ESKORT Designer for Clearance

User Guide, Section III - Knowledge Engineering

DOCUMENT

Title:	User Guide, Section III - Knowledge Engineering
Document:	icdk/Application/Designer/UG/3
Date:	2010.05.05
Version:	1.1
Author:	Marco Dijkstra, Jørgen Rune Mortensen
Contributions by:	
Classification:	
Distribution:	

Versions:	1.0	Intiial Version
	1.1	Footer changed to Intracom
Printed:	16.05.12	

Table of Contents

°LastInfoPage

1. Overview

This document is aimed at establishing a basic understanding of the knowledge engineering process as well as introducing the related concepts and terminology.

1.1 Conventions

Definitions – The first time a specific term or concept is mentioned in a relevant subsection it is printed in italic. If required, further definition will follow shortly after the first time it is mentioned and is preceded by an icon.

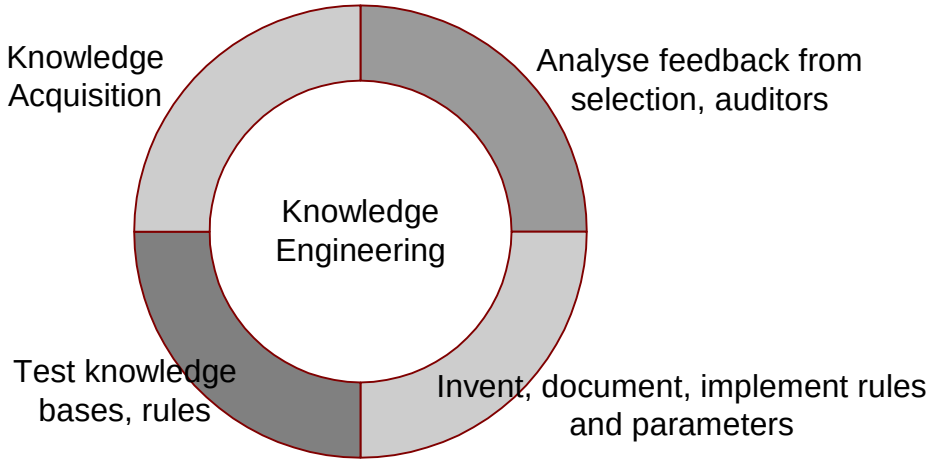
Exercise – Exercises are printed in bold and are preceded by an icon. 

[Menu] – Menu options, buttons, window and field names are printed within square brackets. The word option, button, field, dialog/window behind the word will specify the type.

2. Knowledge Engineering

2.1 Introduction

The knowledge engineering process covers the work required to establish and maintain the *knowledge base(s)* used in the risk analysis process to generate *rules.* It is a highly iterative process.



Knowledge engineering will primarily be carried out by a number of specially trained knowledge engineers. Other people could be involved in providing input to the process - in the form of inspiration and feedback - and could participate in varying degrees in the actual knowledge engineering.

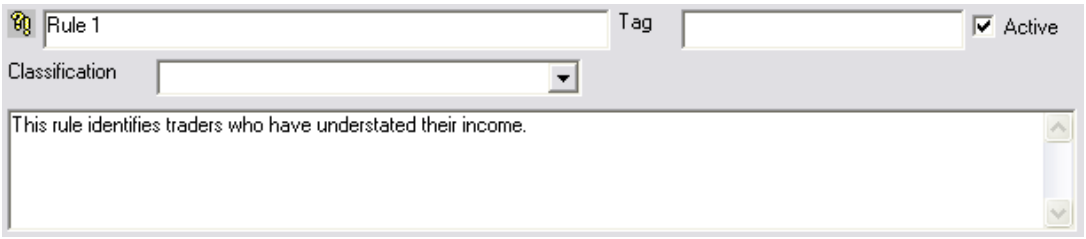
Most aspects of knowledge engineering are accomplished using the ESKORT Designer tool. For an introduction to the Designer, refer to [ESKORT Designer, Section I - Introduction].

2.2 Defining, documenting and implementing rules

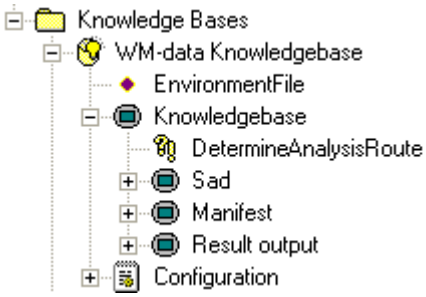
2.2.1 Overview

The selection knowledge used for the automated risk analysis is documented and formalized in a knowledge base.

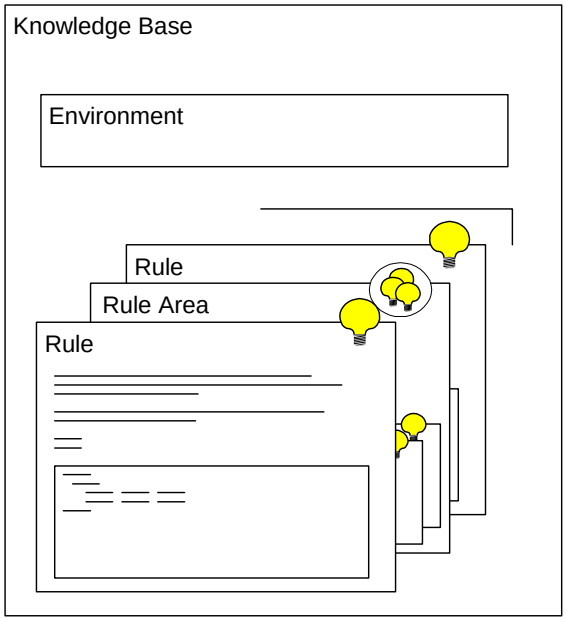
The basic building blocks of the knowledge base, are rules - each defining, documenting and implementing (through a *rule formalization*) a single 'element of knowledge'. The rule is typically a simple implication - a specific characteristic within the declarationdata that e.g. implies or indicates a discrepancy or possible error from a deficiency perspective.



The knowledge base can be structured into a number of *areas*, each comprising one or more rules (or other rule areas). This allows aspects common to a group of rules to be specified once rather than repeated for each rule. Also, it may be used to reflect dependency between rules.

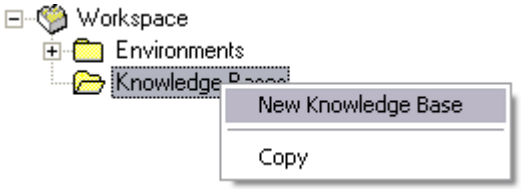


In order to work with a knowledge base, it is necessary to define the data model for the data used in the rule formalizations. This information - and other information about the context in which the knowledge base is being used - is referred to as the *environment* of the knowledge base.



2.2.2 Inserting a Knowledge Base

You can insert a new knowledge base by selecting the 'Knowledge Bases' folder and selecting the [New Knowledge Base] option in the context menu



This will add a new (empty) knowledge base to the configuration archive.

The next step is to give the new knowledge base a name and a description and to specify in what context the knowledge base will be developed. This can be achieved by making the new knowledge base the current element and enter the information in the editor pane of the Designer.

Each knowledge base has the following attributes:

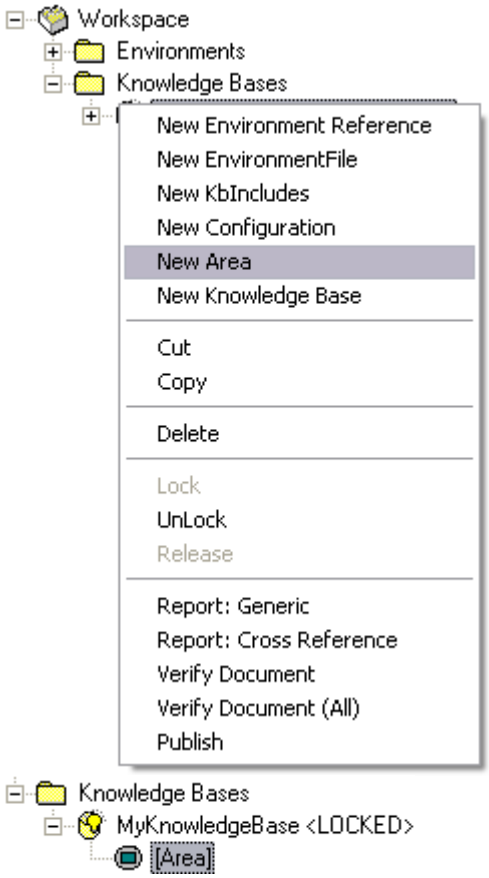
- Name – name of the knowledge base.
- Description – description of the knowledge base.

2.2.3 Inserting KbInclude Elements

2.2.4

2.2.5 Inserting Areas

Each knowledge base needs at least one top area.



In the editor pane, give this area a name and a description.

The Tag field is an attribute that is stored in the knowledge base along with the area. The Tag can be used by the ESKORT Clearance Risk Analysis System to recognize and process specific areas in the knowledgebase. E.g. the Configuration Provider in the ESKORT Clearance Risk Analysis System merges Trap Rules[1] (as defined using Clearance Workbench) into the knowledgebase by searching for the area tagged by "[TRAP-RULES]".

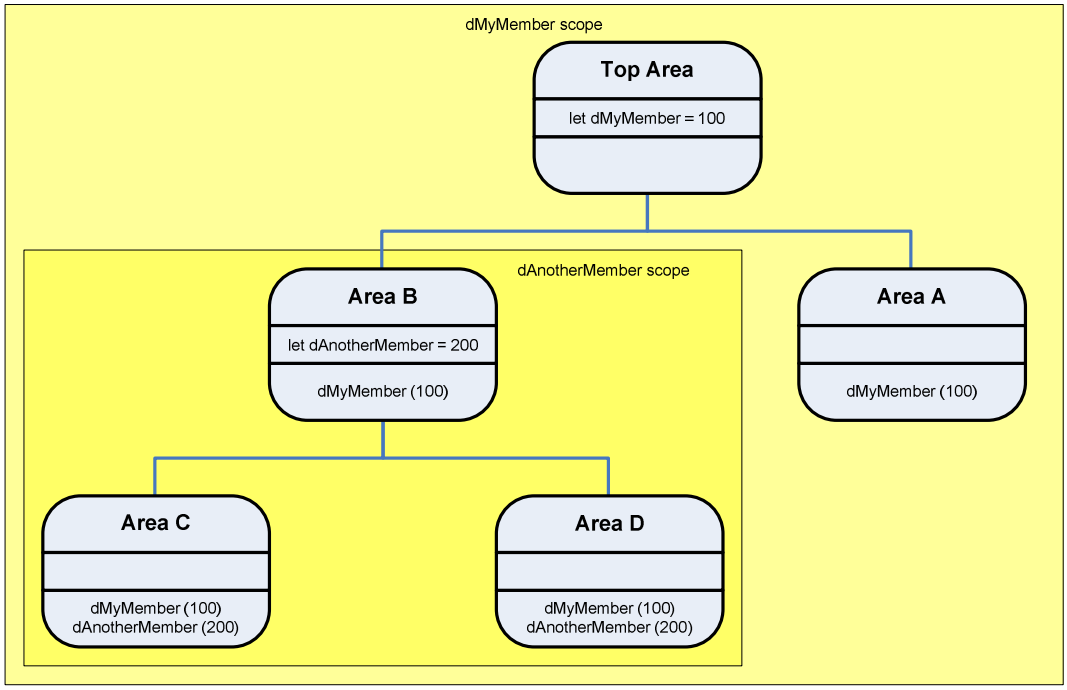
2.2.6 Area Formalization

You may notice that the editor pane also has an attribute allowing you to enter formalization.

[1] In some implementations Trap Rules are known as Risk Profiles.

As explained before, areas allow aspects common to a group of rules to be specified once rather than repeated for each rule. Also, it may be used to reflect dependency between rules.

Most typically you would define so-called *derived members* in the formalization of an area. Once derived members have been defined in an area, all child areas and child rules will be able to access them. See also the illustration below:

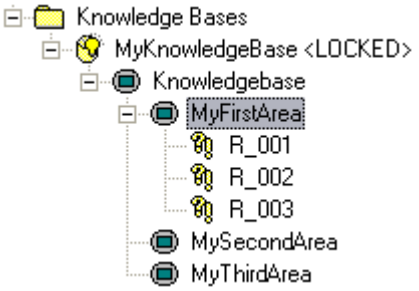


Because *dMyMember* has been defined in the top area, it is accessible in all child areas. This does not count for *dAnotherMember*, which is only accessible in the child areas 'C' and 'D'.

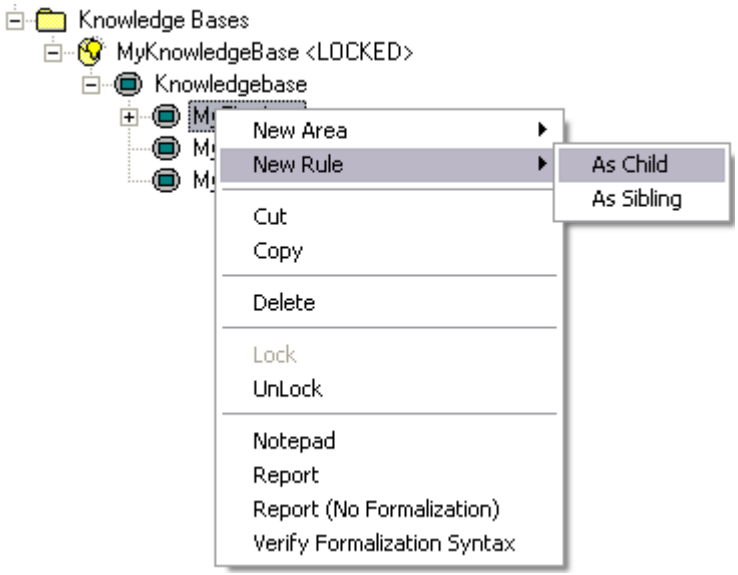
For more information about derived members see chapter 3.

2.2.7 Inserting Rules

Rules are the basic building blocks of knowledge bases. They can appear anywhere in the hierarchy.



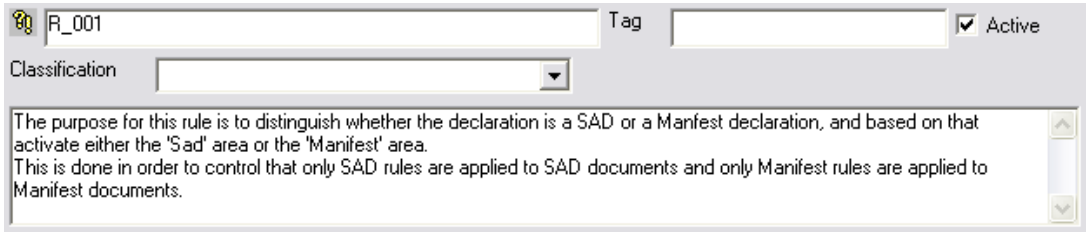
To insert a rule, select the area of which the rule should be a child element and choose [New Rule] | [As Child] from the context menu.



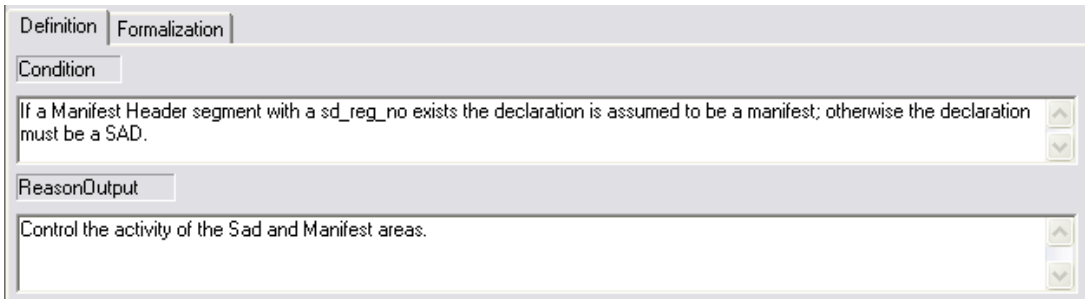
A new rule will now be inserted.

2.2.8 Documenting Rules

The next step would be to give the rule a name and to enter a description.



Most typically, one would now define and document the rule using the [Definition] tab in the editor pane.



Note that the rule attributes shown above may be different in your implementation.

During the risk analysis process declaration data will be evaluated using the rules in the knowledge base. Rules for which the condition is met are said to *fire*. This leads to the creation of an *observation*, where each observation reflects a given observation made about the declaration item and is given some measure of importance (e.g. in the form of scores or a green or red routing).

An observation is a finding, which is typically composed of a measure indicating the significance of the finding and a textual explanation.

2.2.9 Formalizing Rules

The formalization ([Formalization] tab in editor pane) is used for the formalized representation of a rule's implementation (if ... then...). The formal rules are expressed in a *formalization language* (MDC), which allows them to be automatically evaluated during the analysis process.

For more information about the formalization language, refer to [ESKORT Designer, Section IV – Formalization Language].

2.2.10 Active Attribute

For both the areas and rules there is an attribute called Active:

This specifies whether or not an area or rule initially is active.

If an area or rule is active it will be evaluated automatically in an analysis – this is the default and most common behavior.

If an area or rule is not active it will **not** be evaluated automatically, and it will only be evaluated if another rule activates it. It is possible for rules to activate other rules and areas. This is done in the action part of the formalization; refer to [ESKORT Designer, Section IV – Formalization Language] for more information about this.

2.3 Generic Reports

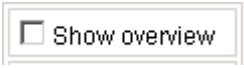
The generic reports allow you to choose what to show after you have opened the report.

2.3.1 Generic Report for Knowledge Bases

The generic report for knowledge bases allows information on the rules in a knowledge base to be reported. The report is activated through [Report: Generic] on the context menu for knowledge bases.

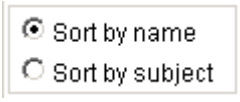
2.3.1.1 Overview

You can start the report with an overview of all the rules included. The overview is included when you have checked the [Show overview] checkbox.



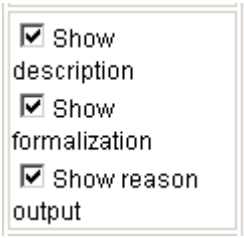
2.3.1.2 Sorting

You can choose how the rules should be sorted, by selection either [Sort by name] or [Sort by subject].



2.3.1.3 Selecting Information

You can select the information you want to include in the report by checking one or more of the following check boxes:



2.3.1.4 Filtering

You can filter the report (i.e. only show information for selected rules), by using one or more of the filters on the right. The check box indicates whether the filter is active, while the listbox to the left shows all possible values (e.g. all rule status values).

When a filter checkbox is checked, the report will only show information on rules with the value(s) selected in the listbox. In the following example, the report has been configured to only show information for rules that are ready for production.

To disable the filter, uncheck the filter checkbox.

Note, that you can select several values by holding down the Ctrl key while clicking the cube names.

2.3.1.5 Hiding or Showing the Controls.

The 

button allows the controls to be hidden – and showed again. Click this to leave more room for the actual report contents.

2.3.1.6 Other Features

The generic report supports the following additional features:

- Refresh – Clicking [Refresh] forces the report to be refreshed, so that the contents reflect the control settings.
- Auto Refresh – By default, the report is updated every time you change a setting (e.g. change the selection in a filter listbox). For very large reports, it can take some time to update the report. You can turn of the automatic refresh by un-checking [Auto Refresh]. The report will then only be refreshed when you click [Refresh].
- Reset – Clicking [Reset] resets the contents off all controls.

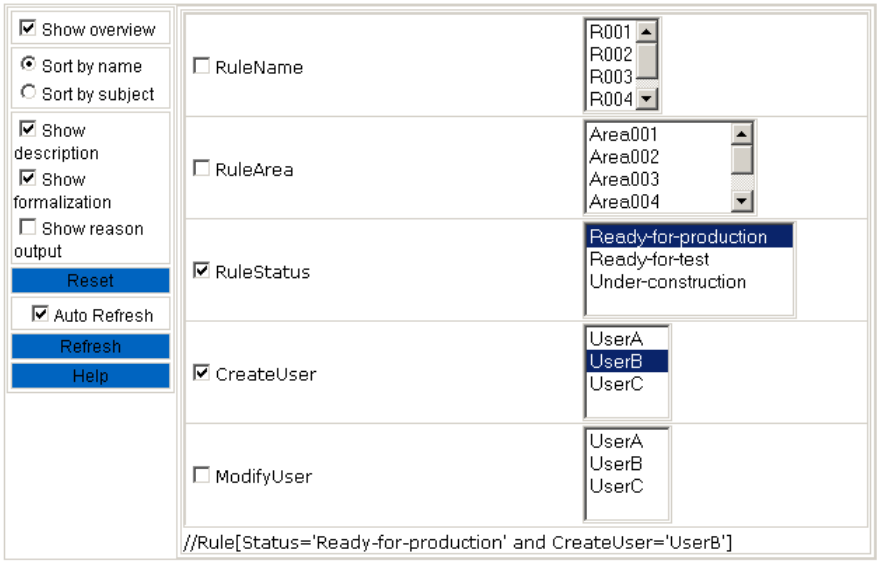
- Help – Shows a short help text with tips for using the report. The help text can be hidden again by clicking [Help] again.

2.3.1.7 Example

To show the description and formalization of all rules that are ready for production, and created by a specific user (UserB), complete the following steps:

1. Click the [Show cubes] checkbox.
2. Click the [Show description] checkbox.
3. Click the [Show formalization] checkbox.
4. Click Ready-for-production in the listbox next to the [RuleStatus] checkbox. This will automatically check the RuleStatus filter as being active.
5. Click UserB in the listbox next to the CreateUser checkbox. This will automatically check the CreateUser filter as being active.

The controls should now be set up as below, and the report will show the requested information.



To show rules for all users – instead of just for UserB – removed the CreateUser filter by unchecking the [CreateUser] checkbox.

2.3.2 Generic Report for Cross References

The generic report for cross-references allows cross-reference information on the rules and use of members in the rules to be reported; i.e. which rules use which members, or which members are used in which rules. The report is activated through [Report: Cross Reference] on the context menu for knowledge bases.

2.3.2.1 Perspective

You can choose whether to show the reference rule based, member based or both, by checking the [Show rule based] and [Show member based] checkboxes.

- When you have checked [Show rule based], the report will include a section listing all members used for each rule.
- When you have checked [Show member based], the report will include a section listing all rules in which each member is used.

2.3.2.2 Sorting

You can choose how the rules should be sorted, by selection either [Rule], [Dimension] or [Member] in the Sort by section.

2.3.2.3 Filtering

You can filter the report (i.e. only show information for selected rules or members), by using one or more of the filters on the right. The check box indicates whether the filter is active, while the listbox to the left shows all possible values (e.g. all rule names).

When a filter checkbox is checked, the report will only show information on rules with the value(s) selected in the listbox. In the following example, the report has been configured to only show information for the rule R003.

To disable the filter, uncheck the filter checkbox.

Note, that you can select several values in the list box by holding down the Ctrl key while clicking the cube names.

2.3.2.4 Hiding or Showing the Controls.

See section 2.3.1.5.

2.3.2.5 Other Features

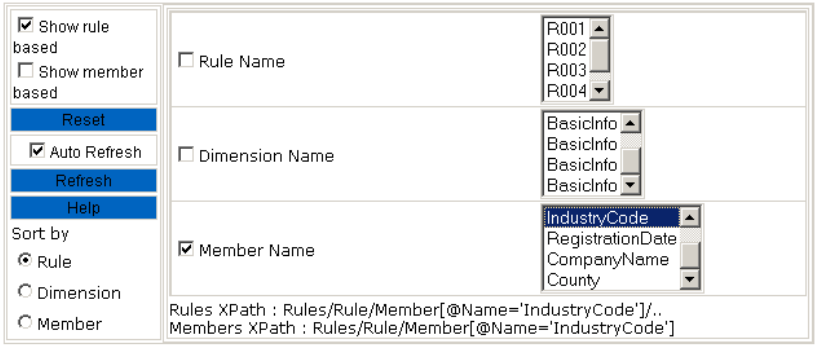
See section 2.3.1.6.

2.3.2.6 Example

To show the list of rules using the member IndustryCode, complete the following steps:

1. Ensure the [Show rule based] checkbox is checked (as it is by default when the report is opened).
2. Click IndustryCode in the listbox next to Member Name. This will automatically check the RuleStatus filter as being active.

The controls should now be set up as below, and the report will show the requested information.



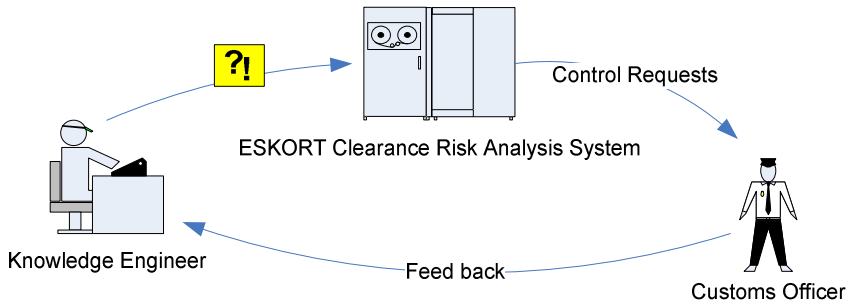
2.4 Knowledge Acquisition

Knowledge Acquisition is the process of acquiring knowledge about possible risk areas, which may lead to the development of new or improved rules.

Typically, a central team of knowledge engineers will be responsible for the implementation of rules, but they will also coordinate the efforts to 'tap' the knowledge of other people within the organization.

2.5 Analyzing Feedback

To ensure an increasing knowledge base quality the knowledge engineers will coordinate, collect and examine feedback from especially customs officers. Feedback may include comments on the validity of the observations made by existing rules, and/or suggestions for new rules.



3. Introduction to Rule Formalization

3.1 Overview

Rules need to be formalized using the formalization language to be evaluated by the risk analysis server.

Formalizing rules is a specialized task, which requires knowledge of the following aspects (in addition to general knowledge about the relevant ESKORT tools):

- The (internal) multi dimensional representation of declaration data
- Formalization language (especially regarding how data is being accessed)
- The context (environment) in which the knowledge base is being developed (cubes and their dimensions)

The following sections aim at establishing a basic understanding of the aspects mentioned above and introducing the main concepts and terminology.

For more information about rule formalization refer to [ESKORT Designer, Section IV – Formalization Language].

3.2 Accessing Data

Within the context of screening, one declaration is considered at a time. When writing the rules, data is accessed through a multi-dimensional view on available data in the declaration.

The data in the declaration can be viewed as spanning a number of *dimensions*, many of which are common to several tables within the declaration.

Cube - A data structure with declaration data spanning one or more dimensions. Each cube has a *measure* dimension corresponding to the (non-key) fields in the declaration.

Dimension - A set of values reflecting one way of classifying the data of the declaration (e.g. item number or declaration registration number).

Member - a value within a dimension (e.g. a specific declaration field). A *qualified member* is a member that specifically mentions the cube or dimension in which it is a member.

Cell - an individual value within a cube.

Coordinate - a set of members identifying a specific cell within a cube.

Tuple - a set of members making up part of a coordinate.

Data is always referenced by specifying a tuple. When specified within the context of a range, it is only necessary to specify members in the dimensions, which are not bound as part of the range.

For more information about environment refer to [ESKORT Designer, Section II – Introduction to Environments].

3.3 Basic Rule Formalization

Very basically at rule is of the format:

within <*default cube*> ***for*** <*ranges*>

if <*condition*> ***then*** <*actions*>

3.3.1 Default Cube and Ranges

The default cube is the cube that is used on all unqualified measures (all measures for which a cube name is not explicitly stated).

Ranges specify the tuples that will form the evaluation contexts for the rule within the area. The ranges are specified as a comma-separated sequence of tuples or set-value expressions.

In the ranges it is necessary to refer to a member of a dimension, a set of members from a dimension or a dimension itself. We call this to *bind* a dimension.

Note: The ranges must bind all dimensions in the default cube except the measure dimension.

For instance if a cube, ManifestLineOfPackages, has three dimensions: LineOfPackages, TransportDocuments and ManifestLineOfPackages (the measure dimension), we need to bind the LineOfPackages and TransportDocuments in the ranges.

If we just write the dimensions:

within (ManifestLineOfPackages) **for** (LineOfPackages, TransportDocuments)

the rule will be evaluated for all members in the LineOfPackages and TransportDocuments dimension.

Say the LineOfPackages dimension has three members (the years 1, 2 and 3). If we want the rule to be evaluated for only year 2, we refer to this member explicitly in the ranges:

within (ManifestLineOfPackages) **for** (TransportDocuments, {LineOfPackages.'2'})

and the rule will then be evaluated for all members in the TransportDocuments but only 2 in the LineOfPackages dimension.

3.3.2 Conditions

The conditions of a rule should express under which conditions the rule is intended to fire.

Conditions can test (examples):

- Specific values
- Relations between different values
- Changes over time in a specific value
- Existence (or non-existence) of a value
- Complex algorithms

The conditions are written as a logical expression, where different conditions can be combined using AND and OR.

3.3.3 Actions

The actions of a rule should express what should happen if the conditions of the rule fires.

The following actions exist:

- **Create observations**

  This is the main type of actions that is used in almost all cases. An observation contains information about

  - The rule that has fired
  - The declaration for which the rule fired
  - The period (if any) for which the rule fired
  - The profile for the observation and the related measures for that profile (e.g. the points if the profile is “RiskPoints”).

- **Make Extended Risk Profile Observation**

  When a Risk Profile (Trap Rule) fires an observation is made which identifies the firing Risk Profile along with a level of importance.

- **Activate areas or rules**

  An action of a rule can also be activating an area or another rule, i.e. this area or rule will be evaluated if the rule fires.

- **Assign**

  Assigns a value to a specific cell in a cube. This is usually used to store analysis results in the result cubes.

- **Bind Macro**

  Bind a macro (a small program section) to a name that subsequently can be executed by name.

- **Write String**

  Writes a text to the ESKORT Analysis Server’s standard output.

- **Write Cube to Result**

  Write the contents of a cube to the analysis result returned from the ESKORT Analysis Server.

3.3.4 Example of Rule

We have the cube *ManifestLineOfPackages* with dimension: *LineOfPackages*, *TransportDocuments* and *ManifestLineOfPackages* (the measure dimension). The measure dimension has two members: *DeclaredGrossWeight* and *DeclaredNetWeight*.

A simple rule using this cube could be:

within ManifestLineOfPackages **for** (LineOfPackages, TransportDocuments)
if
DeclaredGrossWeight < 0.5**DeclaredNetWeight*
then
Call Observation(...)

This rule will be evaluated for all members in the *LineOfPackages* dimension and all *TransportDocuments* dimension, meaning that the rule is checked for all combination of Line of Packages in all Transport Documents. For instance if there is two *TransportDocuments* and three *LineOfPackages*, the rule will be evaluated six times.

For each combination of the *LineOfPackages* and *TransportDocuments* the rule will look up the measures *DeclaredGrossWeight* and *DeclaredNetWeight*, and check whether *DeclaredGrossWeight* is less than half of *DeclaredNetWeight*. If this is the case it creates an observation in the results. This means that there can be created up to six observations from this rule, one for each combination of *TransportDocuments* and *LineOfPackages* where the rule fires.

For instance if we have:

TransportDocuments	LineOfPackages	Measure	Value
A	1	DeclaredGrossWeight	100
A	1	DeclaredNetWeight	150
A	2	DeclaredGrossWeight	90
A	2	DeclaredNetWeight	150
A	3	DeclaredGrossWeight	40
A	3	DeclaredNetWeight	100
B	1	DeclaredGrossWeight	1000
B	1	DeclaredNetWeight	1800
B	2	DeclaredGrossWeight	800
B	2	DeclaredNetWeight	2000
B	3	DeclaredGrossWeight	1100
B	3	DeclaredNetWeight	2000

Evaluating the rule will result in two observations, one for

TransportDocuments = A and *LineOfPackages* = 3

and one for

TransportDocuments = B and *LineOfPackages* = 2

Refer to [ESKORT Designer, Section IV – Formalization Language] for more examples.