



# NEW DEVELOPMENTS FOR THE TANGO ALARM SYSTEM

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

Elettra Sincrotrone Trieste

TUPHA165

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

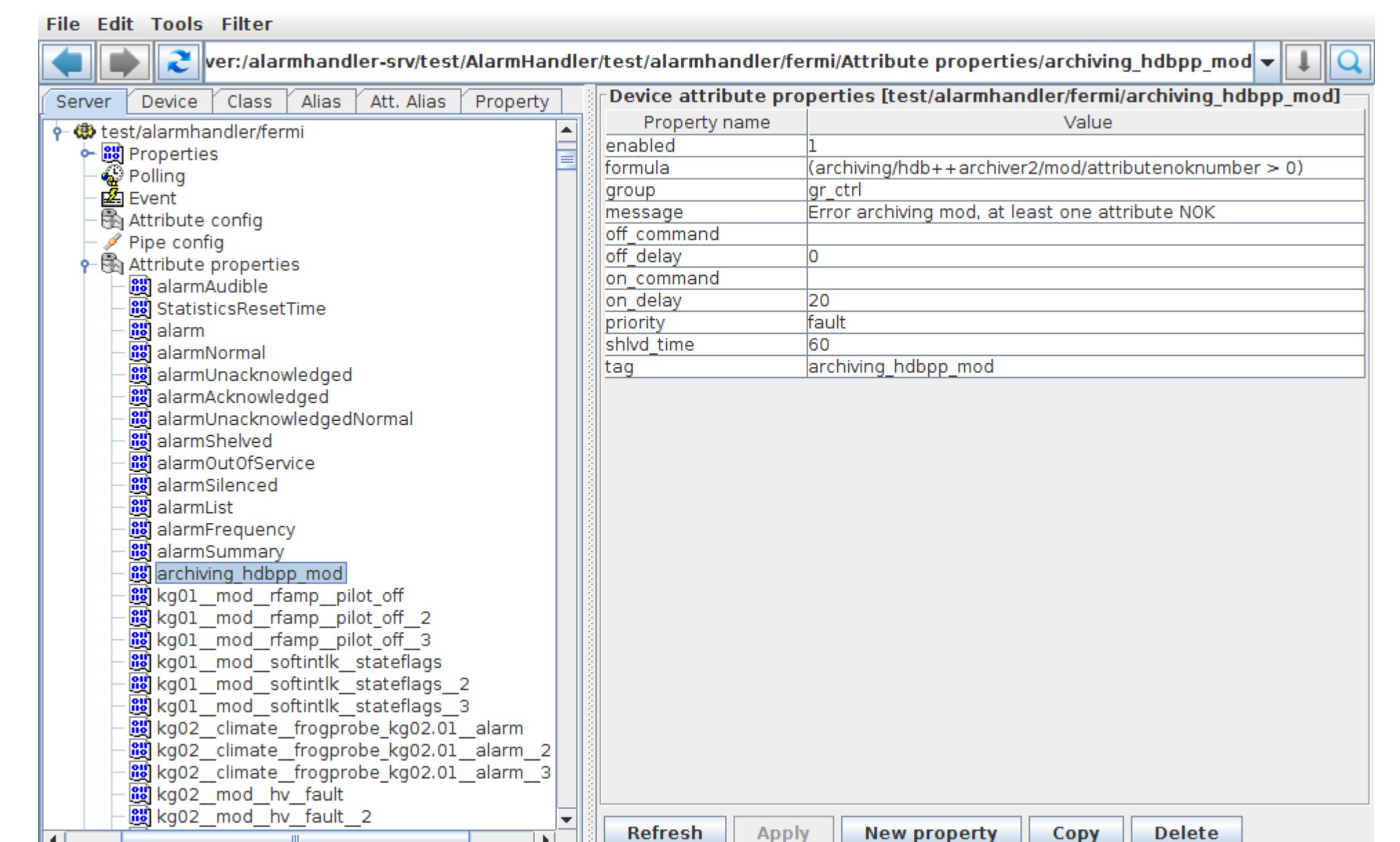
- It pushes change/archives events as soon as its state change
- It is of enumerated type to contain informations on process condition, acknowledged status, enabled / disabled:

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.



## Extended parser grammar

Some new predicates are supported:

Ternary conditional  
(**expr1 ? expr2 : expr3**)

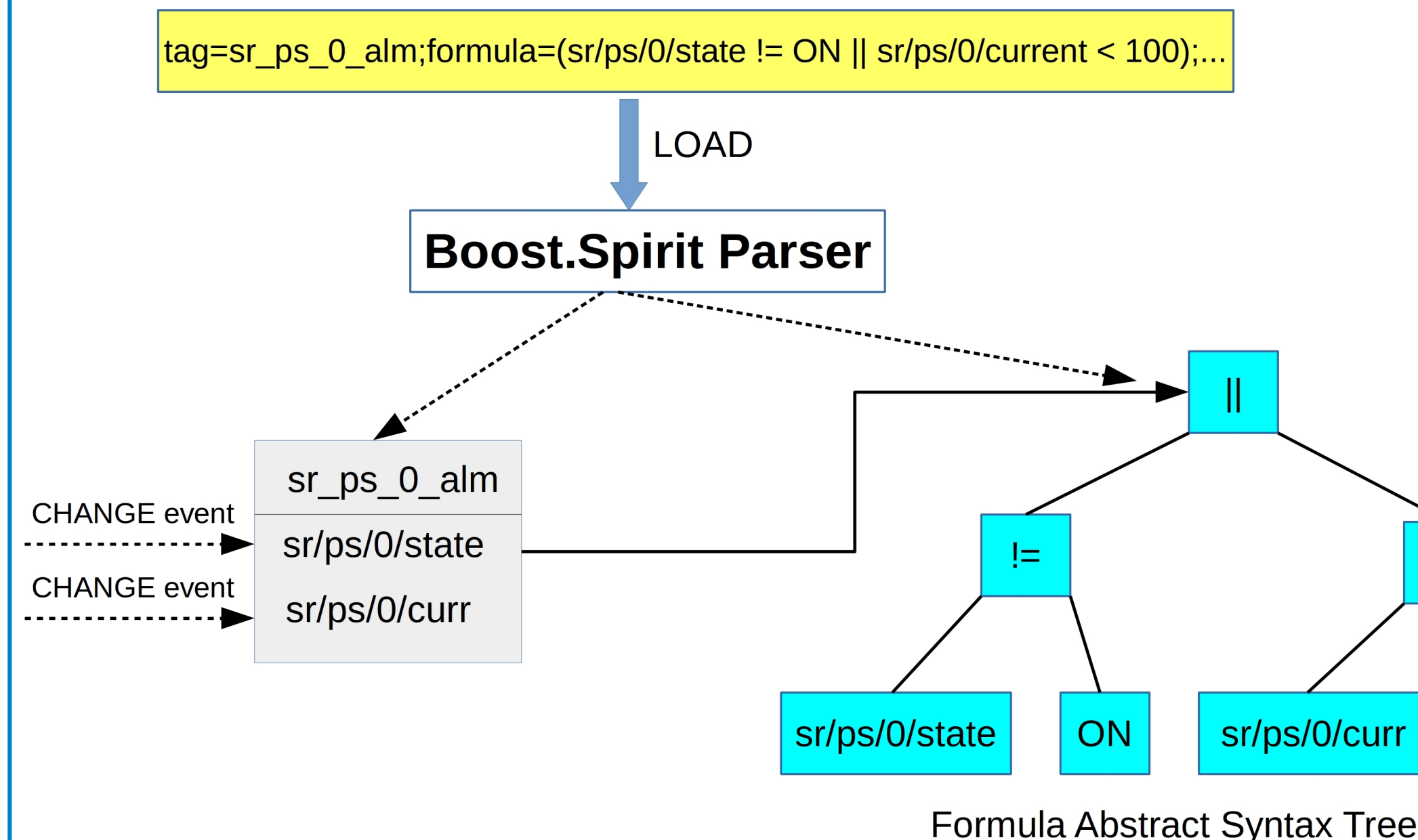
Quality  
**name/of/device/attr.quality**  
**quality(expression)**

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)**

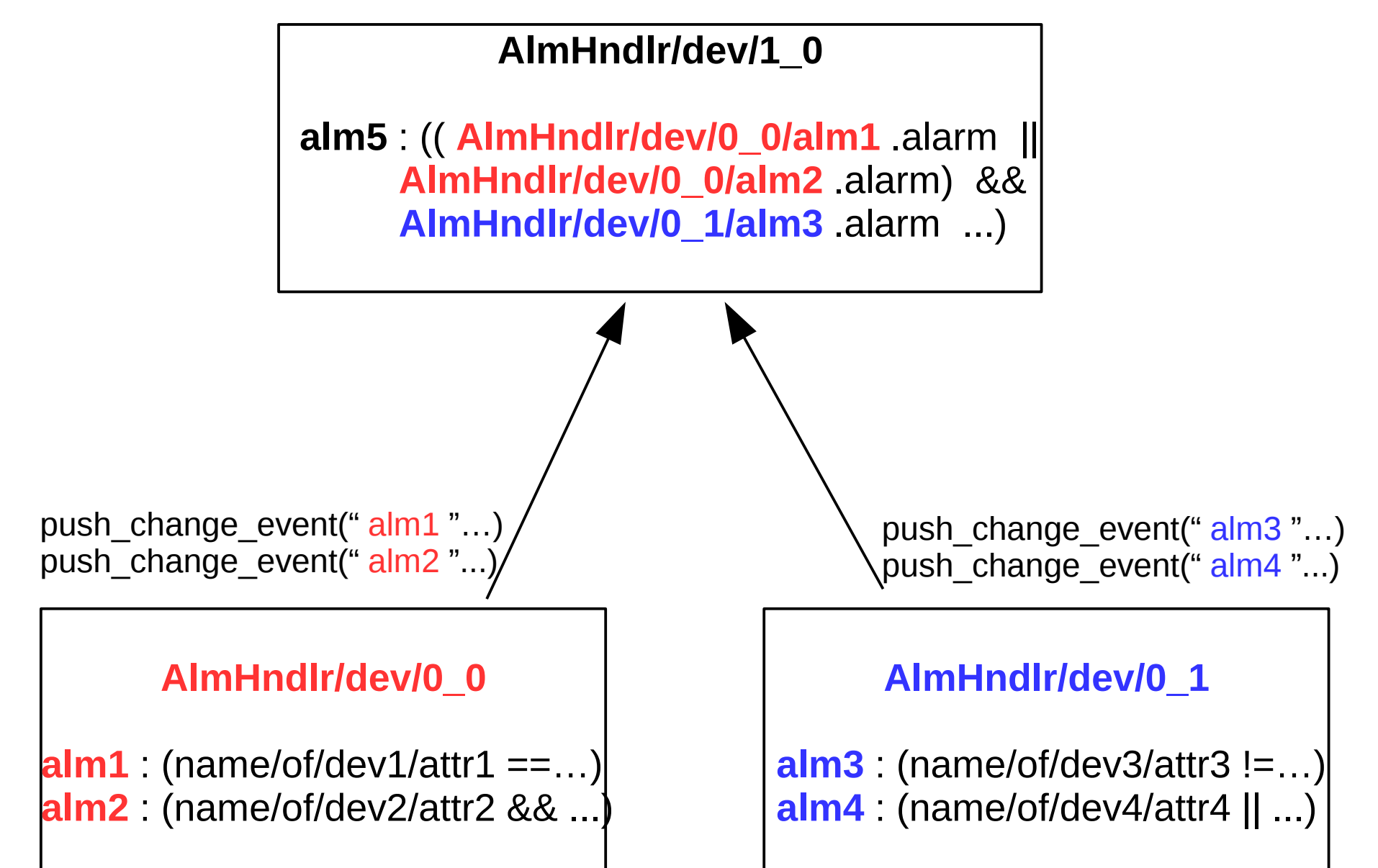
## High performance parser

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.



## AlarmHandler hierarchy

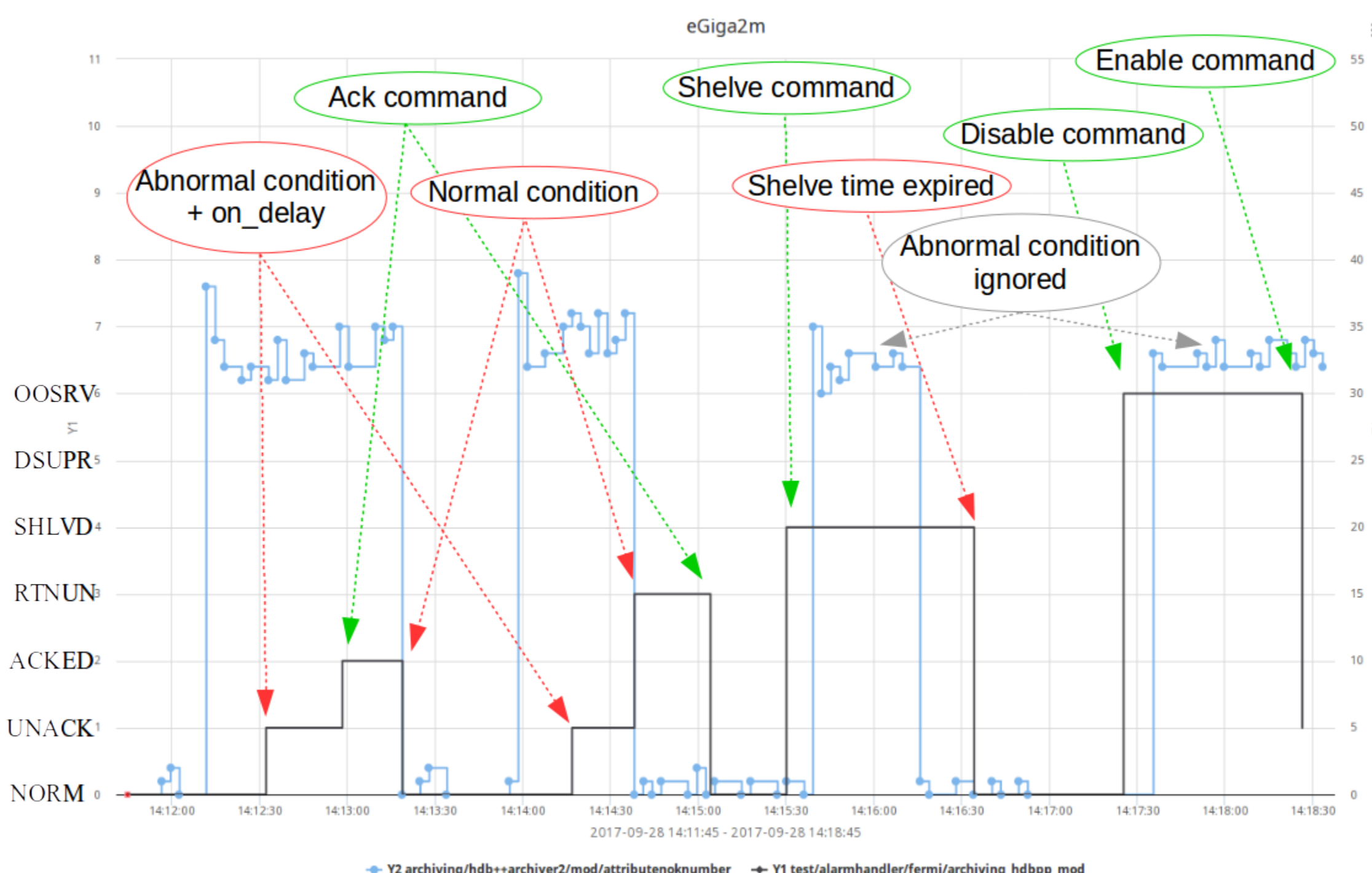
Easy to build a hierarchy of AlarmHandlers: just use an alarm attribute name in another alarm formula.



## 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 archivation 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++.

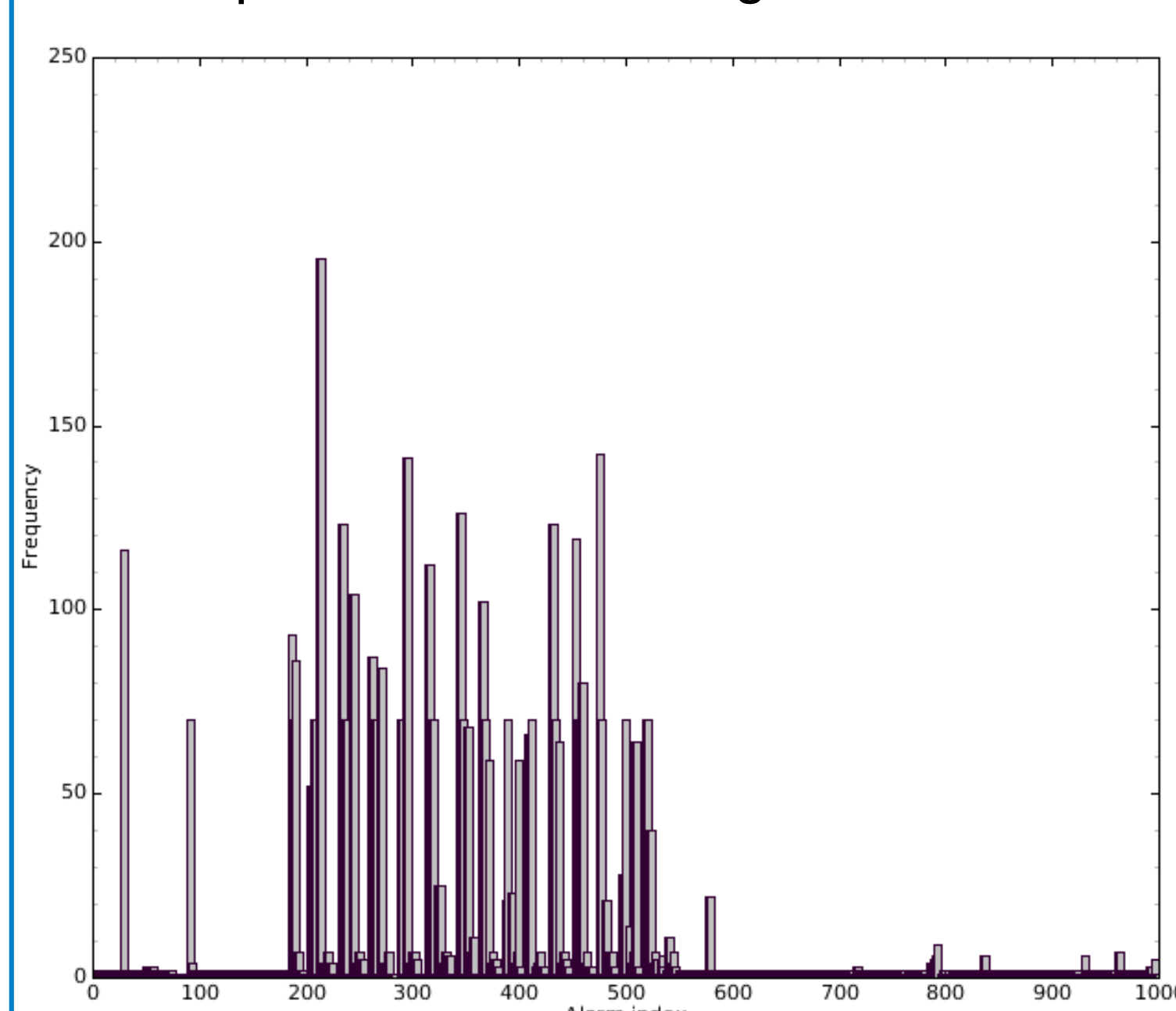
tag=archiving\_hdbb\_mod;formula=(archiving/hdbb++archiver/attributenoknumber > 0);on\_delay=20



## Performance with thousands of alarms

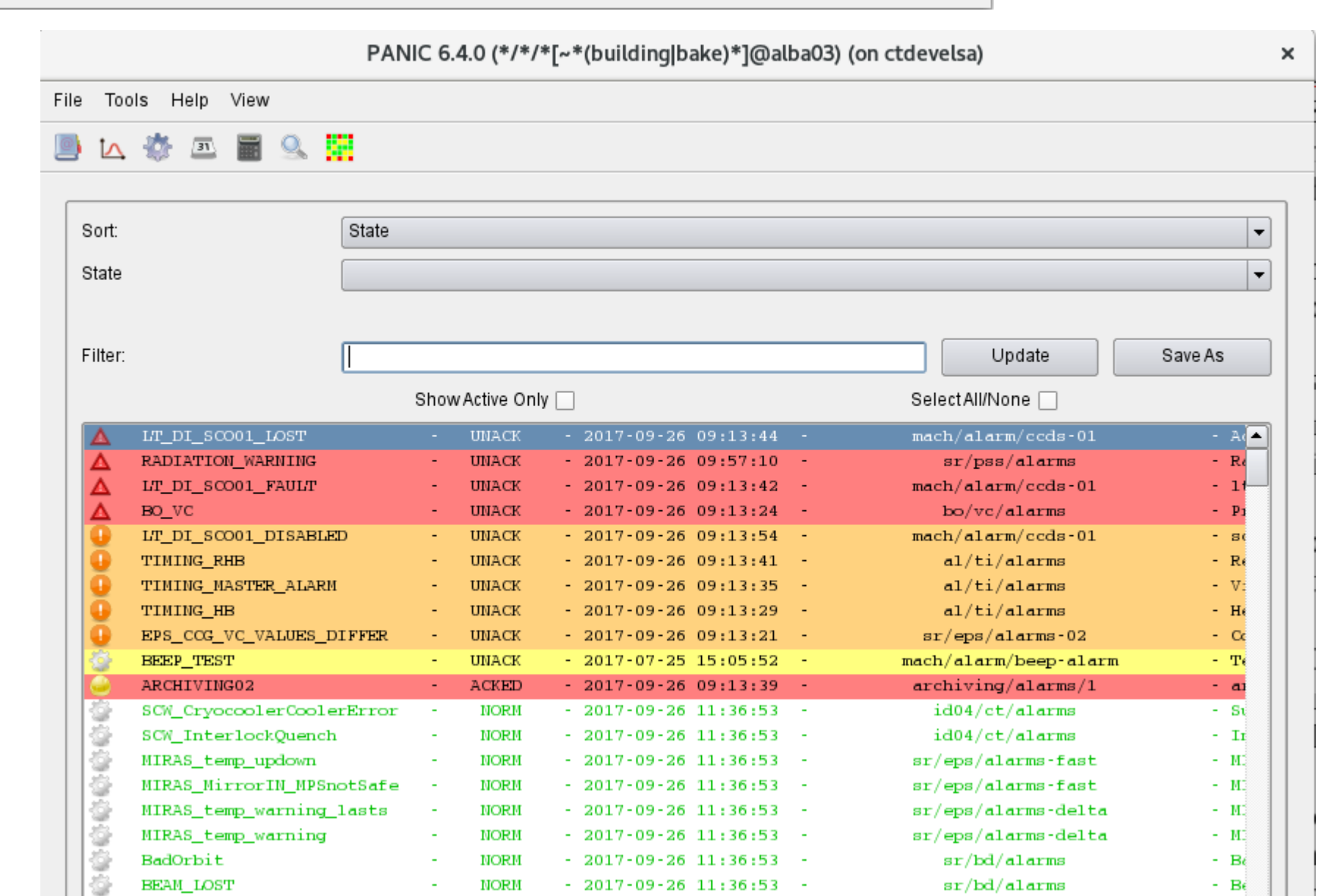
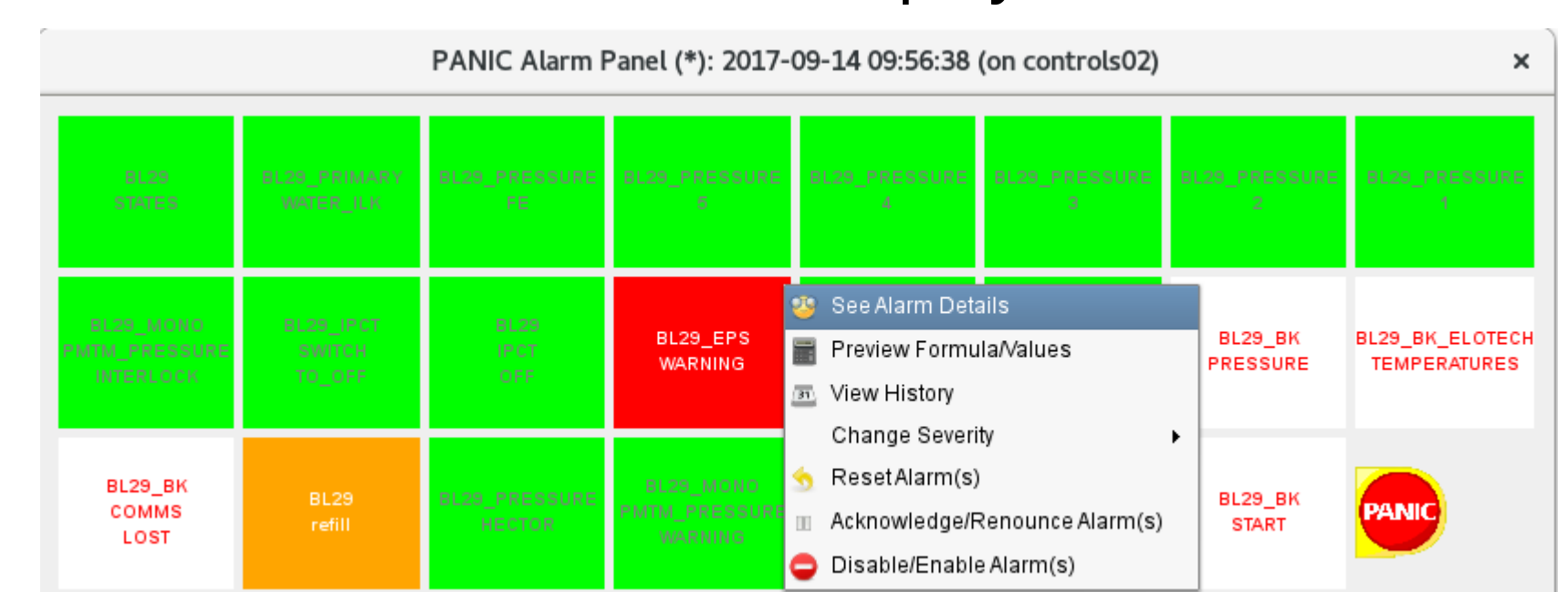
AlarmHandler is able to manage thousands of alarms in one instance. Tested with 1000 alarms evaluated once per minute on average without any significant load on the system.

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.



## Useful links:

<https://github.com/ELETTA-SincrotroneTrieste/alarmhandler>  
<https://github.com/tango-controls/PANIC>  
[www.tango-controls.org](http://www.tango-controls.org)