hp Unified Correlation Analyzer

Unified Correlation Analyzer
for
Event Based Correlation

Version 3.0

Value Pack Examples

Edition: 1.0

June 2013

Legal Notices

Warranty

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

License Requirement and U.S. Government Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

Trademark Notices

Adobe®, Acrobat® and PostScript® are trademarks of Adobe Systems Incorporated.
HP-UX Release 10.20 and later and HP-UX Release 11.00 and later (in both 32 and 64-bit configurations) on all HP 9000 computers are Open Group UNIX 95 branded products.
Java™ is a trademark of Oracle and/or its affiliates.
Microsoft® , Windows® and Windows NT® are U.S. registered trademarks of Microsoft Corporation.
Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.
UNIX® is a registered trademark of The Open Group.
X/Open® is a registered trademark, and the X device is a trademark of X/Open Company Ltd. in the UK and other countries.

Contents

Figures

Tables

Preface

This guide provides some examples of Unified Correlated Analyzer for Event Based Correlation (EBC) Value Packs.

Such examples should be taken as good practice examples for developing new Value Packs.

Product Name: Unified Correlation Analyzer for Event Based Correlation (also referred in this document as UCA for EBC)

Product Version: V3.0

Intended Audience

Here are some recommendations based on possible reader profiles:

- Solution Developers and integrators
- Software Development Engineers

Software Versions

The term UNIX is used as a generic reference to the operating system, unless otherwise specified.

The software versions referred to in this document are as follows:

Product Version	Supported Operating systems
UCA for Event Based Correlation Server Version V3.0	• HP-UX 11.31 for Itanium • Red Hat Enterprise Linux Server release 5.8 & 6.3
UCA for Event Based Channel Adapter V3.0	• HP-UX 11.31 for Itanium • Red Hat Enterprise Linux Server release 5.8 & 6.3
UCA for Event Based Correlation Software Development Kit Version V3.0	• Windows XP / Vista • Windows Server 2007 • Windows 7

Table 1 - Software versions

Typographical Conventions

`Courier` Font:

- Source code and examples of file contents.
- Commands that you enter on the screen.
- Pathnames
- Keyboard key names

Italic Text:

- Filenames, programs and parameters.
- The names of other documents referenced in this manual.

Bold Text:

- To introduce new terms and to emphasize important words.

Associated Documents

The following documents contain useful reference information:

References

[R1] *Unified Correlation Analyzer for Event Based Correlation - Installation Guide*

[R2] *Unified Correlation Analyzer for Event Based Correlation – Administration, Configuration and Troubleshooting Guide*

[R3] *Unified Correlation Analyzer for Event Based Correlation – Reference Guide*

Support

Please visit our HP Software Support Online Web site at www.hp.com/go/hpsoftwaresupport for contact information, and details about HP Software products, services, and support.

The Software support area of the Software Web site includes the following:

- Downloadable documentation,
- Troubleshooting information,
- Patches and updates,
- Problem reporting,
- Training information,
- Support program information.

Chapter 1

Introduction

This guide gives some examples of standard correlation value packs developed for the UCA for Event Based Correlation product.

Throughout this document, we use the *${UCA_EBC_HOME}* environment variable to reference the root directory ("static" part) of UCA for EBC. The default value for the *${UCA_EBC_HOME}* environment variable is */opt/UCA-EBC*. The *${UCA_EBC_HOME}* environment variable thus references the */opt/UCA-EBC* directory unless UCA for EBC "static" part has been installed in an alternate directory.

We also use *${UCA_EBC_DATA}* environment variable to reference the data directory ("variable" part) of UCA for EBC. The default value for the *${UCA_EBC_DATA}* environment variable is */var/opt/UCA-EBC*. The *${UCA_EBC_DATA}* environment variable thus references the */var/opt/UCA-EBC* directory unless UCA for EBC "variable" part has been installed in an alternate directory.

Since UCA-EBC V2.0, the *${UCA_EBC_DATA}* directory may contain multiple instances of UCA-EBC. In this document, we will use the value *${UCA_EBC_INSTANCE}* for referring to *${UCA_EBC_DATA}/instances/<instance-name>* directory. At installation, a single *<instance-name>* is configured: *default*.

Chapter 2

A Low Level Event Filter Value Pack

The Low Level Event Filter Value pack delivers a predefined set of Scenarios that demonstrate event stream processing and provide standard low-level alarm filtering capability.

The following correlation scenarios are provided:

1. **Time Wait**: this low-level filtering scenario discards fault alarms and their associated alarm clearances if they are received during a configurable time window.
2. **Statistical**: this low-level filtering scenario counts all fault alarms received during the configurable time window. If the total number of faults reaches a configurable threshold, then a statistic alarm is generated.
3. **Grouping**: this low-level filtering scenario is similar to the “Time Wait” scenario, but adds root cause correlation functionality by grouping alarms.
4. **Inactivity**: this low-level filtering scenario identifies when a specific technology stops forwarding raw messages.
5. **Up/Down**: this low-level filtering scenario is a complementary scenario, created for the “Time Wait” and “Grouping” scenarios. It handles alarms that have a common clearance alarm.

In the Low Level Event Filter Value Pack, the input Alarms flow is dispatched to the different scenarios according to the following figure:

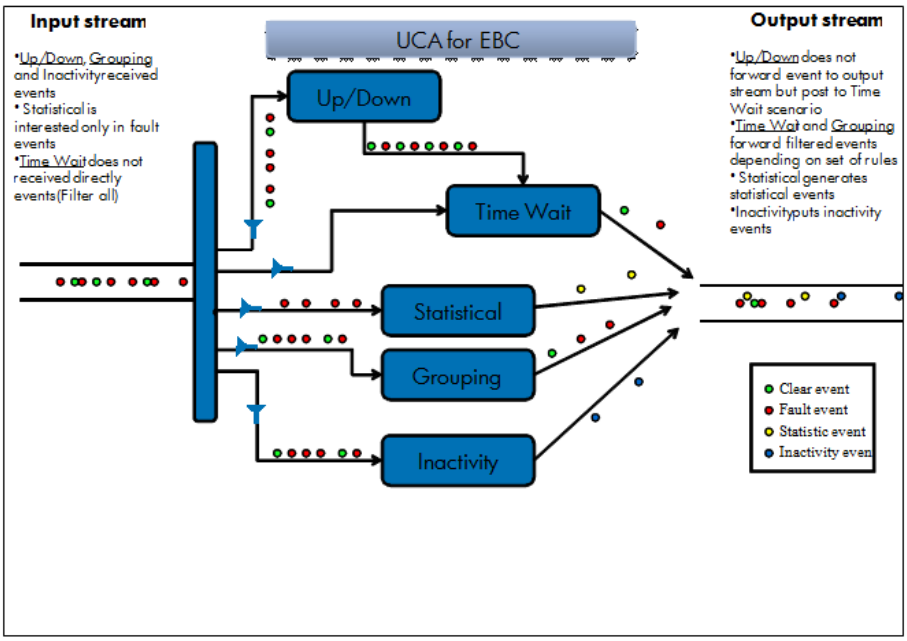

Figure 1 - Alarm flow in Low Level Event Filtering Value Pack

The following assumptions have been used:

- A cleared alarm is an alarm that has the "Perceived Severity" attribute equal to "CLEAR"
- Parent (Root Cause) alarms are identified by the "AddText" attribute equal to "Root Cause ..."
- Child alarms are identified by the "AddText" attribute equal to "Symptom ..."
- Common clear alarms (that need to 'clears' several faults) are identified by the "AddText" attribute equal to "Common clearance ..."

2.1 Software Prerequisites

The Low Level Event Filtering Value Pack is installed on top of the UCA for EBC product.

2.2 Deploying the Low Level Event Filtering Value Pack

Several steps are needed to deploy an EVP (UCA for EBC Value Pack):

1. Install the EVP package in the *${UCA_EBC_INSTANCE}/valuepacks* directory (*${UCA_EBC_INSTANCE}* translates to */var/opt/UCA-EBC/instances/<instance name>* by default unless UCA for EBC was installed at an alternate location)
2. Deploy the Value Pack
3. Start the Value Pack.

2.2.1 Installing the Value Pack

The Low Level Event Filtering EVP package example is installed with the UCA for EBC Server kit. The Low Level Event Filtering EVP is located in the *${UCA_EBC_HOME}/defaults/valuepacks* directory. You will need to copy the Low Level Event Filtering Value Pack zip file (named *llef-example-vp-3.0.zip*) to the *${UCA_EBC_DATA}/valuepacks* directory, so that it can be seen by UCA for EBC.

Alternatively, if you have installed the UCA for EBC Development Toolkit, you can (modify and) re-build the Low Level Event Filtering EVP from the source code by executing the following commands:

On Windows:

```
$ cd %UCA_EBC_DEV_HOME%\llef-example
$ ant all
```

On Linux:

```
$ cd ${UCA_EBC_DEV_HOME}\llef-example
$ ant all
```

When rebuilt, the package is ready to be deployed on UCA for EBC. You just need to copy the Value Pack package you have just generated to the *${UCA_EBC_INSTANCE}/valuepacks* directory.

2.2.2 Deploying the Value Pack

To deploy the Low Level Event Filtering Value Pack, please use the "--deploy" option of the **uca-ebc-admin** command-line administration tool (executed as **'uca'** user):

On both HP-UX and Linux:

```
$ cd ${UCA_EBC_HOME}/bin
$ uca-ebc-admin --deploy -vpn llef-example -vpv 3.0
```

An output similar to the following will be displayed:

```
UCA for EBC Home directory set to: /opt/UCA-EBC
UCA for EBC Data directory set to: /var/opt/UCA-EBC
INFO  - Value Pack name: llef-example version: 3.0 has been
successfully deployed
INFO  - Exiting...
```

Or simply deploy the Value Pack from the UCA for EBC User Interface.

2.2.2.1 File organization

At the end of the deployment step, the files delivered by the Value Pack are deployed in *${UCA_EBC_INSTANCE}/deploy/llef-example-3.0* directory, according to the following file structure:

Directories	Description

lib/	Some additional jar files are installed for this package
conf/	Configuration files that defines the Value Pack, and the scenarios
grouping/	Directory containing all files defining **Grouping** scenario
inactivity/	Directory containing all files defining **Inactivity** scenario
statistical/	Directory containing all files defining **Statistical** scenario
timewait/	Directory containing all files defining **Time Wait** scenario
updown/	Directory containing all files defining **Up/Down** scenario

Table 2 - File structure of the LLEF value pack

2.2.3 Starting the Value Pack

Value Packs can be started in two different manners depending on whether UCA for EBC is already started or not.

If UCA for EBC is stopped, restarting the application will automatically start all Value Packs deployed in the *${UCA_EBC_INSTANCE}/deploy* directory.

If UCA for EBC is already running, use the "--start" option of the **uca-ebc-admin** command-line administration tool (executed as **'uca'** user) to start the Value Pack:

On both HP-UX and Linux:

```
$ cd ${UCA_EBC_HOME}/bin
$ uca-ebc-admin --start -vpn llef-example -vpv 3.0
```

An output similar to the following will be displayed:

```
UCA for EBC Home directory set to: /opt/UCA-EBC
UCA for EBC Data directory set to: /var/opt/UCA-EBC
INFO  - Exiting...
```

Or simply start it from the UCA for EBC Web User Interface.

2.3 Stopping the Low Level Event Filtering Value Pack

You can stop the Value Pack when UCA for EBC is running using the "--stop" option of the **uca-ebc-admin** command-line administration tool (executed as **'uca'** user):

On both HP-UX and Linux:

```
$ cd ${UCA_EBC_HOME}/bin
$ uca-ebc-admin --stop -vpn llef-example -vpv 3.0
```

An output similar to the following will be displayed:

```
UCA for EBC Home directory set to: /opt/UCA-EBC
UCA for EBC Data directory set to: /var/opt/UCA-EBC
INFO  - Exiting...
```

Or simply stop it from the UCA for EBC Web User Interface.

2.4 Undeploying the Low Level Event Filtering Value Pack

To undeploy the Low Level Event Filtering Value Pack, use the "--undeploy" option of the **uca-ebc-admin** command-line administration tool (executed as **'uca'** user):

On both HP-UX and Linux:

```
$ cd ${UCA_EBC_HOME}/bin
$ uca-ebc-admin --undeploy -vpn llef-example -vpv 3.0
```

An output similar to the following will be displayed:

```
UCA for EBC Home directory set to: /opt/UCA-EBC
UCA for EBC Data directory set to: /var/opt/UCA-EBC
INFO  - Value Pack name: llef
example version: 3.0 has been successfully undeployed
INFO  - Exiting...
```

Or simply undeploy it from the UCA for EBC Web User Interface.

2.5 Low Level Event Filtering Value Pack Scenarios

2.5.1 The Time Wait scenario

2.5.1.1 Functional description

The Time Wait scenario receives fault alarms and waits during a configurable time period for a clear alarm to arrive. If a clear alarm is received during the time period both alarm and alarm clearance are discarded and none of them are forwarded to the Trouble Ticket (TT) generator external application. The purpose of this scenario is to avoid unreasonable trouble ticket to be opened due to unusual situations, such as a temporary system overload or an equipment restart.

The main rules in the Time Wait scenario are:

- Discard both fault alarm and fault alarm clearance when both are received during the time wait period
- Keep fault alarms (or fault alarm clearance) when no corresponding fault alarm clearance (or fault alarm) is received during the time wait period

The following sections describe the possible use cases in detail.

Note

The rules are parameterized. You can set the value for the following parameters in the *timewait-params.xml* file:

- **timewait**: duration of the time wait period (the default value is 2 seconds)

2.5.1.2 Both Fault and Clearance are discarded

This use case applies when both the alarm and the associated alarm clearance are received during the time wait period:

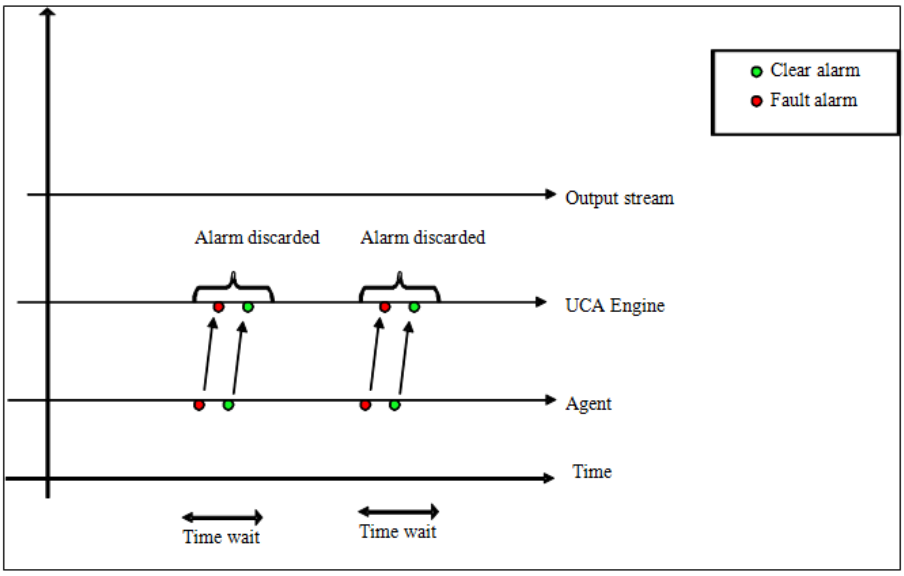

Figure 2 - Time Wait – Both Fault and Clearance are discarded

2.5.1.3 Fault or Clearance is kept

This use case applies when an alarm is not cleared during the time wait period or when a clearance is received without the corresponding alarm during the time wait period:

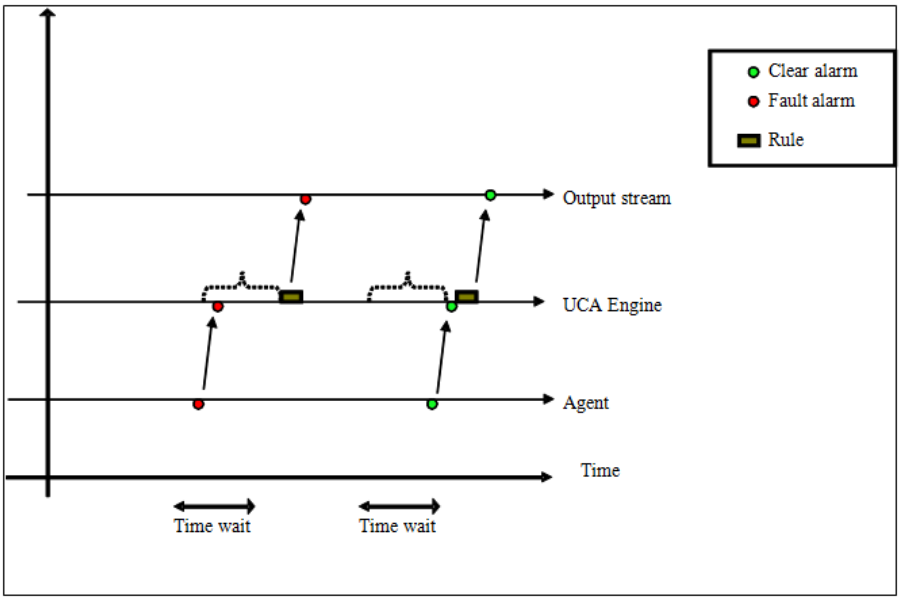

Figure 3 - Time Wait – Fault or Clearance is kept

2.5.2 The Statistical scenario

2.5.2.1 Functional description

The Statistical scenario counts all fault alarms received during a configurable period of time. If the total number of alarms reaches a configurable threshold, then a statistical alarm is generated. If not, then no statistical alarm is generated. In any case the original alarms are discarded.

The main rules in the Statistical scenario are:

- Create a Statistical alarm whenever a configurable number of alarms (alarm clearances excepted) is received during a configurable time window
- Don't create any Statistical alarm if the number of alarms (alarm clearances excepted) received during a configurable time window does not reach a configurable number of alarms (threshold)
- Discard all original alarms

The following sections describe the possible use cases in detail.

Note

The rules are parameterized. You can set the value for the following parameters in the `statistical-params.xml` file:

- **threshold**: number of alarms required before sending a Statistical output alarm (the default value is 3 alarms)
- **timewindow**: duration of the time window period during which alarms are counted (the default value is 5 seconds)

2.5.2.2 Number of faults is above the threshold

This use case applies when the number of alarms received during the configurable *timewindow* is higher than the *threshold* parameter:

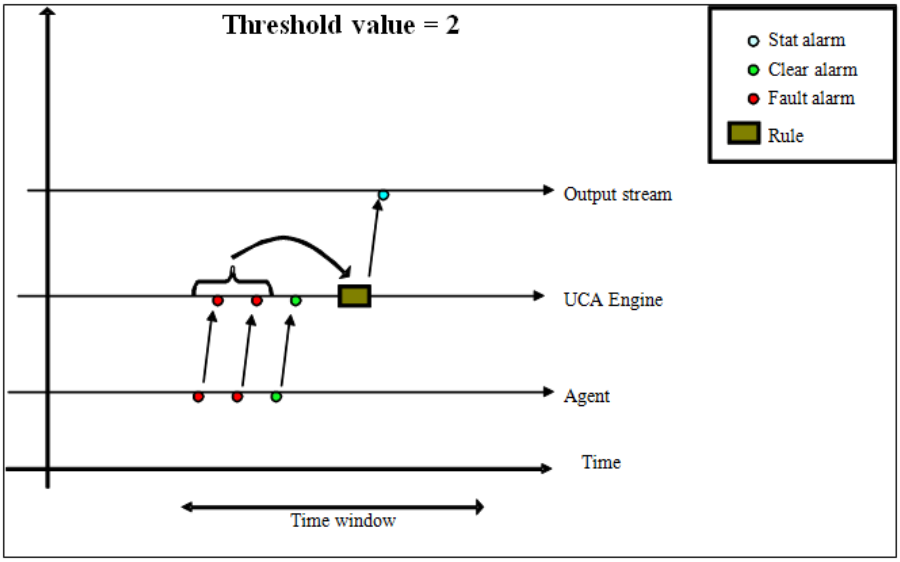

Figure 4 - Statistical – Number of faults is above the threshold

2.5.2.3 Number of faults is below the threshold

This use case applies when the number of alarms received during the configurable *timewindow* is lower than the *threshold* parameter:

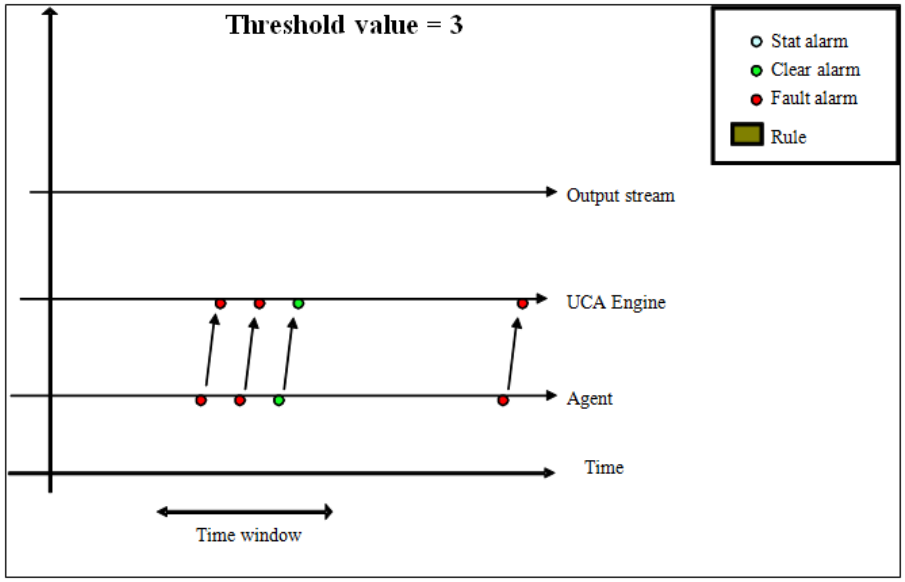

Figure 5 - Statistical – Number of faults is below the threshold

2.5.2.4 Number of faults is several times the threshold

This use case applies when the number of alarms received during the configurable *timewindow* is equal to several times the value of the *threshold* parameter:

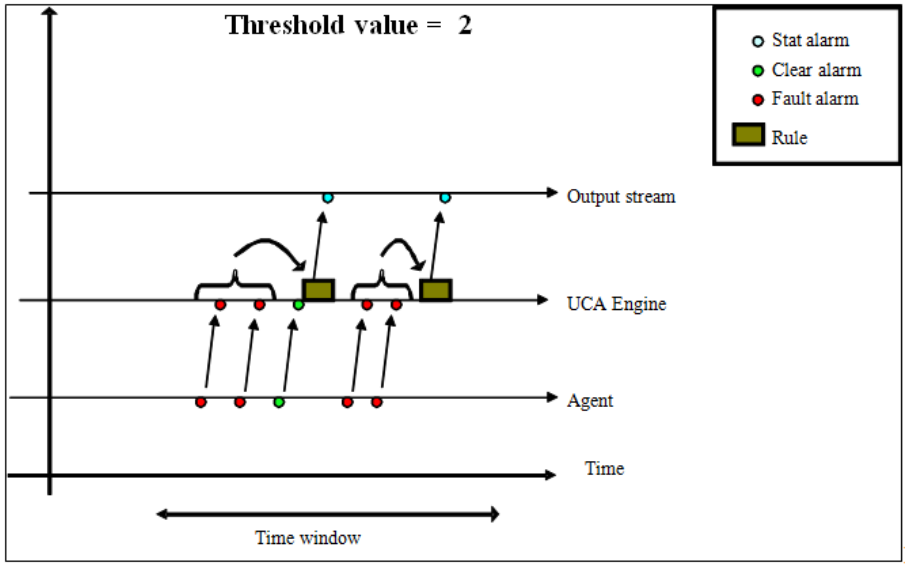

Figure 6 - Statistical – Number of faults is several times the threshold

2.5.3 The Grouping scenario

2.5.3.1 Functional description

Besides discarding fault alarms that are cleared during a configurable period of time, the Grouping scenario groups fault alarms that share another alarm as a common root cause. The root cause alarm is called “parent” and symptom alarms are called “children”.

The scenario works during a configurable period of time, starting at the receipt of a root cause alarm. During this period, if a “child” alarms is received, the ‘parent’ alarm is enriched with information from the “child” alarm. The “child” alarm is discarded and the “parent” alarm is forwarded.

If no “child” alarm is received, the “parent” alarm is forwarded unchanged. If an alarm clearance is received during the configurable time period, only the cleared alarm is discarded. If the “parent” alarm is cleared, it is discarded and all not-cleared “child” alarms are forwarded.

The following sections describe the possible use cases in detail.

Note

The rules are parameterized. You can set the value for the following parameters in the `grouping-params.xml` file:

- **timeslot**: duration of the time window period (the default value is 10 seconds)

2.5.3.2 No alarm clearance received during the time window

This use case applies when the root cause alarm is followed by child alarms:

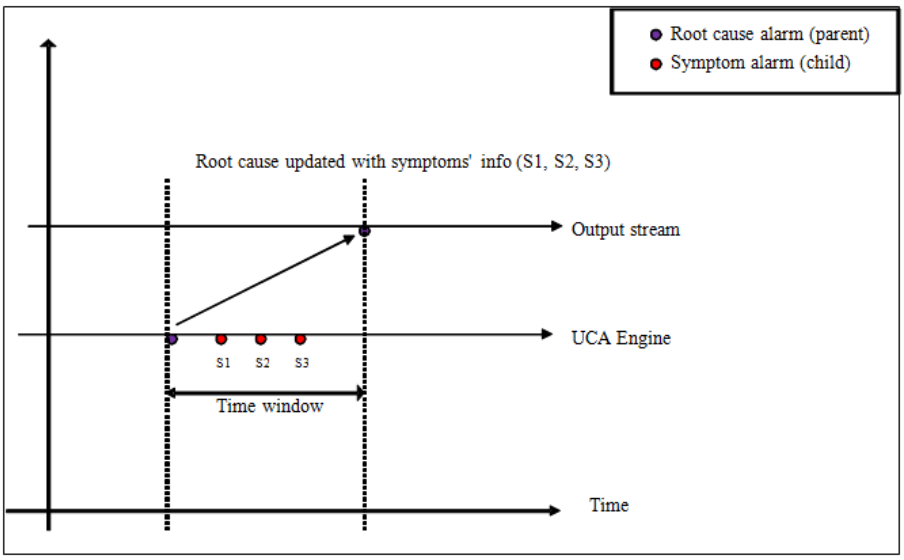

Figure 7 - Grouping – No alarm clearance received during the time window

2.5.3.3 No symptom alarm received during the time window

This use case applies when the root cause alarm is not followed by child alarms:

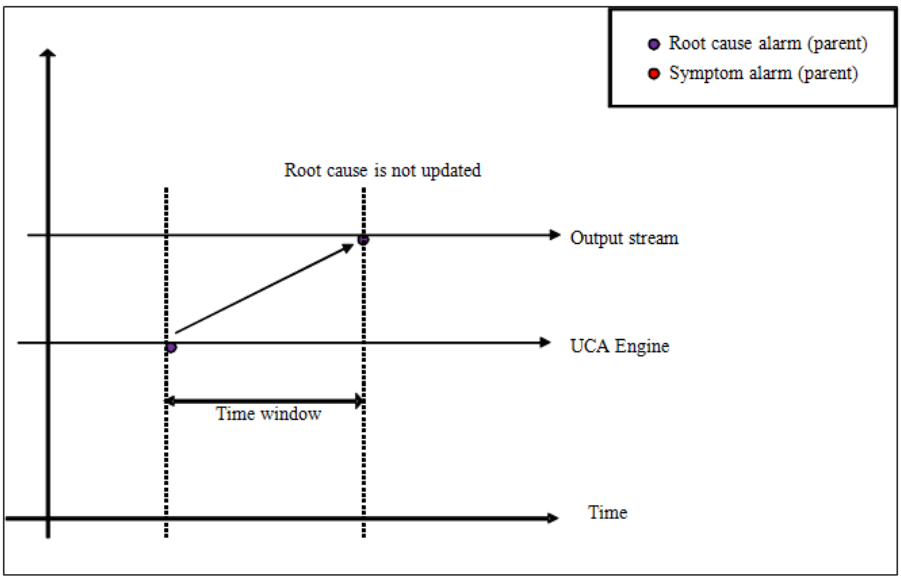

Figure 8 - Grouping – No symptom alarm received during the time window

2.5.3.4 Clearance on symptom alarms received during the time window

This context applies when the root cause alarm is followed by child alarms and child clearance.

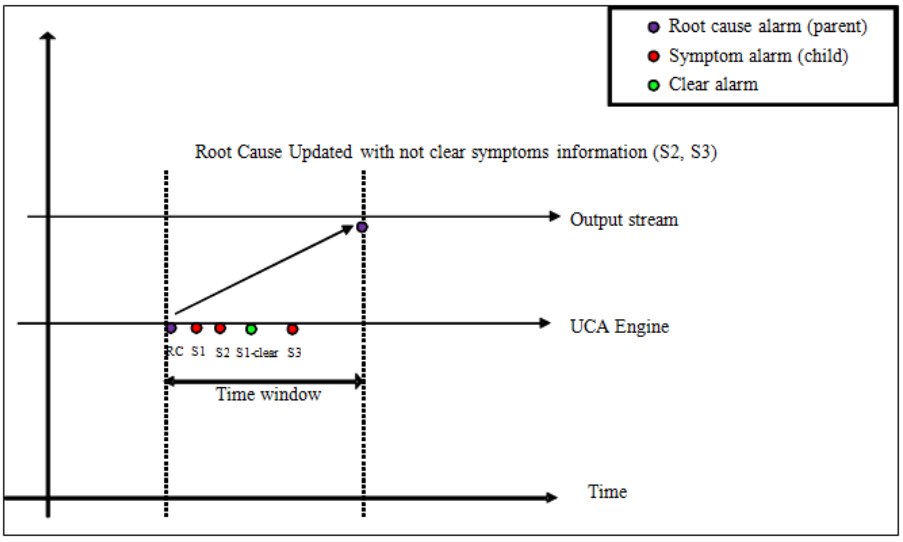

Figure 9 - Grouping – Clearance on symptom alarms received during the time window

2.5.3.5 Clearance on root cause and symptom alarms received during the time window

This use case applies when the root cause alarm is followed by the root alarm clearance as well as child alarms and clearances:

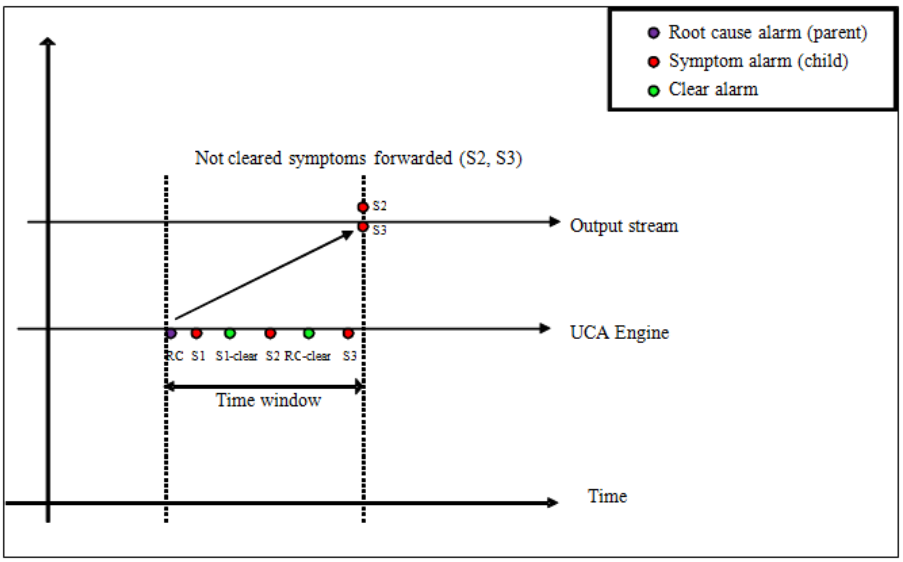

Figure 10 - Grouping – Clearance on root cause and symptom alarms received during the time window

2.5.4 The Inactivity scenario

2.5.4.1 Functional description

The Inactivity scenario identifies when a specific technology stops forwarding raw messages. The Inactivity scenario receives all alarms for the specific technology.

For each valid alarm received, the scenario resets a counter associated with the monitored item. If no valid alarm is received during a configurable period of time, an "Inactivity" alarm is created and injected into the Alarm Management system, so it can be enriched and moved forward.

The main rules in the Inactivity scenario are:

- Generate an "Inactivity" alarm if no alarm is received during a configurable time period.
- Normal alarms are discarded

The following sections describe the possible use cases in detail.

Note

The rules are parameterized. You can set the value for the following parameters in the `inactivity-params.xml` file:

- **inactivityTime**: duration of the time window period (the default value is 5 seconds)

2.5.4.2 Inactivity detected

This uses case applies when no recurrent alarm is received during the configurable time window:

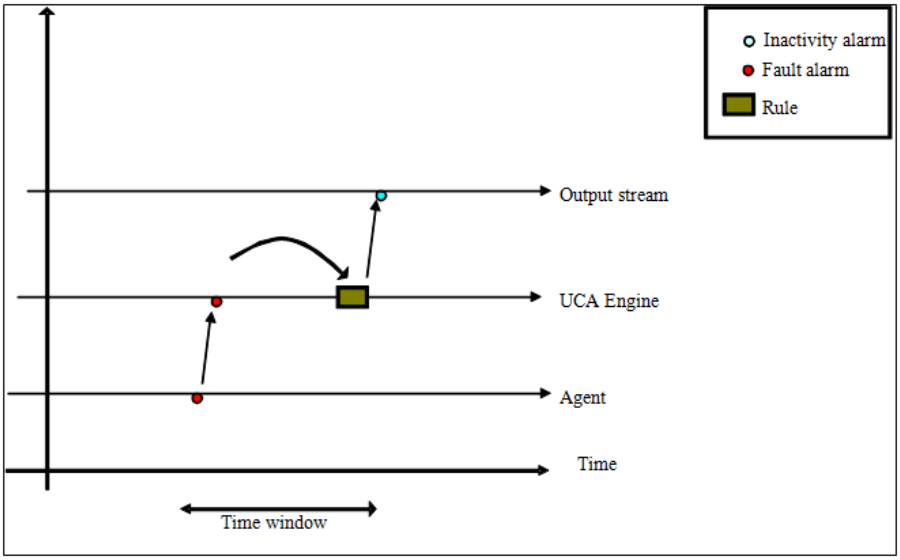

Figure 11 - Inactivity– Inactivity detected

2.5.4.3 Inactivity not detected

This use case applies when recurrent alarms are received during the configurable time window:

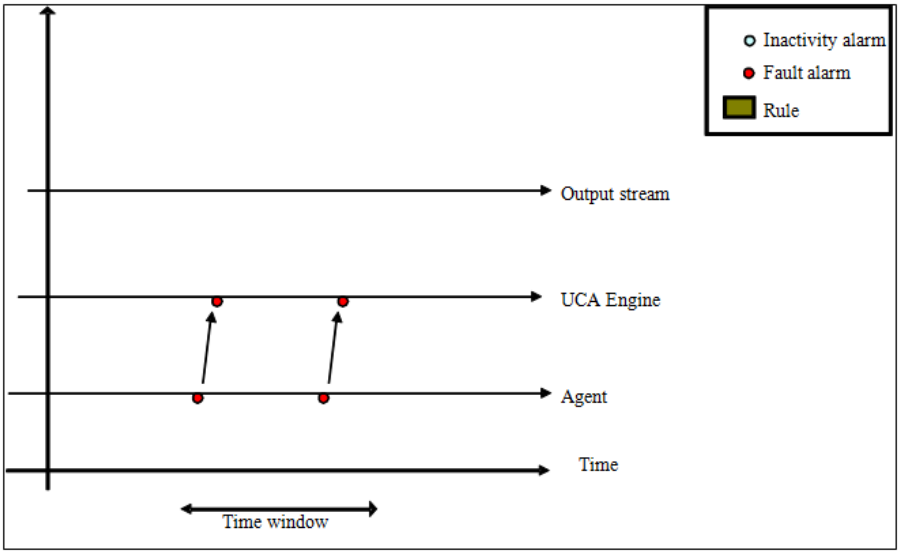

Figure 12 - Inactivity – Inactivity not detected

2.5.4.4 Inactivity detection in mix technology context

This use case applies when one technology is active and another one is not:

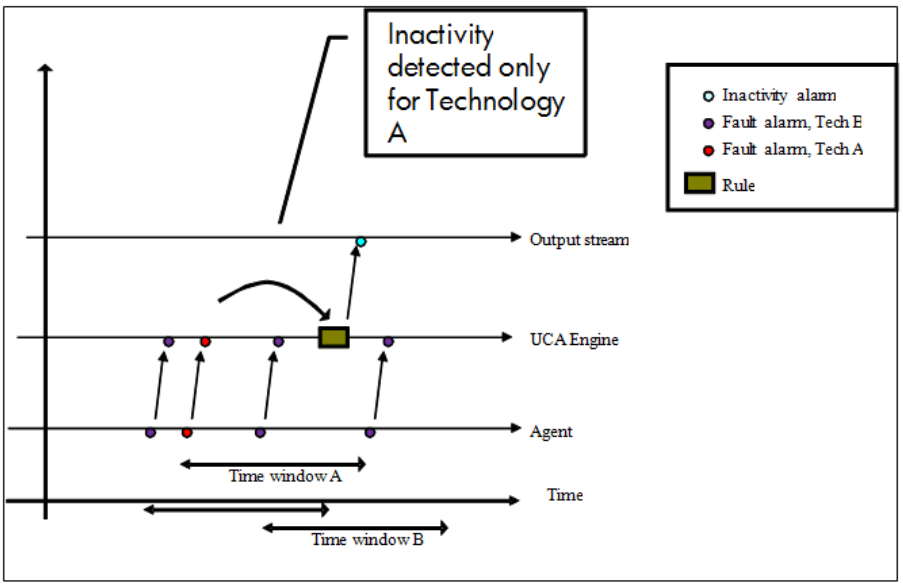

Figure 13 - Inactivity – Inactivity detection in mix technology context

2.5.5 The Up/Down scenario

2.5.5.1 Functional description

The Up/Down scenario is a base scenario, used by both the Time Wait and Grouping scenarios. The purpose is to increase performance. It handles alarms that have a common clearance alarm.

All alarms that share the same clearance enter this scenario. When one of such alarms is received, the scenario keeps a copy of it. When the common clearance alarm is received, an independent clearance alarm is generated for each alarm stored by the scenario.

The following sections describe the possible use cases in detail.

2.5.5.2 Independent clearance

In this use case, the Up/Down scenario stores incoming alarms until it receives a common clearance alarm, in which case an independent clearance alarm is created for each alarm stored by the scenario:

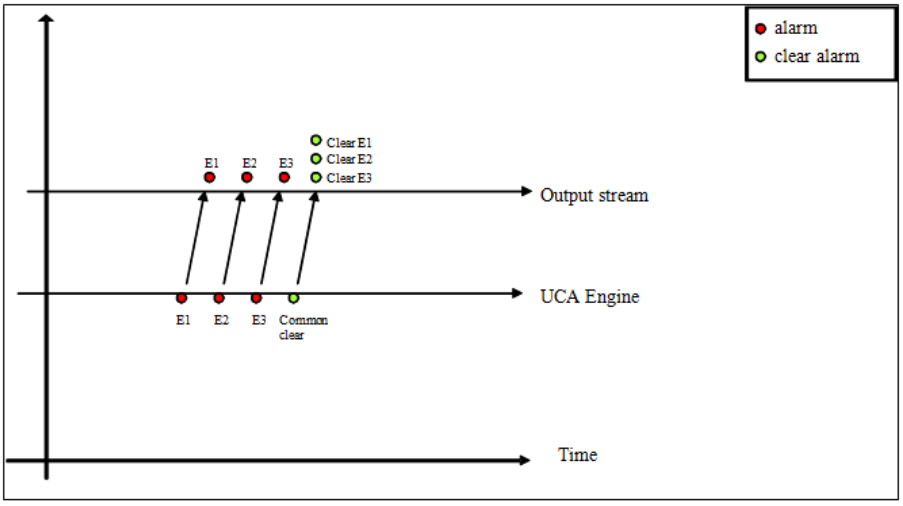

Figure 14 - Up/Down – Independent clearance

2.5.5.3 Independent clearance cascaded with Time Wait scenario

In this use case the Up/Down scenario feeds (cascades) into the Time Wait scenario:

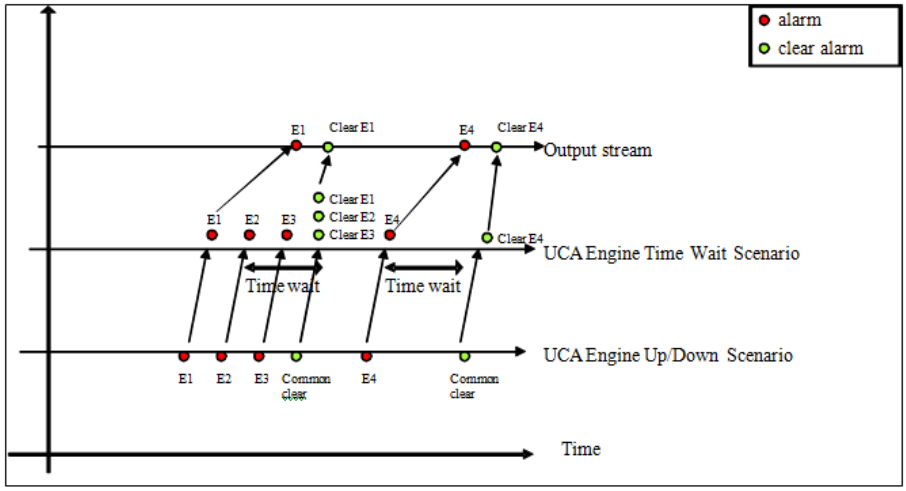

Figure 15 - Up/Down – Independent clearance cascaded with Time Wait scenario

2.6 Testing the Low Level Event Filtering Value Pack

For all the LLEF example scenarios described earlier, some alarm sets are delivered in order to test the scenario behavior.

These alarm samples are present in the *${UCA_EBC_INSTANCE}/deploy/llef-example-3.0/<scenario name>/Alarms.xml* files after the LLEF Value Pack as been deployed.

The Alarm.xml files can be injected into UCA for EBC by using the **uca-ebc-injector** command-line tool as follows:

On both HP-UX and Linux:

```
$ ${UCA_EBC_HOME}/bin/uca-ebc-injector -file Alarms.xml
```

Checking the scenario result:

Rules actions of the LLEF example are designed to simulate real alarm actions by logging the actions into a log file:
${UCA_EBC_INSTANCE}/logs/llef_example.log.

Playing the Up/Down scenario with the Up/Down alarm sample file (*${UCA_EBC_INSTANCE}/deploy/llef-example-3.0/updown/Alarms.xml*) generated the following output in the *${UCA_EBC_INSTANCE}/logs/llef_example.log* log file:

```
2012-01-04 17:22:44.982:
com.hp.uca.expert.vp.llef.timewait.TimeWait: Dummy Action
processed on alarm: 8036aa86-046e-4cbc-9dd4-ba67ff528452
2012-01-04 17:22:46.995:
com.hp.uca.expert.vp.llef.timewait.TimeWait: Dummy Action
processed on alarm: 12303
2012-01-04 17:22:46.995:
com.hp.uca.expert.vp.llef.timewait.TimeWait: Dummy Action
processed on alarm: 12307
```

You will notice that according to the Up/Down scenario design, alarms have indeed been forwarded to the Time Wait scenario which performed the actions.

Glossary

EVP: UCA for EBC Value Pack

GUI: Graphical User Interface

JMS: Java Messaging Service

JMX: Java Management Extension, used to access or process action on the UCA for EBC product.

JNDI: Java Naming and Directory Interface

Inference engine: Process that uses a Rete algorithm

DRL: Drools Rule file

NMS: Network Management System

SDK: Software Development Kit

TT: Trouble Ticket

XML: Extensible Markup Language

XSD: Schema of an XML file, describing its structure. XSD stands for XML Schema Definition

X733: Norma describing the structure of an Alarm used in telecommunication environment.