different formats/types of) trace data.
  - Support of various trace types
  - More flexibility
    - Than the default analysis offered by ***Trace Compass*** (TMF)
  - Easy to maintain (less code)
  - In both: trace -> model & model to view.

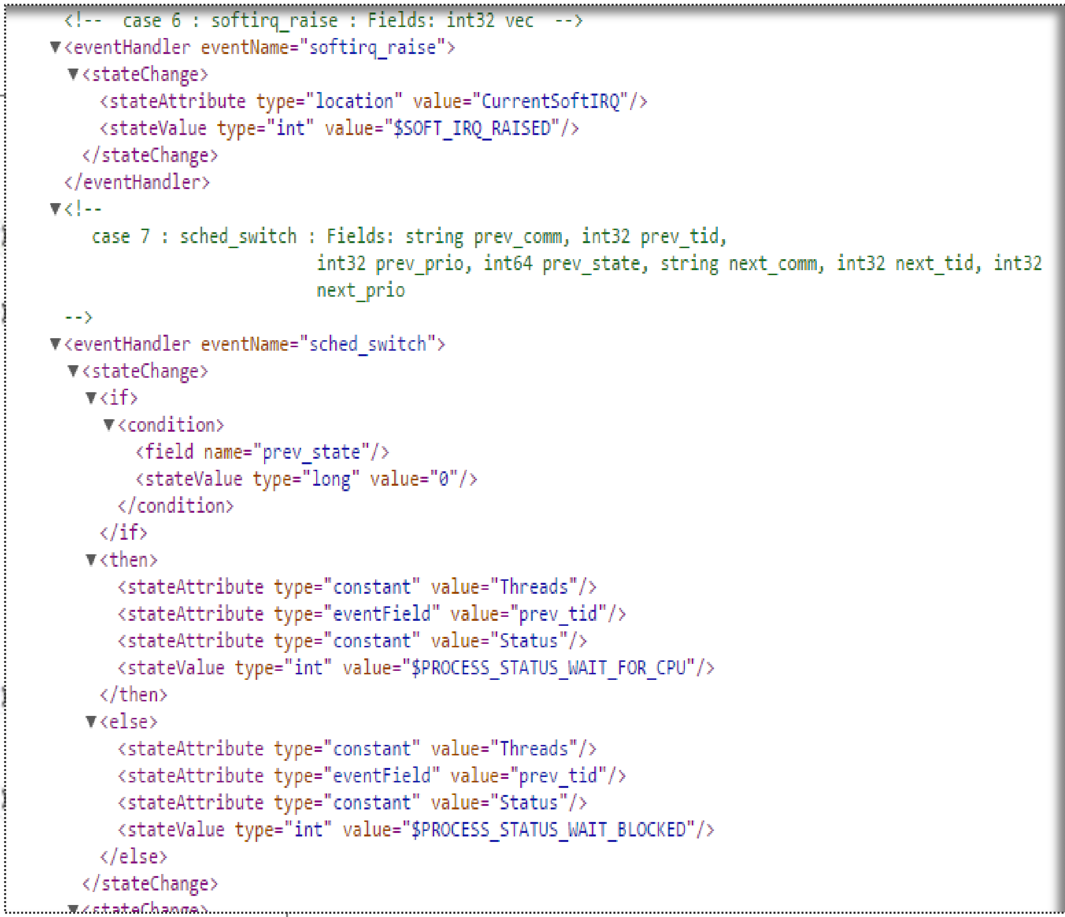


Data Driven Analysis



```java
    break;

case LttngStrings.SOFTIRQ_EXIT:
/* Fields: int32 vec */
{
    Integer softIrqId = ((Long) event.getContent().getField(LttngStrings.VEC)

    /* Put this SoftIRQ back to inactive (= -1) in the resource tree */
    quark = ss.getQuarkRelativeAndAdd(getNodeSoftIRQs(), softIrqId.toString()
    value = TmfStateValue.nullValue();
    ss.modifyAttribute(ts, value, quark);

    /* Set the previous process back to running */
    setProcessToRunning(ts, currentThreadNode);

    /* Set the CPU status back to "busy" or "idle" */
    cpuExitInterrupt(ts, currentCPUNode, currentThreadNode);
}
    break;

case LttngStrings.SOFTIRQ_RAISE:
/* Fields: int32 vec */
{
    Integer softIrqId = ((Long) event.getContent().getField(LttngStrings.VEC)

    /* Mark this SoftIRQ as *raised* in the resource tree.
     * State value = -2 */
    quark = ss.getQuarkRelativeAndAdd(getNodeSoftIRQs(), softIrqId.toString()
    value = StateValues.SOFT_IRQ_RAISED_VALUE;
    ss.modifyAttribute(ts, value, quark);
}
    break;

case LttngStrings.SCHED_SWITCH:
/*
 * Fields: string prev_comm, int32 prev_tid, int32 prev_prio, int64 prev_state,
 *         string next_comm, int32 next_tid, int32 next_prio
 */
```

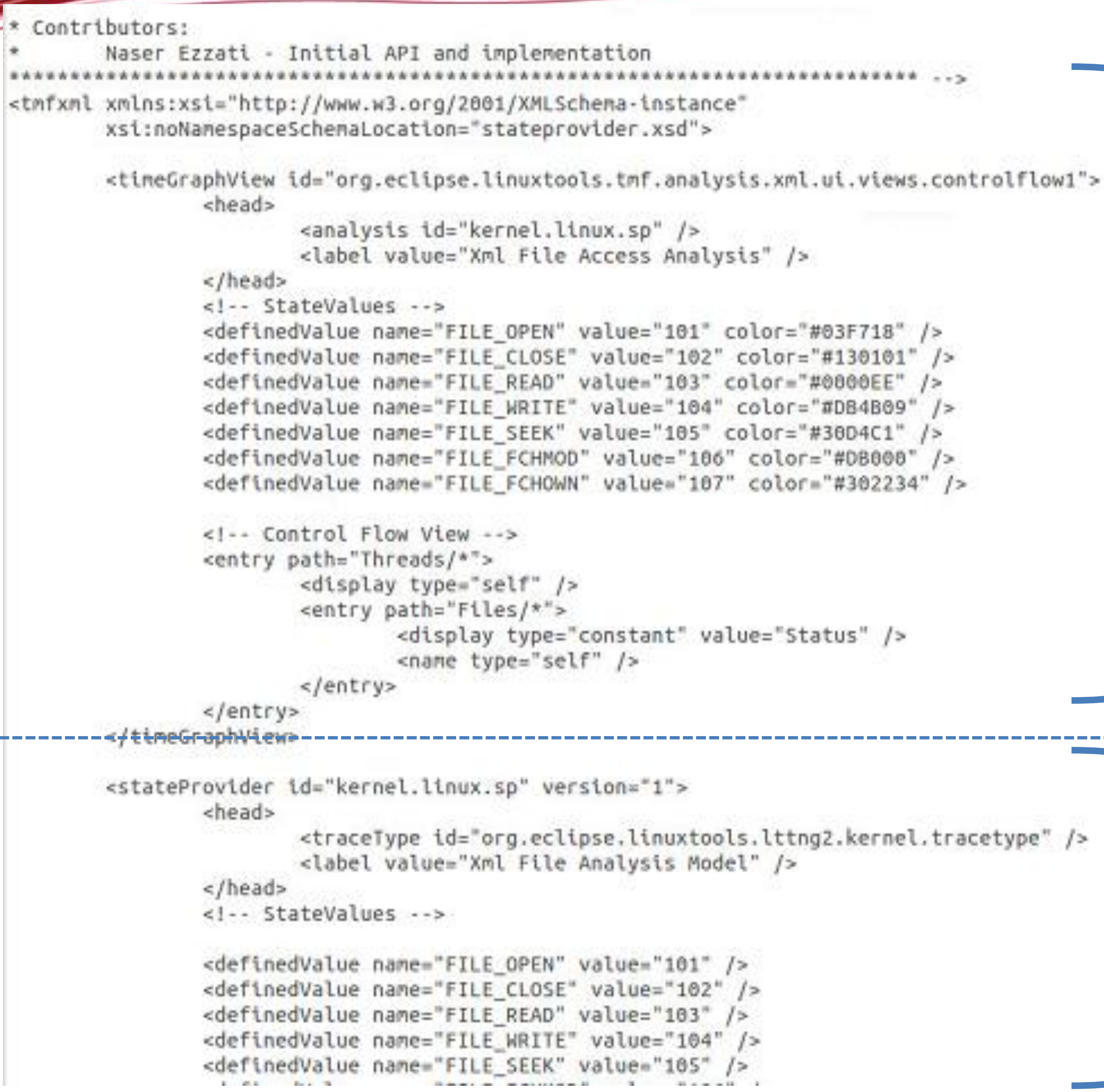
Java



```xml
  <!--  case 6 : softirq_raise : Fields: int32 vec  -->
▼<eventHandler eventName="softirq_raise">
  ▼<stateChange>
     <stateAttribute type="location" value="CurrentSoftIRQ"/>
     <stateValue type="int" value="$SOFT_IRQ_RAISED"/>
   </stateChange>
  </eventHandler>
▼<!--
    case 7 : sched_switch : Fields: string prev_comm, int32 prev_tid,
                            int32 prev_prio, int64 prev_state, string next_comm, int32 next_tid, int32
                            next_prio
  -->
▼<eventHandler eventName="sched_switch">
  ▼<stateChange>
    ▼<if>
      ▼<condition>
         <field name="prev_state"/>
         <stateValue type="long" value="0"/>
       </condition>
     </if>
    ▼<then>
       <stateAttribute type="constant" value="Threads"/>
       <stateAttribute type="eventField" value="prev_tid"/>
       <stateAttribute type="constant" value="Status"/>
       <stateValue type="int" value="$PROCESS_STATUS_WAIT_FOR_CPU"/>
     </then>
    ▼<else>
       <stateAttribute type="constant" value="Threads"/>
       <stateAttribute type="eventField" value="prev_tid"/>
       <stateAttribute type="constant" value="Status"/>
       <stateValue type="int" value="$PROCESS_STATUS_WAIT_BLOCKED"/>
     </else>
   </stateChange>
  ▼<stateChange>
```

XML

XML

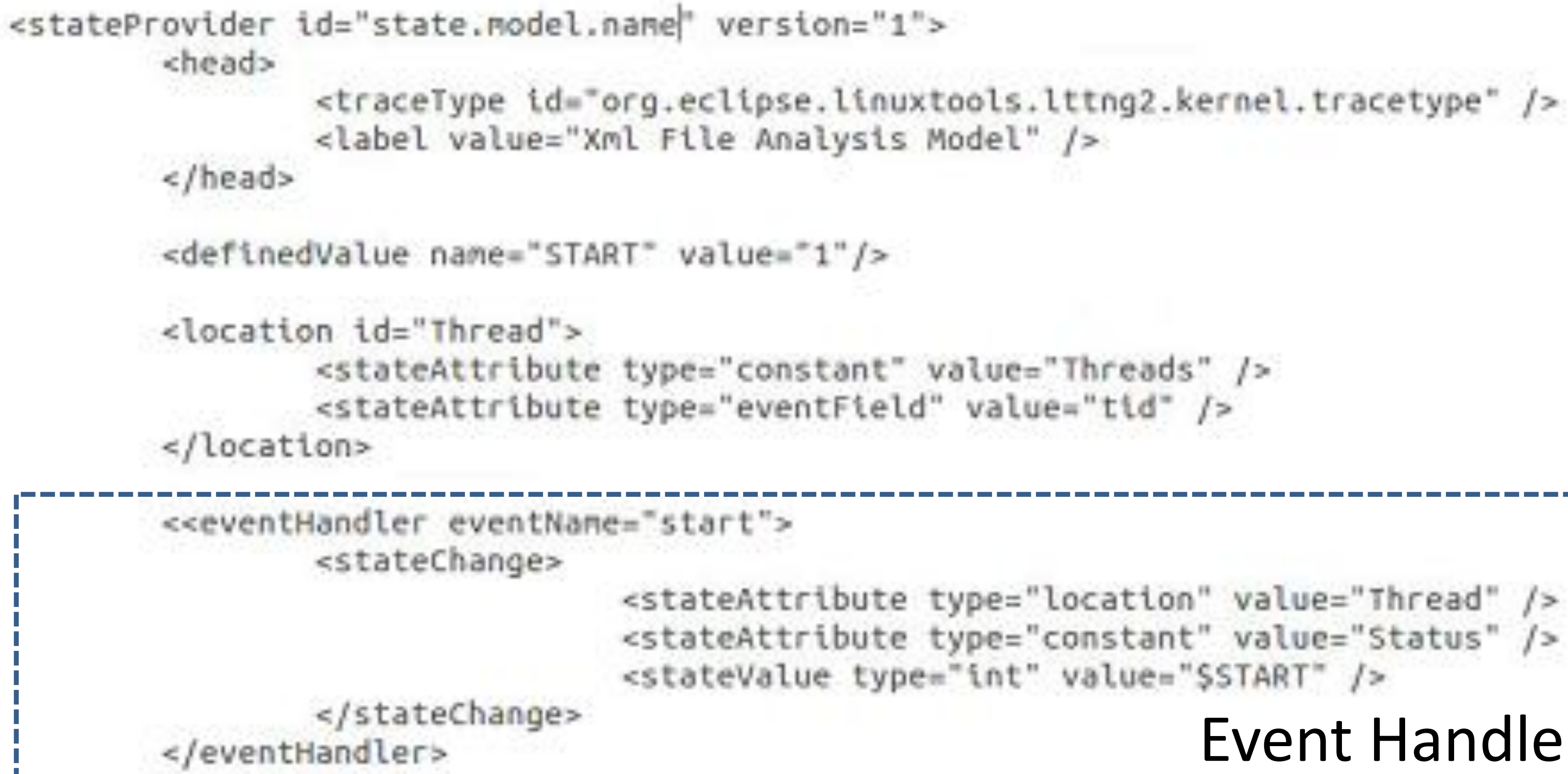


```
* Contributors:
*       Naser Ezzati - Initial API and implementation
************************************************************************ -->
<tmfxml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="stateprovider.xsd">

        <timeGraphView id="org.eclipse.linuxtools.tmf.analysis.xml.ui.views.controlflow1">
                <head>
                        <analysis id="kernel.linux.sp" />
                        <label value="Xml File Access Analysis" />
                </head>
                <!-- StateValues -->
                <definedValue name="FILE_OPEN" value="101" color="#03F718" />
                <definedValue name="FILE_CLOSE" value="102" color="#130101" />
                <definedValue name="FILE_READ" value="103" color="#0000EE" />
                <definedValue name="FILE_WRITE" value="104" color="#DB4B09" />
                <definedValue name="FILE_SEEK" value="105" color="#30D4C1" />
                <definedValue name="FILE_FCHMOD" value="106" color="#DB0000" />
                <definedValue name="FILE_FCHOWN" value="107" color="#302234" />

                <!-- Control Flow View -->
                <entry path="Threads/*">
                        <display type="self" />
                        <entry path="Files/*">
                                <display type="constant" value="Status" />
                                <name type="self" />
                        </entry>
                </entry>
        </timeGraphView>

        <stateProvider id="kernel.linux.sp" version="1">
                <head>
                        <traceType id="org.eclipse.linuxtools.lttng2.kernel.tracetype" />
                        <label value="Xml File Analysis Model" />
                </head>
                <!-- StateValues -->

                <definedValue name="FILE_OPEN" value="101" />
                <definedValue name="FILE_CLOSE" value="102" />
                <definedValue name="FILE_READ" value="103" />
                <definedValue name="FILE_WRITE" value="104" />
                <definedValue name="FILE_SEEK" value="105" />
```

VIEW

Model

State Provider:: Event Handler

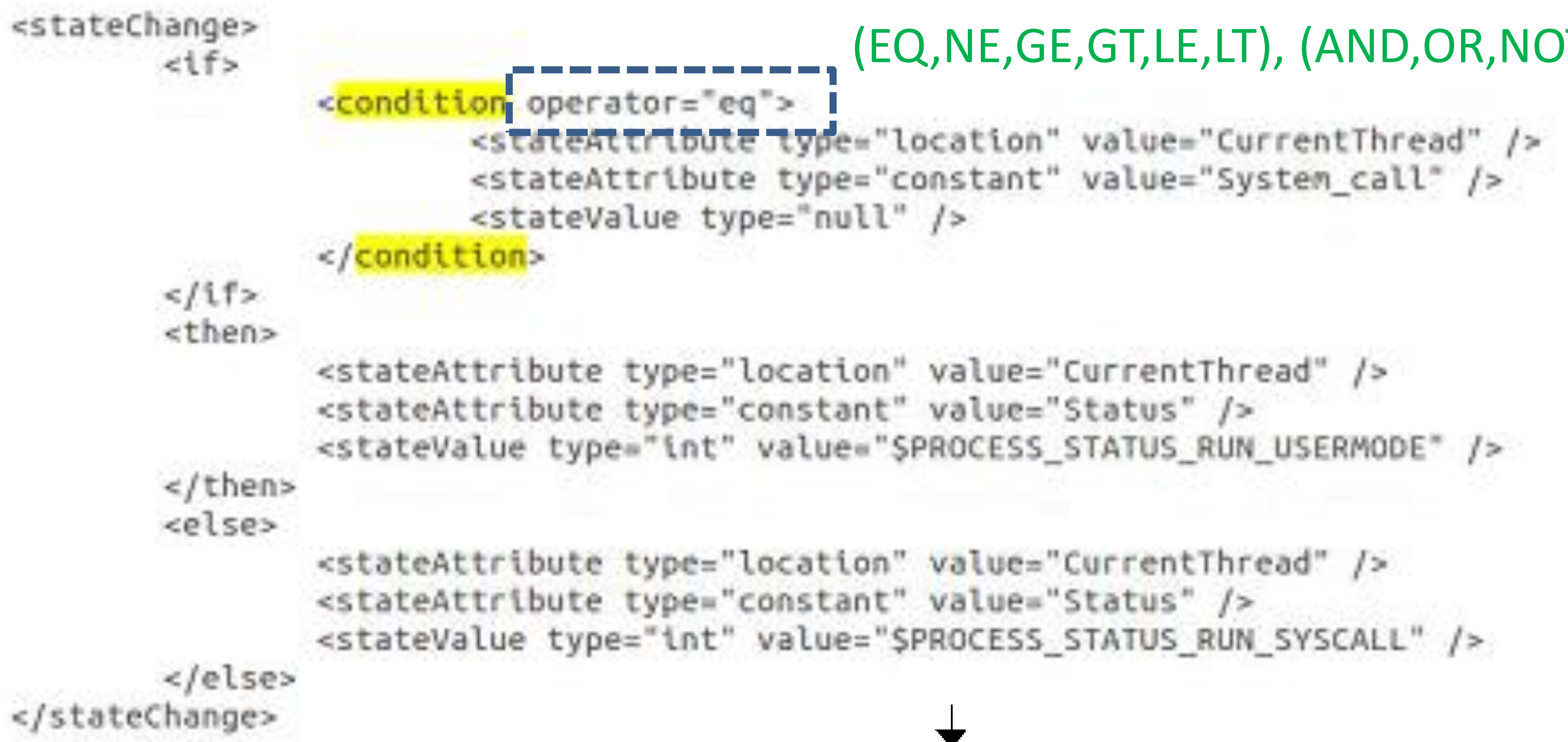


```
<stateProvider id="state.model.name" version="1">
        <head>
                <traceType id="org.eclipse.linuxtools.lttng2.kernel.tracetype" />
                <label value="Xml File Analysis Model" />
        </head>

        <definedValue name="START" value="1"/>

        <location id="Thread">
                <stateAttribute type="constant" value="Threads" />
                <stateAttribute type="eventField" value="tid" />
        </location>

        <<eventHandler eventName="start">
                <stateChange>
                                <stateAttribute type="location" value="Thread" />
                                <stateAttribute type="constant" value="Status" />
                                <stateValue type="int" value="$START" />
                </stateChange>
        </eventHandler>
```

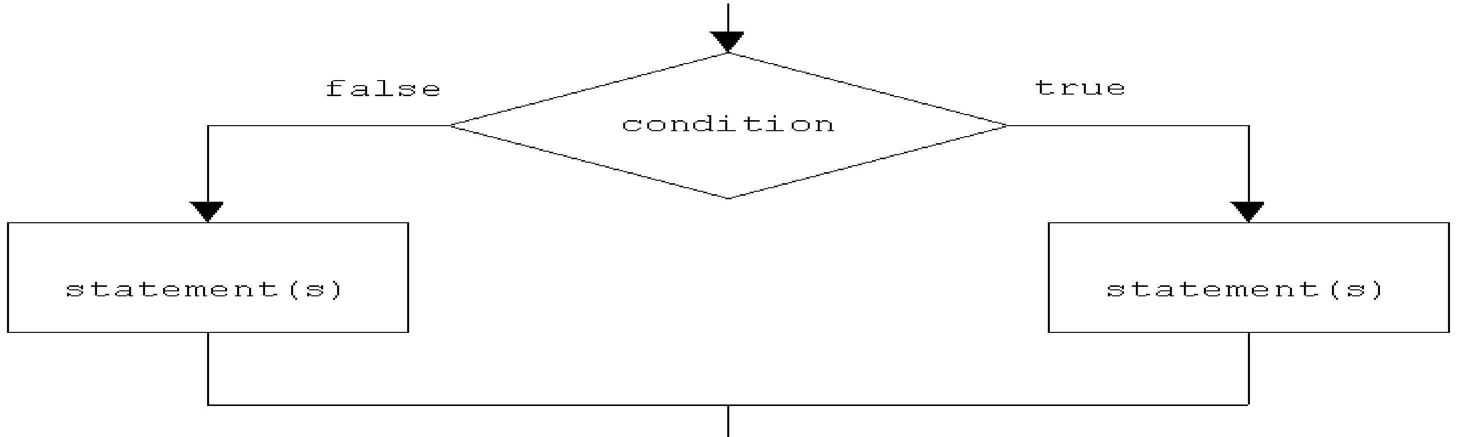
Event Handler

Event Handler:: Condition

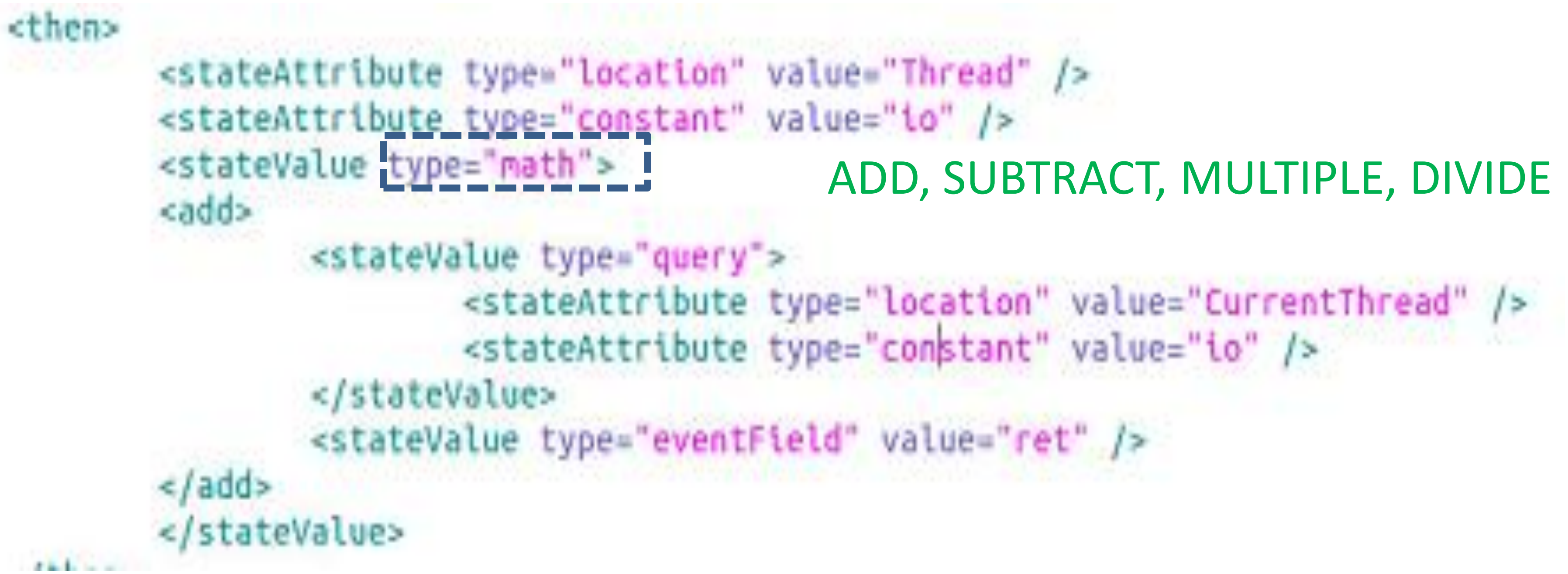
(EQ,NE,GE,GT,LE,LT), (AND,OR,NOT)



```
<stateChange>
        <if>
                <condition operator="eq">
                        <stateAttribute type="location" value="CurrentThread" />
                        <stateAttribute type="constant" value="System_call" />
                        <stateValue type="null" />
                </condition>
        </if>
        <then>
                <stateAttribute type="location" value="CurrentThread" />
                <stateAttribute type="constant" value="Status" />
                <stateValue type="int" value="$PROCESS_STATUS_RUN_USERMODE" />
        </then>
        <else>
                <stateAttribute type="location" value="CurrentThread" />
                <stateAttribute type="constant" value="Status" />
                <stateValue type="int" value="$PROCESS_STATUS_RUN_SYSCALL" />
        </else>
</stateChange>
```



Event Handler:: MATH

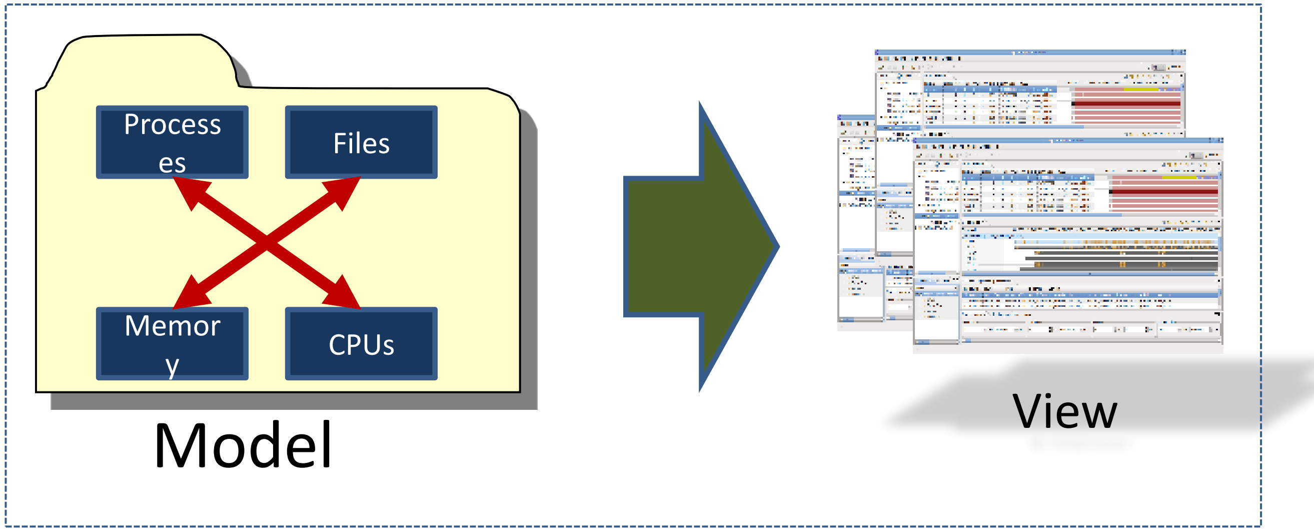


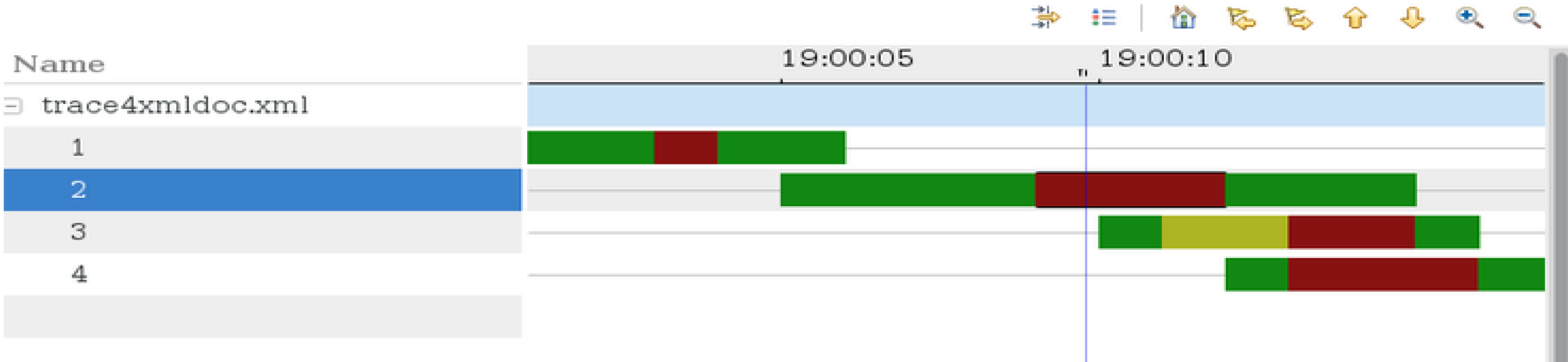
```
<then>
        <stateAttribute type="location" value="Thread" />
        <stateAttribute type="constant" value="io" />
        <stateValue type="math">
        <add>
                <stateValue type="query">
                        <stateAttribute type="location" value="CurrentThread" />
                        <stateAttribute type="constant" value="io" />
                </stateValue>
                <stateValue type="eventField" value="ret" />
        </add>
        </stateValue>
```

ADD, SUBTRACT, MULTIPLE, DIVIDE

View

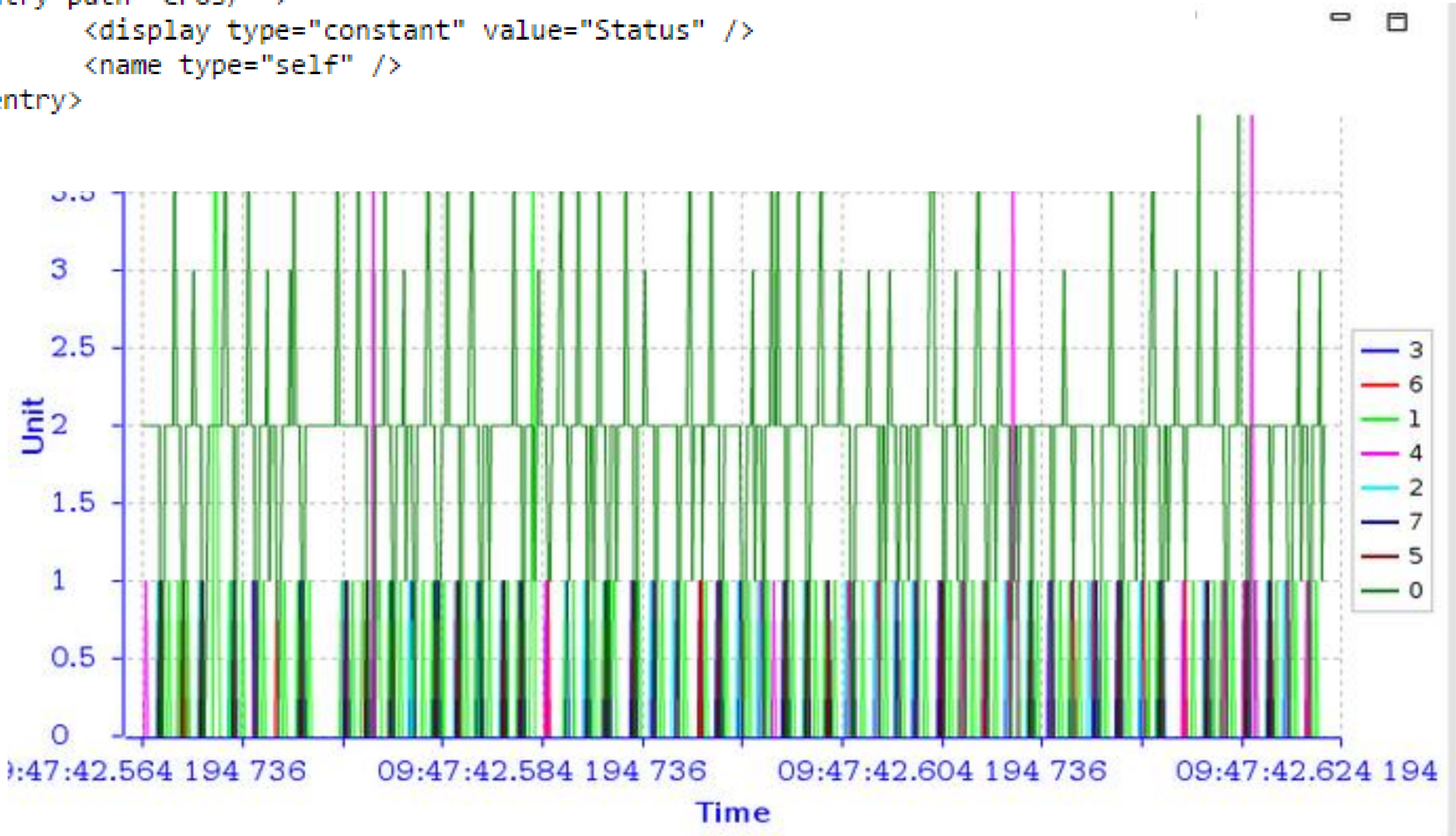
- Time graph view
  - Label





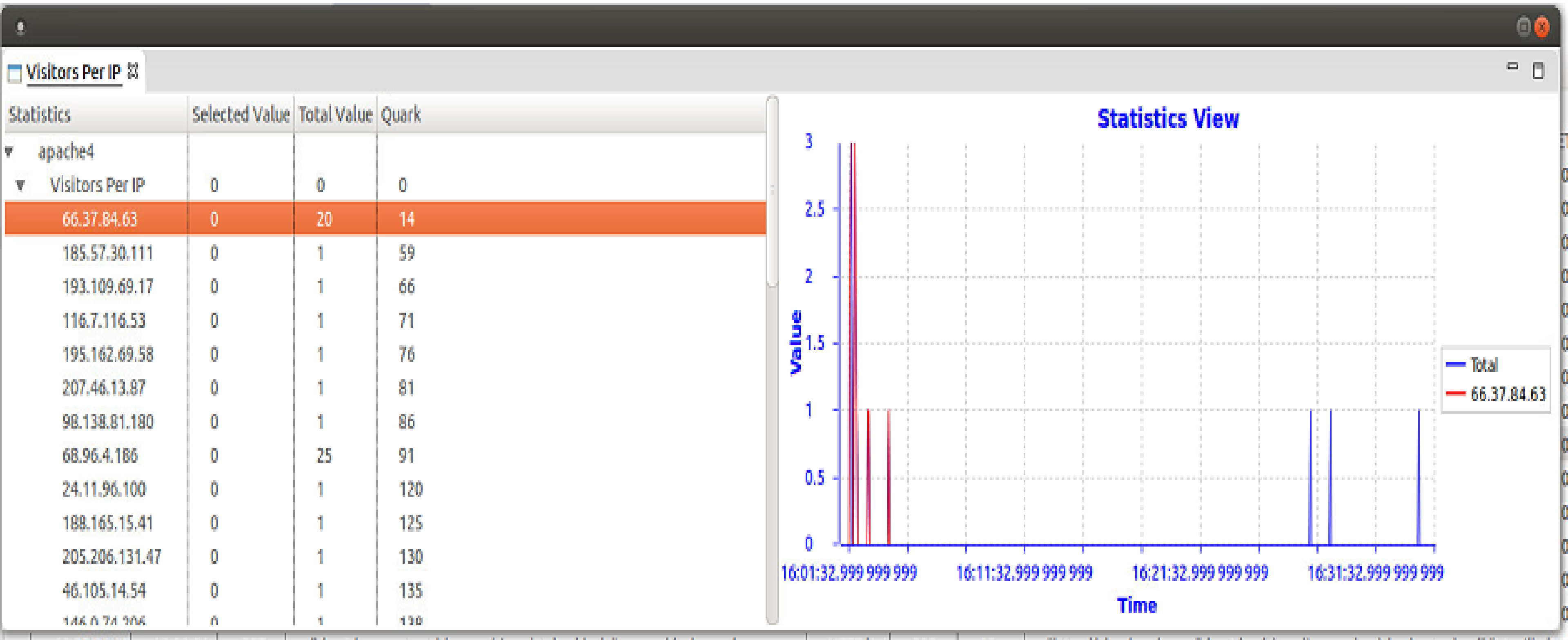
New View: XY Chart

```
<tmfxml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation=".
        <xyView id="my.test.xy.chart.view">
                <head>
                        <analysis id="org.eclipse.linuxtools.lttng2.kernel.analysis" />
                        <label value="CPU status XY view" />
                </head>

                <entry path="CPUs/*">
                        <display type="constant" value="Status" />
                        <name type="self" />
                </entry>
        </xyView>
</tmfxml>
```



New View: Statistics View

```
<statisticsView id="org.eclipse.linuxtools.tmf.analysis.xml.ui.views.statesystem.customtrace.statistics">
        <head>
            <analysis id="apache.ss2" />
            <label value="Visitors Statistcs" />
        </head>
        <entry path="Hosts/*" displayType="delta" xyView="true">
            <display type="constant" value="count"/>
            <metric value="Visitors per IP" />
        </entry>
</statisticsView>
```



Statistics	Selected Value	Total Value	Quark
▾ apache4			
▾ Visitors Per IP	0	0	0
66.37.84.63	0	20	14
185.57.30.111	0	1	59
193.109.69.17	0	1	66
116.7.116.53	0	1	71
195.162.69.58	0	1	76
207.46.13.87	0	1	81
98.138.81.180	0	1	86
68.96.4.186	0	25	91
24.11.96.100	0	1	120
188.165.15.41	0	1	125
205.206.131.47	0	1	130
46.105.14.54	0	1	135
146.0.74.206	0	1	139

Complex Patterns (under test)

- Current State changes are based on single events.
- We may need some state changes based on “a pattern of events” or a more processing of events/state system for specific use cases.
- State changes based on a pattern/group of events:
  - A pattern of
    - Events (and arguments),
    - States,
    - Abstract Events (new generated events)
  - Output can be:
    - A (group of) State change(s),
    - A new event (synthetic/abstract) event,
    - A change in view (highlight/hide/…)
  - Filtering
  - Abstraction
  - Fault Detection

User Interface

- Manually writing XML
  - difficult for some of you.
- A new tool is coming soon to help users to define their patterns and state changes graphically.

Example1: Kernel Trace (Video1)

Example2: Apache Log Analysis (Video2)

Future Work

- State Provider Extension:
  - Variables
  - String manipulation functions and regular expressions
  - Complex Patterns
  - Triggers
- Library of XML scenarios:
  - Fault Detection
    - bottlenecks
  - Attacks
- New Views:
  - Bar chart
  - Sequence diagram and …
- A GUI to define patterns and scenarios,
- Performance