Alarms @ Elettra
Graziano Scalamera
What has been done

1) Fixed many bugs

2) Solved some performance issues

3) Changed some alarm attribute names to be compatible with PANIC gui (name → tag, level → priority)

4) Implemented alarmSummary attribute with a syntax to be compatible with PANIC gui

5) Implemented GetAlarmInfo to return also evaluated formula values in JSON format

6) First tests to handle AlarmHandler alarms in PANIC GUI
**PANIC integration: something has been done**

<table>
<thead>
<tr>
<th>Alarm configuration</th>
<th>Alarm attributes</th>
<th>PyAlarm</th>
<th>Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alarm attributes</td>
<td>name, formula, level,</td>
<td>tag_name, formula, severity, description</td>
<td>Test if Attribute properties performant with ~1000 alarms</td>
</tr>
<tr>
<td></td>
<td>message</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alarm group</th>
<th>grouped using group alarm attribute</th>
<th>grouped using class/device/view</th>
<th>Support both</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Alarm actions</th>
<th>on_command, off_command</th>
<th>receivers</th>
<th>AlarmHandler should support more receivers, agree on syntax</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Alarm-severity</th>
<th>level</th>
<th>severity</th>
<th>priority</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Interface: enum values</th>
<th>NORM, UNACK, ACKED, RTNUN, SHLVD, DSUPR, OOSRV</th>
<th>Bool-attributes</th>
<th>Support enum values with NORM=0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Interface: attributes</th>
<th>Enum attributes one per alarm + one string array for each alarm state</th>
<th>Bool (+quality) attributes one per alarm + string arrays with coded information in it</th>
<th>Enum attributes + AlarmSummary string array with everything needed coded in it as key=value;...</th>
</tr>
</thead>
</table>

**AlarmHandler**

**PyAlarm**

**Integration**
PANIC integration: first tests

1) PANIC gui only looks for devices whose Tango Class is “PyAlarm” → temporarily recompiled AlarmHandler with “PyAlarm” class name

2) PANIC gui does not look for alarms configured in attribute properties → temporarily cloned alarms configuration in device properties
HDB++ integration: first tests

The history of each alarm can be stored with HDB++ → easy to correlate with other attributes stored with HDB++.

tag=archiving_hdb_mod; formula=(archiving/hdb++archiver/attributenoknumber > 0); on_delay=20
AlarmHandler: performances

- An instance of AlarmHandler with 1000 configured alarms was tested.
- For example during 10 minutes of test, ~10000 of formula evaluations were performed.
- The most frequently evaluated alarm was checked ~200 times.
- No significant load on the system during the test.
- AlarmHandler exports the statistic of the number of evaluation per alarm:

![Alarm evaluation frequency chart]

```
Alarm index          Frequency
0 - 99              ~5
100 - 199            ~10
200 - 299            ~15
300 - 399            ~20
400 - 499            ~25
500 - 599            ~30
600 - 699            ~35
700 - 799            ~40
800 - 899            ~45
900 - 999            ~50
1000                  ~55
```