QuickRDA Modeling System

Metamodel & Template Developer Guide

Erik Eidt

Version 2.7

March, 2011

***Copyright (c) 2011, 2012 Hewlett-Packard Development Company, L.P.***

[Spreadsheet Template Developer’s Guide 4](#_Toc336067011)

[Customizing the Standard Template 4](#_Toc336067012)

[Super-Property Expressions 5](#_Toc336067013)

[Attributes 5](#_Toc336067014)

[Coloring 5](#_Toc336067015)

[Style 5](#_Toc336067016)

[Introduction 5](#_Toc336067017)

[Meta Model Developer’s Guide 6](#_Toc336067018)

[Theory of Operation 7](#_Toc336067019)

[References 8](#_Toc336067020)

[Domain Model Interchange 8](#_Toc336067021)

[GraphViz 8](#_Toc336067022)

***Table of Tables***

*No table of figures entries found.*

***Table of Figures***

[Figure 1. Modeling Layers 5](#_Toc336067023)

[Figure 2. Organization Of Modeling Layers in the QuickRDA Tool 6](#_Toc336067024)

# Spreadsheet Template Developer’s Guide

Unfinished.

Each Excel table form used by QuickRDA uses self-describing meta-data. There contain three special rows. Samples can be seen in the QuickRDA Template, these are the hidden rows 6 & 7. *(Row 8 is a blank row that is hidden, and makes it easy to insert a new row at the top of the table without getting the formatting of rows 6 or 7.)*

The first row in the table, a non-hidden row (and row 5 in the standard template) is the normal column header, which of course, names the columns. These names must be unique within the table (though this is already an Excel requirement). These names can be referenced from the other self-describing rows.

The next 2 rows immediately below the column headers (usually hidden) are special, adding additional self-description of the table and columns.

The first such row indicates the type of the entities in the column, such as Roles, Responsibilities, etc… These are generally used to add nodes to the graph.

The second such row indicates the formula that supports adding edges to the graph.

### Customizing the Standard Template

Creating a new template can be done by customizing the standard template. Make a copy of it in another Excel workbook, or as another tab (worksheet) within the same workbook. You can rearrange column ordering; also, columns can be deleted to customize the template for a particular kind of capture. For example, a template can be customized from the standard one for capturing only information about actors, by deleting all columns not relevant to actors.

*Note that some caution is required in deleting columns that are referenced by other columns (via the metadata) even if the (visible portion of) the other columns are blank. Columns that reference other columns need to be deleted together to avoid an annoying but harmless and sometimes helpful error message.*

### Super-Property Expressions

x->y is an expression that represents the super-property of all properties whose domain is compatible with x and range is compatible with y.

x->y

### Attributes

#### Coloring

The color attribute specifies coloring for nodes and edges. It can be specified in hex, decimal, or by using the fill color of the Excel cell. To use the cell fill color the value “y” must be present.

#### Style

The style attribute specifies an artifact whose diagrammatic properties, such as color and shape, should be used.

# Introduction

Model layering

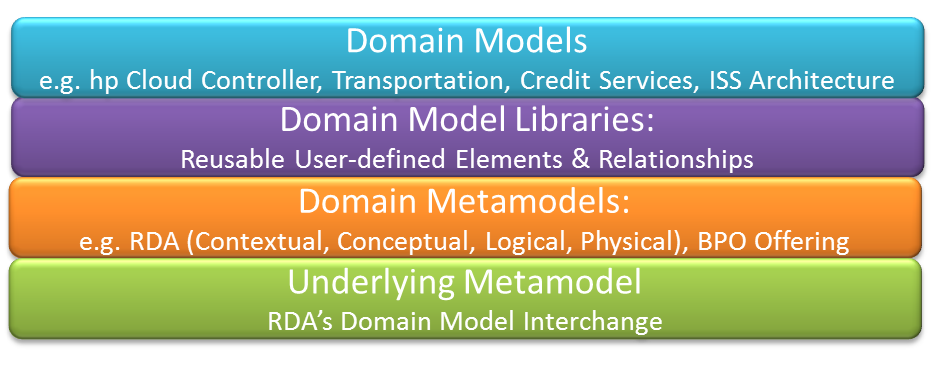


Figure 1. Modeling Layers

The current version of the tool does not support model libraries as a layer.

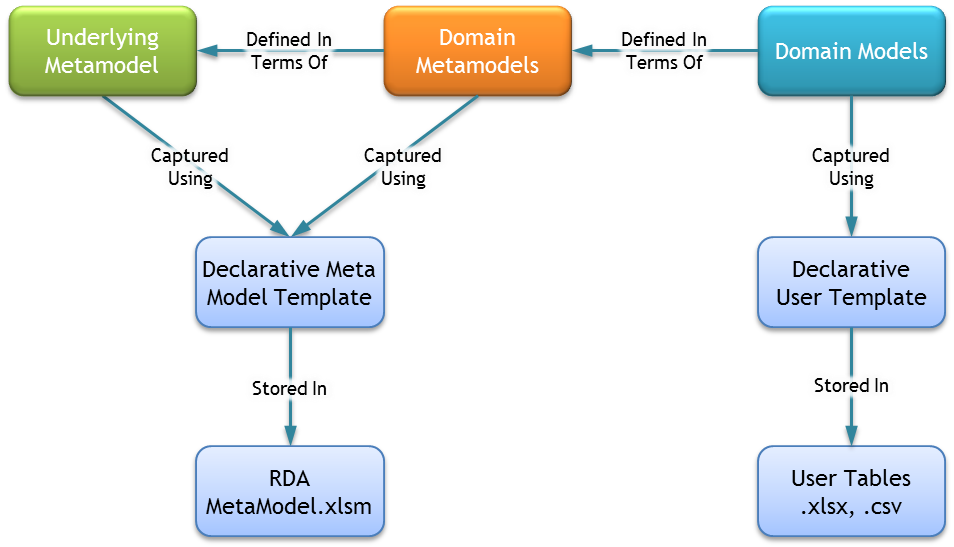


Figure 2. Organization Of Modeling Layers in the QuickRDA Tool

# Meta Model Developer’s Guide

Unfinished.

Each row in the metamodel names a thing or concept.

Each of thing or concept can function like an instance, a class, a property, or a relation; this is not either/or, each row can function as any of these at the same time.

# Theory of Operation

The QuickRDA tool assembles information from spreadsheets into an internal graph of information, which can then be diagrammed with a variety of options. The tool composes an underlying metamodel, with domain metamodels written using the underlying metamodel, with domain models written with the domain metamodels. The underlying metamodel is key to understanding this operation.

The key concepts of the underlying metamodel are its notion of concept, class, property, and relation. Models assembled with

# References

## Domain Model Interchange

## GraphViz

For more information on the DOT language, see:

Website: <http://www.graphviz.org/>

Download: <http://www.graphviz.org/Download.php>

Documentation, basic syntax: <http://www.graphviz.org/doc/info/lang.html>

Documentation, attributes: <http://www.graphviz.org/doc/info/attrs.html>