Elettra Sincrotrone Trieste

NEW DEVELOPMENTS FOR THE TANGO ALARM SYSTEM

TUPHA165

G. Scalamera, L. Pivetta[1], Elettra-Sincrotrone Trieste, Trieste, Italy
S. Rubio-Manrique, ALBA-CELLS Synchrotron, Cerdanyola del Vallès, Barcelona, Spain
[1]also at SKA Organisation, Macclesfield, UK

The TANGO Alarm System, based on an efficient event-driven, highly configurable rule-based engine named AlarmHandler, has undergone a deep refactoring. The dedicated MySQL database has been dropped; the TANGO database now stores all the configuration whereas the HDB++ historical database keeps all the alarms history. Correlating alarms with any other engineering data is now much simpler. A dynamic attribute is provided for each alarm rule; this allows to easily build a hierarchy of AlarmHandlers. The AlarmHandler manages Attribute quality in the alarm rules and provides possible exceptions resulting in alarm evaluation. Mathematical functions, such as sin, cos, pow, min, max and ternary conditionals are available in the alarm formulae. The TANGO AlarmHandler device server is now based on the IEC 62682 standard.

One dynamic attribute created for each alarm

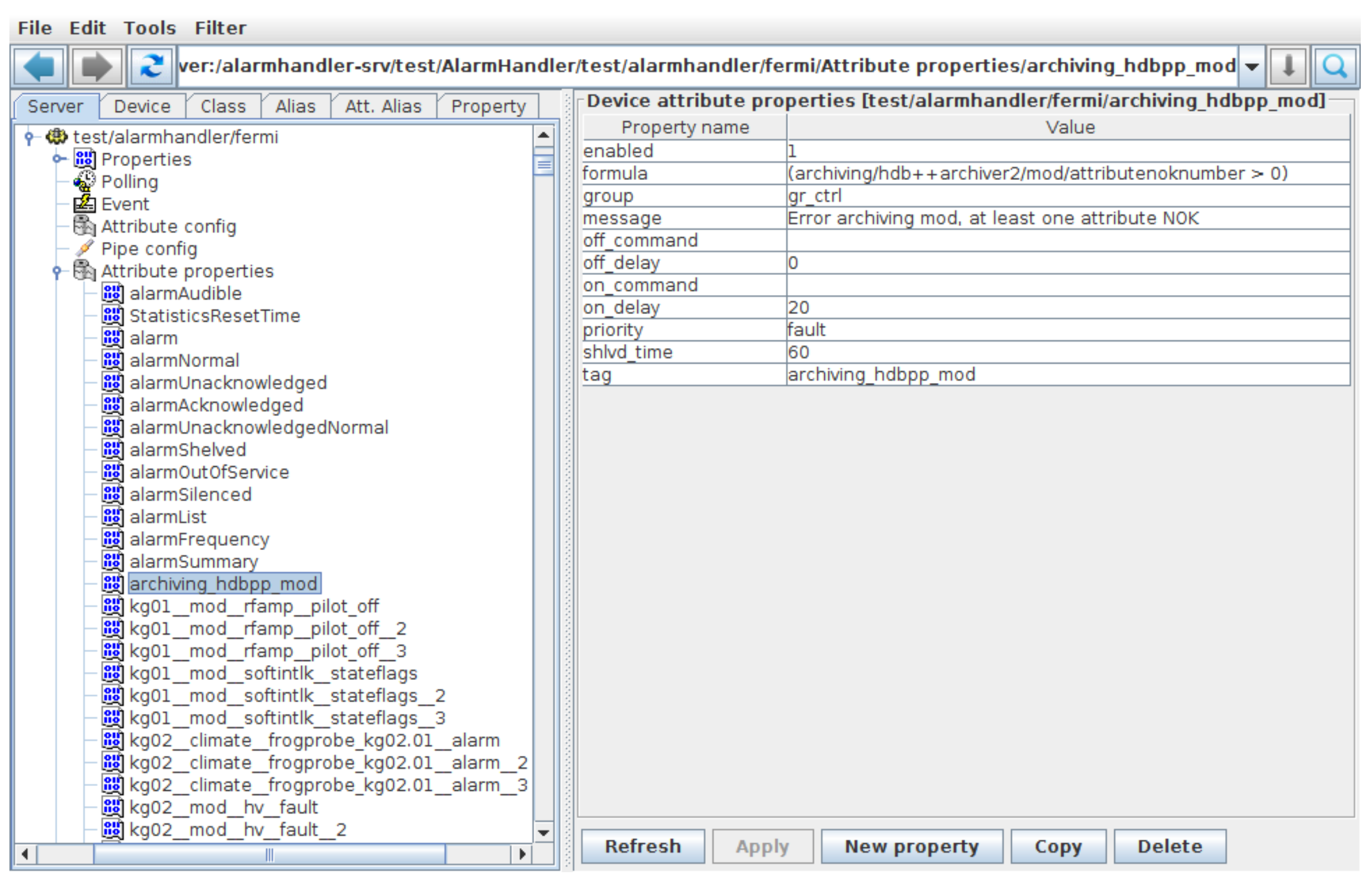
- It pushes change/archives events as soon as its state change
- It is of enumerated type to contain informations on process condition, acknowledged status, enabled / disabled:

NORM	Normal alarm state	Normal process condition	Acknowledged	Enabled
UNACK	Unacknowledged alarm state	Abnormal process condition	Unacknowledged	Enabled
ACKED	Acknowledged alarm state	Abnormal process condition	Acknowledged	Enabled
RTNUN	Returned to normal alarm state	Normal process condition	Unacknowledged	Enabled
SHLVD	Shelved state	Normal or abnormal process condition	Not applicable	Temporarily disabled
DSUPR	Suppressed by design state	Normal or abnormal process condition	Not applicable	Disabled
OOSRV	Out of service alarm state	Normal or abnormal process condition	Not applicable	Disabled

- Its quality is consistent with the quality of attributes in the formula
- It throws exceptions if attributes in the formula are in error or if there are errors from the parser while evaluating

Alarm configuration stored in the Tango DB

The configuration of each alarm is stored in the attribute properties. It's easy to check and modify the configuration with Jive.


File Edit Tools Filter
ver:/alarmhandler-srv/test/AlarmHandler/test/alarmhandler/fermi/Attribute properties/archiving_hdbpp_mod
Server Device Class Alias Att. Alias Property
test/alarmhandler/fermi
Properties
Polling
Event
Attribute config
Pipe config
Attribute properties
alarmAudible
StatisticsResetTime
alarm
alarmNormal
alarmUnacknowledged
alarmAcknowledged
alarmUnacknowledgedNormal
alarmShelved
alarmOutOfService
alarmSilenced
alarmList
alarmFrequency
alarmSummary
archiving_hdbpp_mod
kg01__mod__rfamp__pilot_off
kg01__mod__rfamp__pilot_off__2
kg01__mod__rfamp__pilot_off__3
kg01__mod__softintlk__stateflags
kg01__mod__softintlk__stateflags__2
kg01__mod__softintlk__stateflags__3
kg02__climate__frogprobe_kg02.01__alarm
kg02__climate__frogprobe_kg02.01__alarm__2
kg02__climate__frogprobe_kg02.01__alarm__3
kg02__mod__hv__fault
kg02__mod__hv__fault__2
Refresh Apply New property Copy Delete

Device attribute properties [test/alarmhandler/fermi/archiving_hdbpp_mod]

Property name	Value
enabled	1
formula	(archiving/hdb++archiver2/mod/attributenoknumber > 0)
group	gr_ctrl
message	Error archiving mod, at least one attribute NOK
off_command	
off_delay	0
on_command	
on_delay	20
priority	fault
shlvd_time	60
tag	archiving_hdbpp_mod

Extended parser grammar

Some new predicates are supported:

Ternary conditional
  (expr1 ? expr2 : expr3)

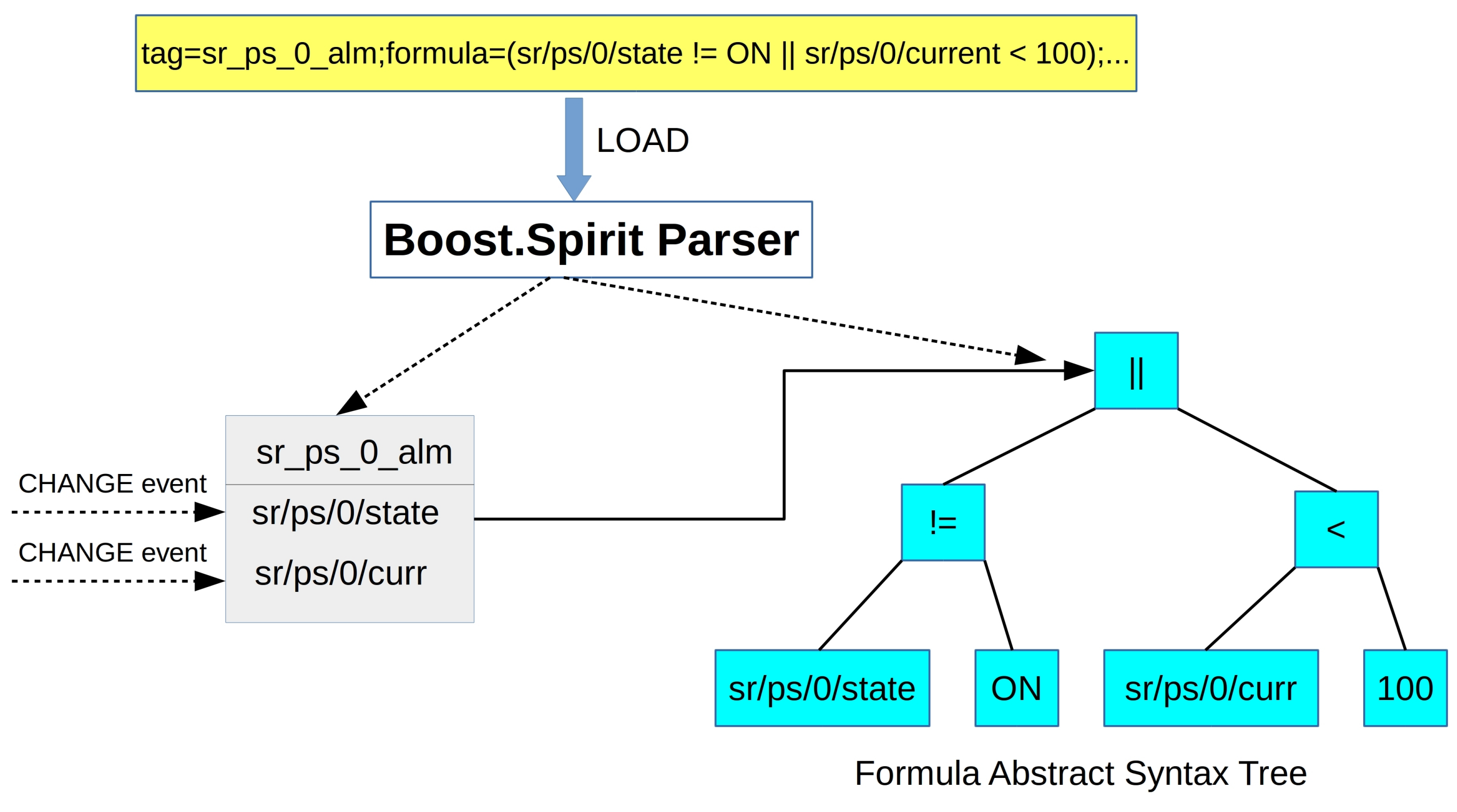
Quality
  name/of/device/attr.quality
  quality(expression)

Alarm condition
  name/of/device/attr.alarm
    (true if UNACK or ACK)
  name/of/device/attr.normal
    (true if NORM or RTNUN)

Functions
  max(expr1,expr2)
  min(expr1,expr2)
  pow(expr1,expr2)
  sin(expr) cos(expr)

High performance parser

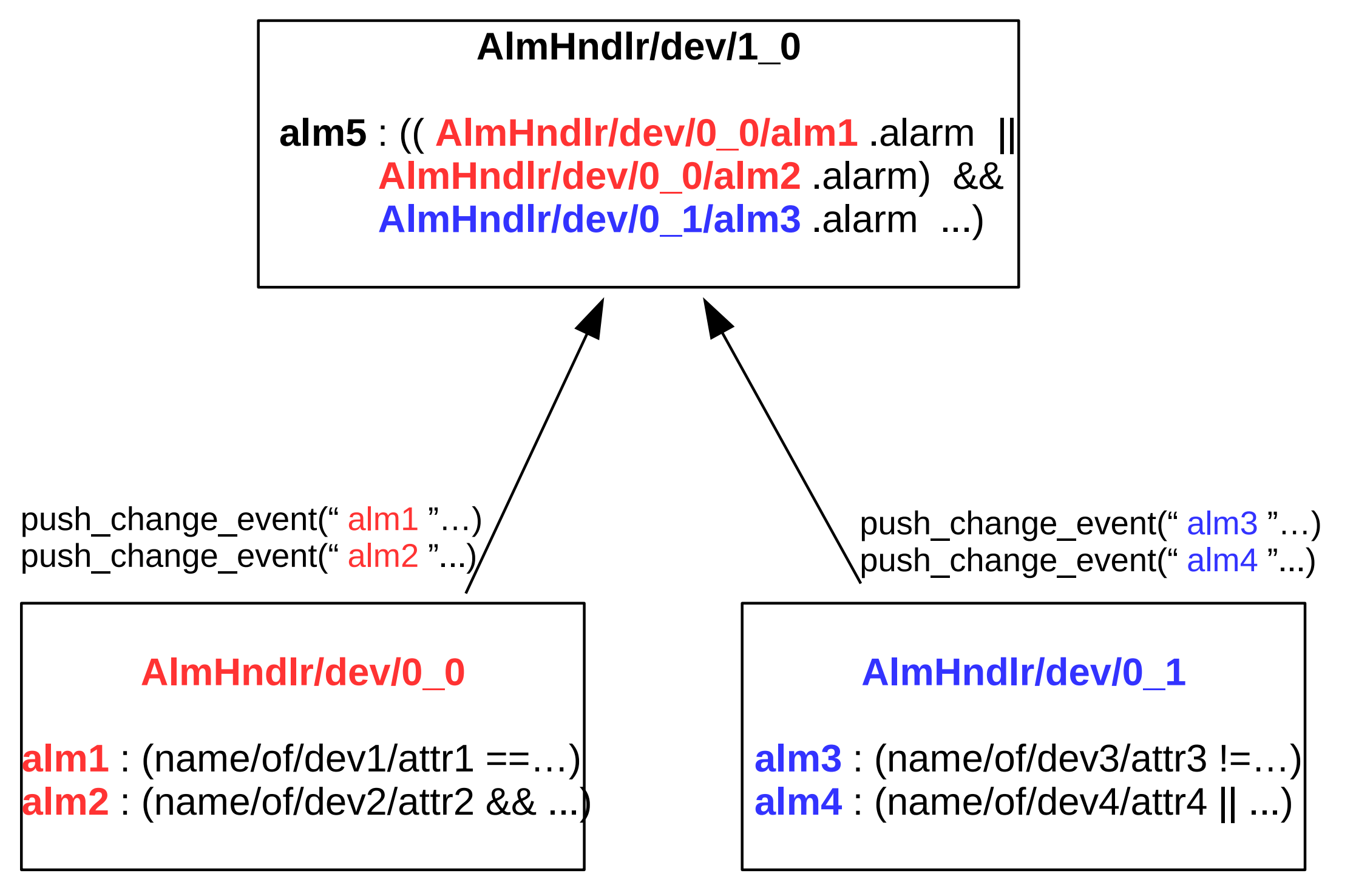
The alarm parser has been implemented, using the Boost.spirit libraries, to build an Abstract Syntax Tree (AST) as the result of the parsing of the formula. Thus, each alarm rule is parsed only once when loaded, evaluation is done using the AST for better performances.



Formula Abstract Syntax Tree

AlarmHandler hierarchy

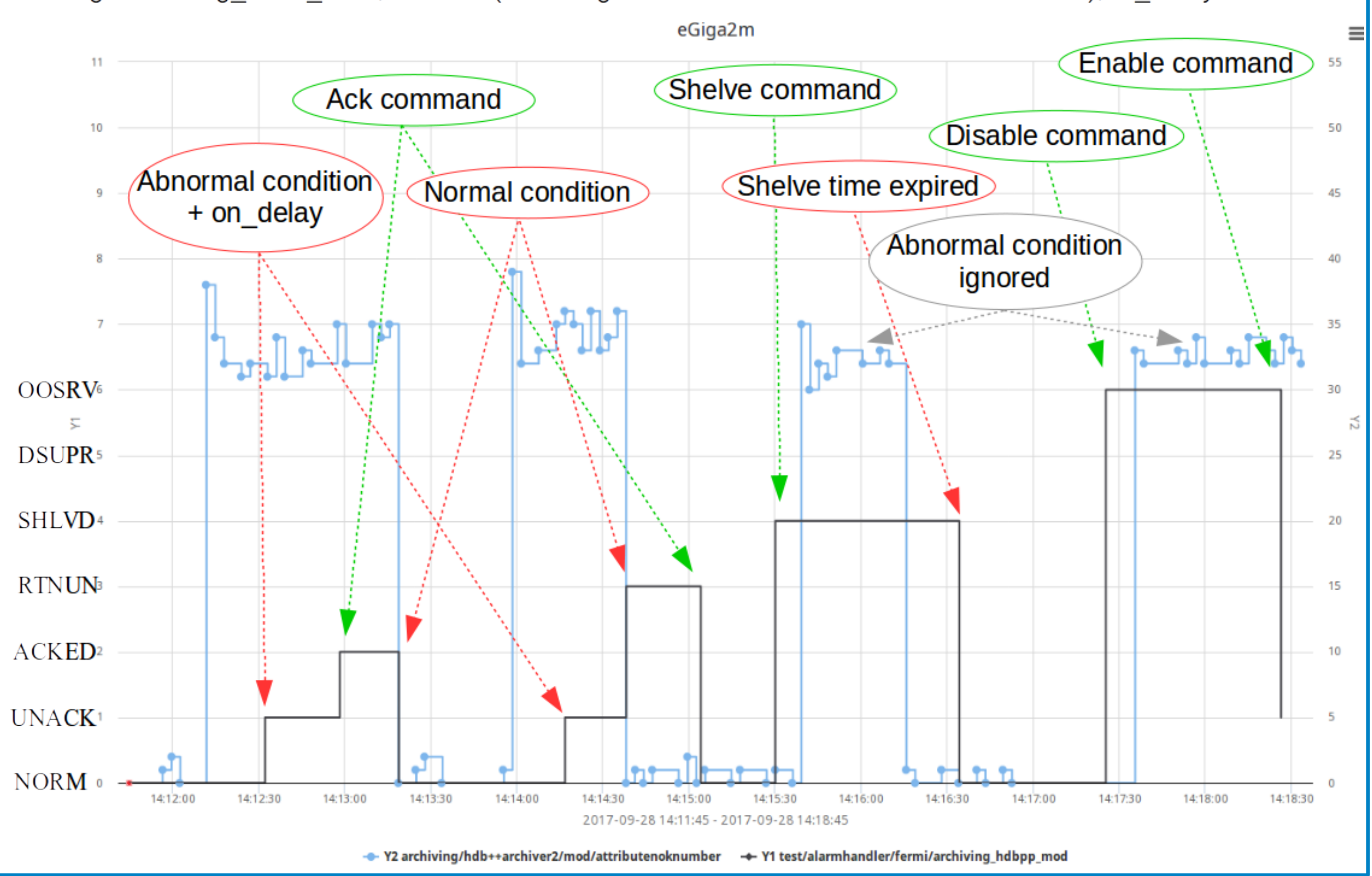
Easy to build a hierarchy of AlarmHandlers: just use an alarm attribute name in another alarm formula.


AlmHndlr/dev/1_0
alm5 : ((AlmHndlr/dev/0_0/alm1 .alarm ||
AlmHndlr/dev/0_0/alm2 .alarm) &&
AlmHndlr/dev/0_1/alm3 .alarm ...)
push_change_event(" alm1 "...)
push_change_event(" alm2 "...)
push_change_event(" alm3 "...)
push_change_event(" alm4 "...)
AlmHndlr/dev/0_0
alm1 : (name/of/dev1/attr1 ==...)
alm2 : (name/of/dev2/attr2 && ...)
AlmHndlr/dev/0_1
alm3 : (name/of/dev3/attr3 !=...)
alm4 : (name/of/dev4/attr4 || ...)

Alarm history stored with HDB++

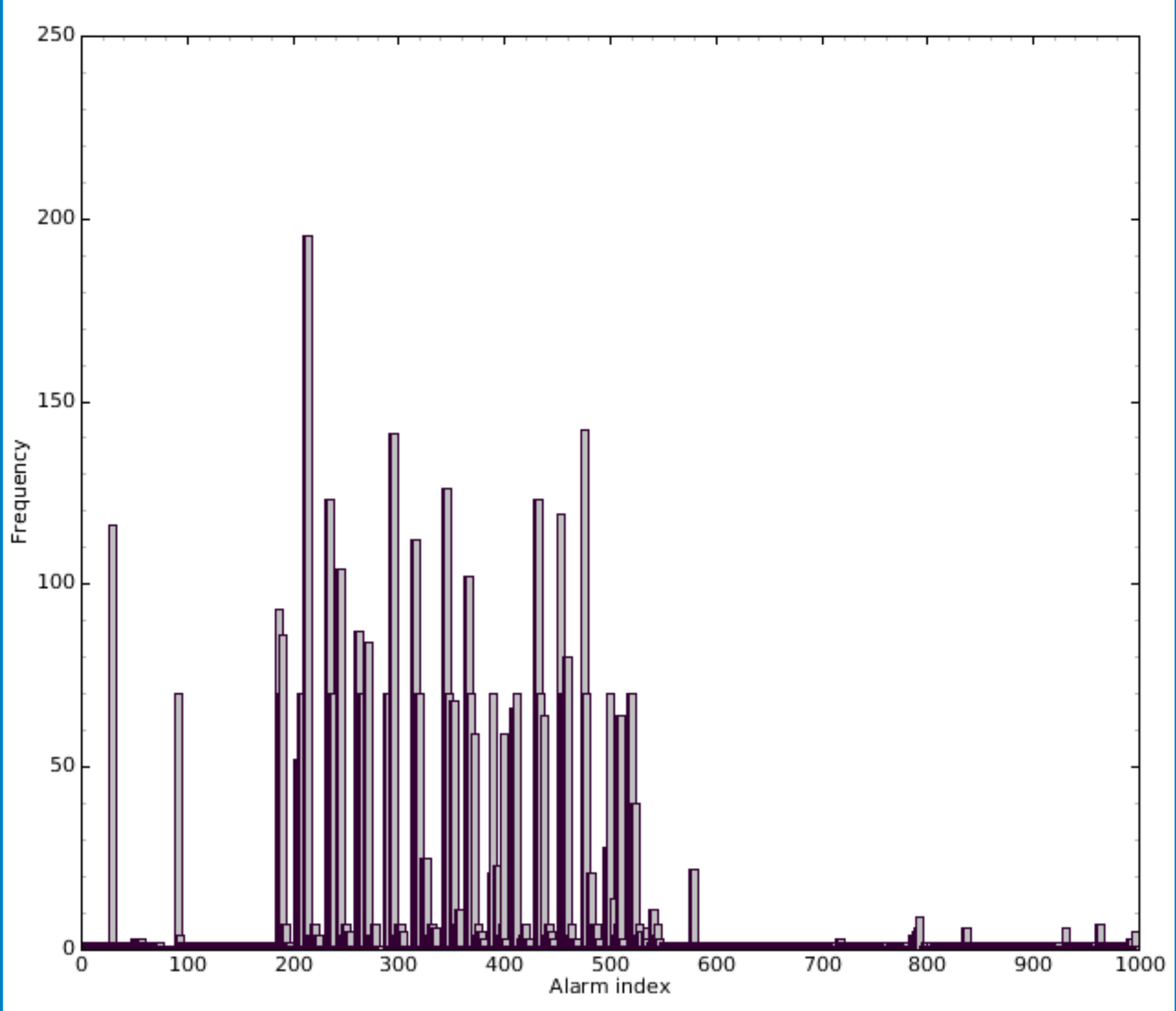
The history of each alarm is stored with HDB++. Every alarm is represented by an attribute which pushes archive events in the code. It is sufficient to configure and start archiviation with the HDB++ Configuration Manager and its value, quality and exceptions will be stored.
It is easy to correlate with other attributes in the system stored with HDB++.

tag=archiving_hdbb_mod;formula=(archiving/hdb++archiver/attributenoknumber > 0);on_delay=20



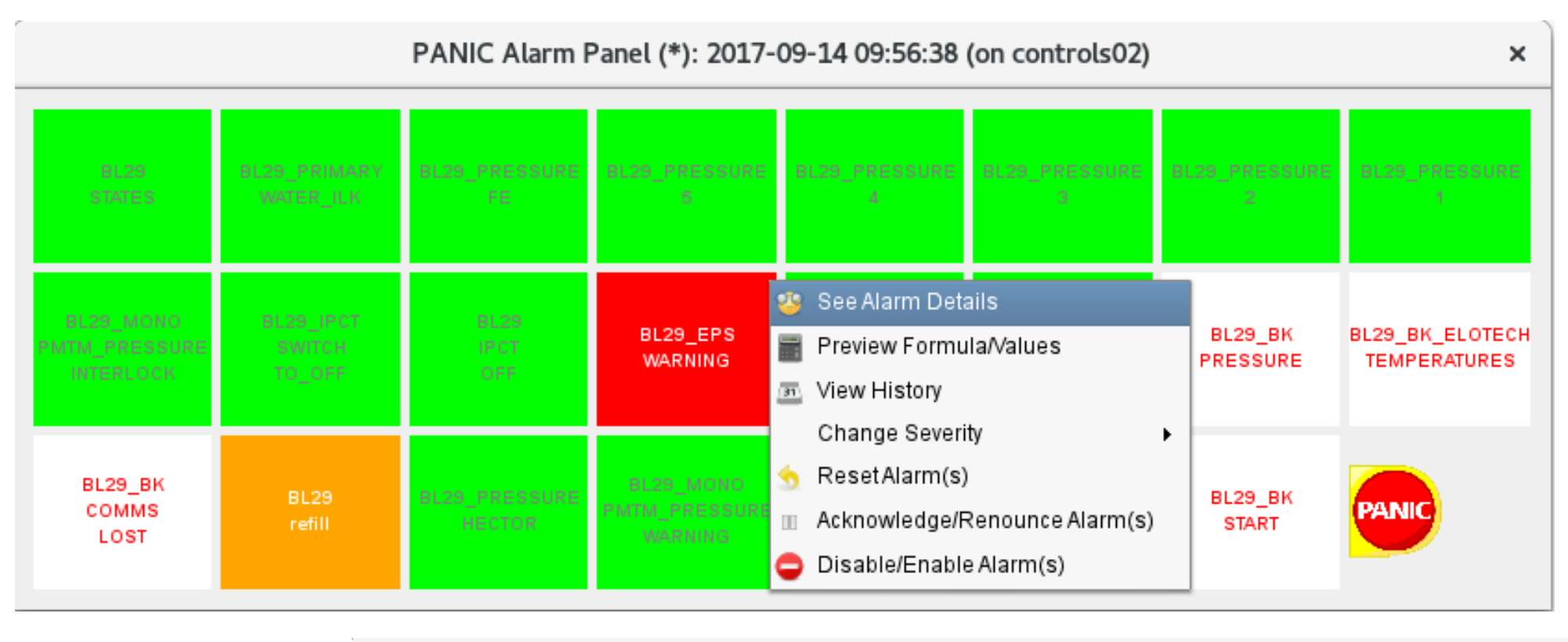
Performance with thousands of alarms

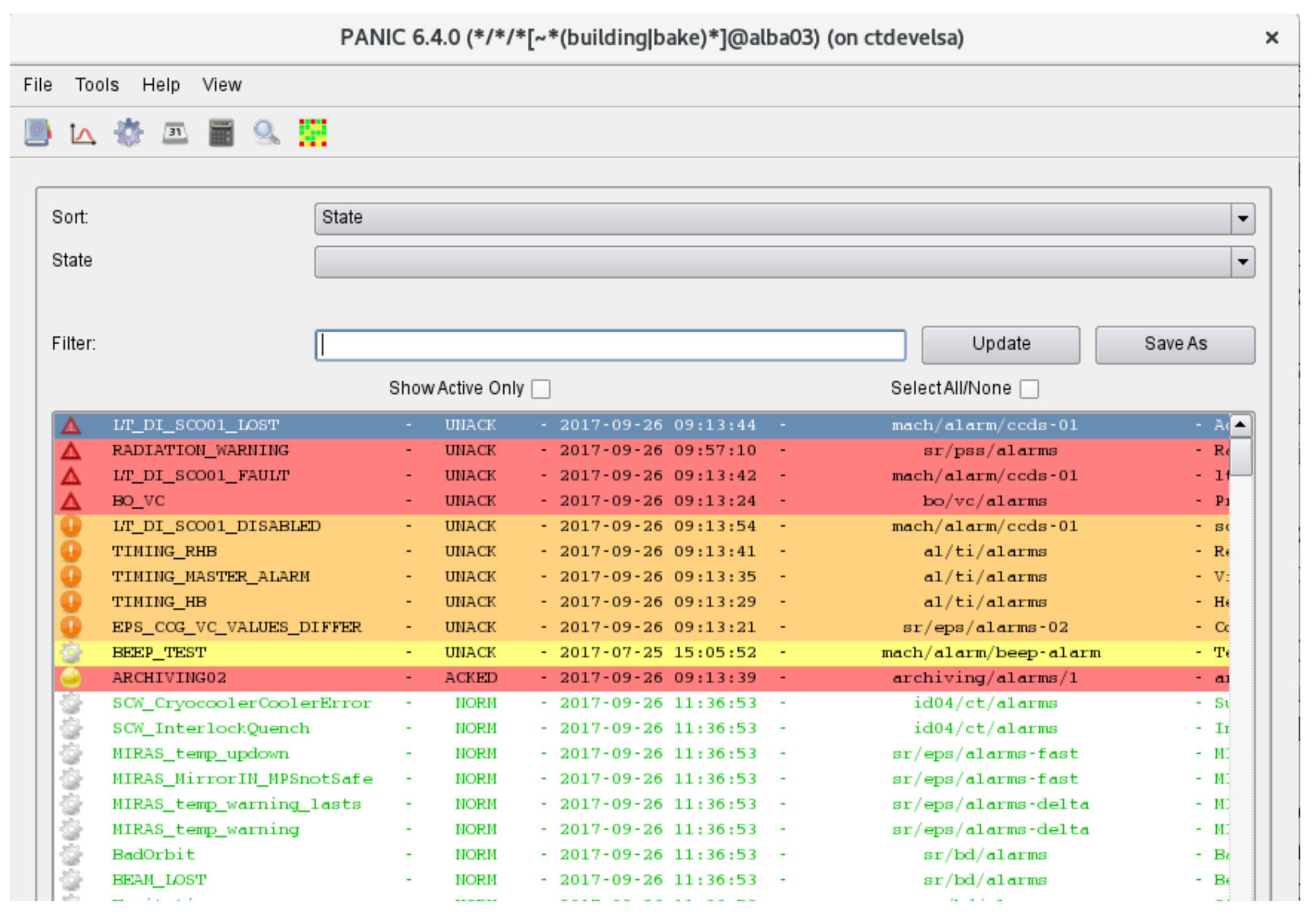
AlarmHandler is able to manage thousands of alarms in one instance. Tested with 1000 alarms evaluated once per minute on average without any significant load on the system.
It exports the statistic of the number of evaluation per alarm, in 10 minutes a peak of 200 per attribute was registered:



AlarmHandler integration with PANIC

Defined a common interface and a minimal set of shared features, PANIC GUIs can display AlarmHandler alarms.





```
LT_DI_SC001_LOST              -  UNACK  -  2017-09-26 09:13:44  -  mach/alarm/ccds-01
RADIATION_WARNING             -  UNACK  -  2017-09-26 09:57:10  -  sr/pss/alarms
LT_DI_SC001_FAULT             -  UNACK  -  2017-09-26 09:13:42  -  mach/alarm/ccds-01
BO_VC                         -  UNACK  -  2017-09-26 09:13:24  -  bo/vc/alarms
LT_DI_SC001_DISABLED          -  UNACK  -  2017-09-26 09:13:54  -  mach/alarm/ccds-01
TIMING_RHB                    -  UNACK  -  2017-09-26 09:13:41  -  al/ti/alarms
TIMING_MASTER_ALARM           -  UNACK  -  2017-09-26 09:13:35  -  al/ti/alarms
TIMING_HB                     -  UNACK  -  2017-09-26 09:13:29  -  al/ti/alarms
EPS_CCG_VC_VALUES_DIFFER      -  UNACK  -  2017-09-26 09:13:21  -  sr/eps/alarms-02
BEEP_TEST                     -  UNACK  -  2017-07-25 15:05:52  -  mach/alarm/beep-alarm
ARCHIVING02                   -  ACKED  -  2017-09-26 09:13:39  -  archiving/alarms/1
SCW_CryocoolerCoolerError     -  NORM   -  2017-09-26 11:36:53  -  id04/ct/alarms
SCW_InterlockQuench           -  NORM   -  2017-09-26 11:36:53  -  id04/ct/alarms
MIRAS_temp_updown             -  NORM   -  2017-09-26 11:36:53  -  sr/eps/alarms-fast
MIRAS_MirrorIN_MPSnotSafe     -  NORM   -  2017-09-26 11:36:53  -  sr/eps/alarms-fast
MIRAS_temp_warning_lasts      -  NORM   -  2017-09-26 11:36:53  -  sr/eps/alarms-delta
MIRAS_temp_warning            -  NORM   -  2017-09-26 11:36:53  -  sr/eps/alarms-delta
BadOrbit                      -  NORM   -  2017-09-26 11:36:53  -  sr/bd/alarms
BEAM_LOST                     -  NORM   -  2017-09-26 11:36:53  -  sr/bd/alarms
```

Useful links:
https://github.com/ELETTRA-SincrotroneTrieste/alarmhandler
https://github.com/tango-controls/PANIC
www.tango-controls.org