QuickRDA

Examples Guide

Erik Eidt

Version 3.21

August, 2011

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

[Introduction 5](#_Toc300147050)

[Templates & Evocation 5](#_Toc300147051)

[Template Definition 6](#_Toc300147052)

[Simple Template Evocation 8](#_Toc300147053)

[Adornment of Evocation 10](#_Toc300147054)

[Concept Substitution in Template Evocation 12](#_Toc300147055)

[The Build Tab for this Example 14](#_Toc300147056)

[Advanced Template & Evocation Topics 16](#_Toc300147057)

[Relationships 16](#_Toc300147058)

[Nested Evocations 16](#_Toc300147059)

[References 17](#_Toc300147060)

[QuickRDA: Introduction & Overview 17](#_Toc300147061)

[QuickRDA: RDA Domain Language 17](#_Toc300147062)

[QuickRDA: RDA Contextual Modeling 17](#_Toc300147063)

[QuickRDA: RDA Conceptual Modeling 17](#_Toc300147064)

[What Makes Good Architecture 17](#_Toc300147065)

***Table of Tables***

[Entry Form 1. A Template defined in a Fact Entry Form 7](#_Toc300147066)

[Entry Form 2. Simple Template Evocation 8](#_Toc300147067)

[Entry Form 3. Simple Template Evocation 10](#_Toc300147068)

[Entry Form 4. Artifact Substitution in Evocation 12](#_Toc300147069)

[Table 5. The Templates & Evocations Build Tab 14](#_Toc300147070)

***Table of Figures***

[Diagram 1. The Templates, Illustrated 7](#_Toc300147071)

[Diagram 2. Simple Template Evocation, Illustrated 8](#_Toc300147072)

[Diagram 3. (3) Simple Template Evocations, Illustrated 11](#_Toc300147073)

[Figure 4. Template Evocation with Substitution (and Adornment) 13](#_Toc300147074)

# Introduction

This document discusses some of the features of the QuickRDA Modeling System using supplied examples.

# Templates & Evocation

QuickRDA 3.20 provides a new mechanism for expressing common domain facts and reusing them, called the template. Templates are defined as a set of facts; later they can be recalled as an evocation, conjuring, or summoning of those facts.

The accompanying example is named **Templates and Evocations.xlsx** and can be found in the Examples folder of the released bits.

*The templates of topic in this document are for the expression and subsequent reuse of domain facts. The terms template & evocation, should not be confused with the earlier usages of the term template in the QuickRDA modeling system. QuickRDA had previously used the term template for the form of source units, which are Excel spreadsheets based on a common yet highly customizable entry form (and hence the original usage of the term template). These source units are now referred to as Domain Fact Source Tables to distinguish them from the template & evocation mechanism, and are the topic of* QuickRDA (6) Modeling System Domain Fact Entry Form User Guide*. Domain Fact Entry Forms are a mechanism based on entry into a custom Excel spreadsheet for regular, structured, yet highly customizable capture of domain facts, whereas templates & evocations are way to reuse captured domain facts to create other or more complex domain facts.*

The fact and reuse mechanism of templates and evocations is the subject of this section. This mechanism consists of three parts:

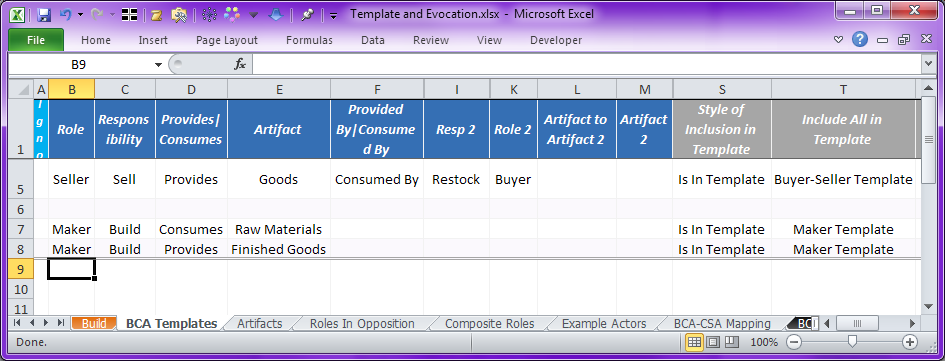
1. The expression of a set of source facts that belong to a named template, or template definition.
2. The evocation of the named template, or template evocation.
3. Customization of the evocation of named templates, or concept substitution.

A template has members. A template definition constitutes the name of the template along with assembly of included members. As with any fact capture, full illumination of the template can and generally requires carrying over onto multiple rows in a source table. Using a template (or reusing the contents of a template) is the conjuring of these members, with modification, for a particular purpose, called an evocation.

## Template Definition

A template is defined by naming it, and by designating the inclusion of members into it.

The members of a template are concepts (e.g. named entities of named types). Further, it is useful to note that in the QuickRDA modeling system, relationships are also concepts; thus, relationships are also significant as members of templates.



Entry Form 1. A Template defined in a Fact Entry Form

Here, the first fact row, 5, develops two simple roles in opposition: Seller and Buyer. These roles are included in a template called Buyer-Seller Template, also on row 5. Further, a Maker role is defined, and included to the Maker Template, rows 7 & 8. We can visualize the templates by selecting the Template diagram in the build table:

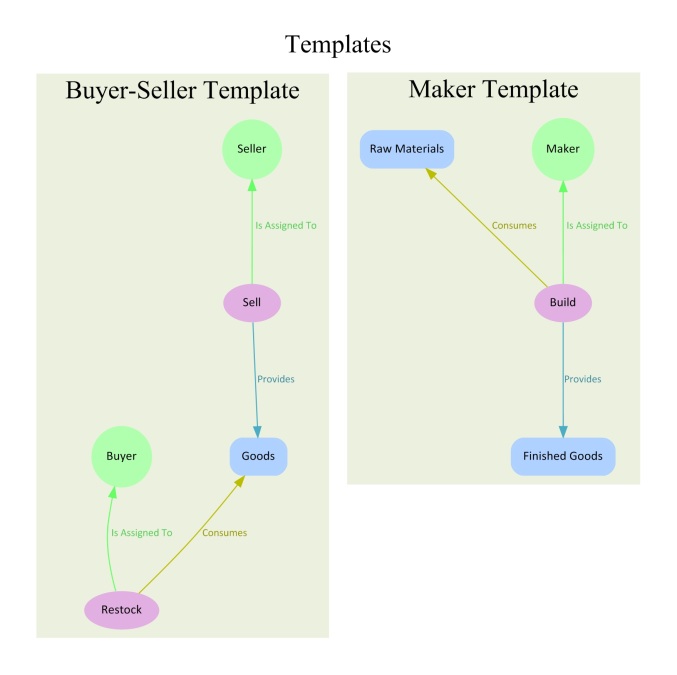
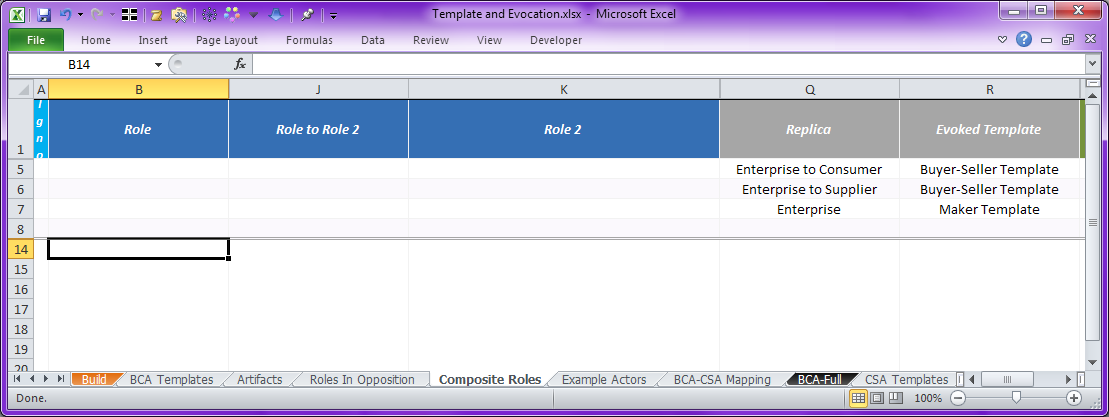


Diagram 1. The Templates, Illustrated

## Simple Template Evocation

To use a template, the template is evoked. An evocation is given a name (which is different from the template and from other evocations).

The following domain fact entry form evokes the above defined templates. We use the term evoked in the sense of declaratively conjuring or summoning (to differentiate from invocation by calling). Buyer-Seller Template is evoked twice, Maker Template once. Each evocation takes one row:



Entry Form 2. Simple Template Evocation

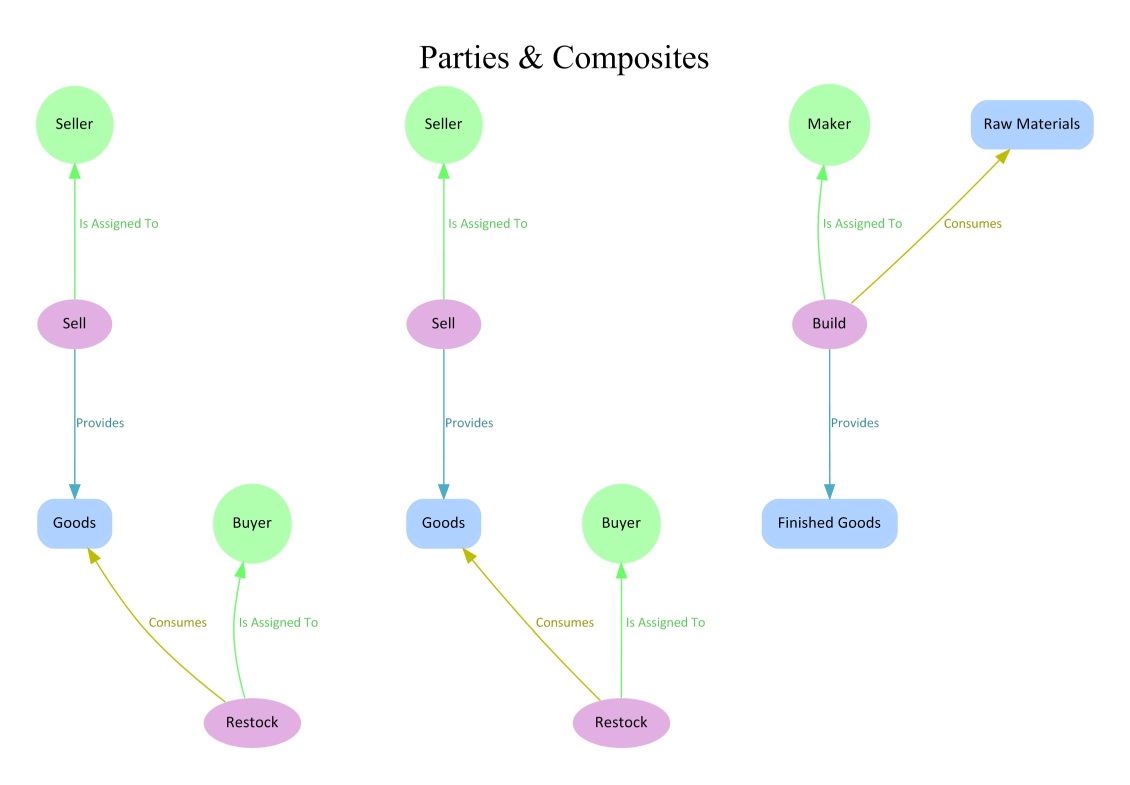


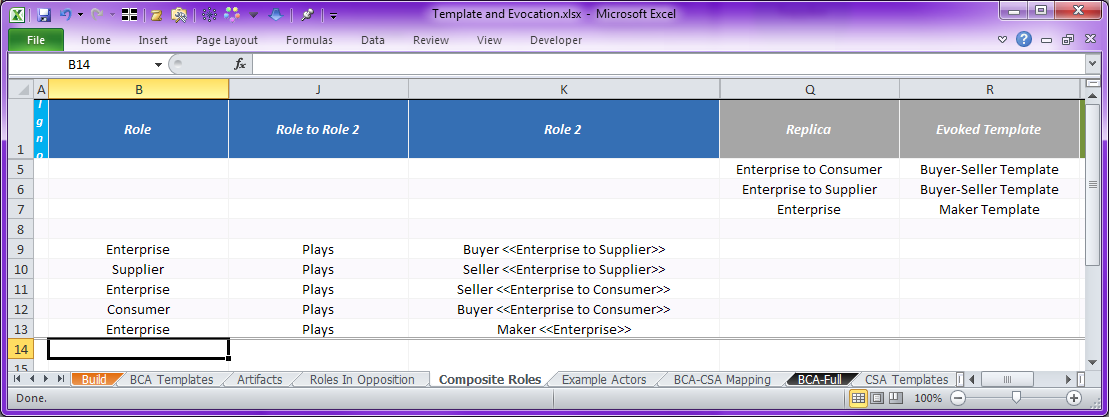
Diagram 2. Simple Template Evocation, Illustrated

Note that the build target for this diagram is Parties & Composites; *however, it is built with rows 9-13 of the Composite Roles tab hidden*.

## Adornment of Evocation

Adornment allows the addition of modeling information, domain facts, to the evocation of a template.

The following entry form shows the same three template invocations described in the above section (still in rows 5-7), plus, adornment of the evoked templates with additional model content. Rows 9-13 adorn the evoked template, namely in column K, by mention of the evoked member names. Evocation member names are referred to by adding the evocation name as a differentiator to the member name.



Entry Form 3. Simple Template Evocation

(A differentiator is text within double angle brackets, whose function is to indicate a different version of the same named item.)

Thus, Buyer <<Enterprise to Supplier>> refers to the Buyer role that was in the evocation named Enterprise to Consumer, which was an evocation of the template Buyer-Seller Template.

The following diagram illustrates the adornment. By having the Enterprise play the 3 of the roles, one from each template, it constructs a large domain model. The other 2 roles from the Buyer-Seller Template are played by our composite roles Supplier and Consumer.

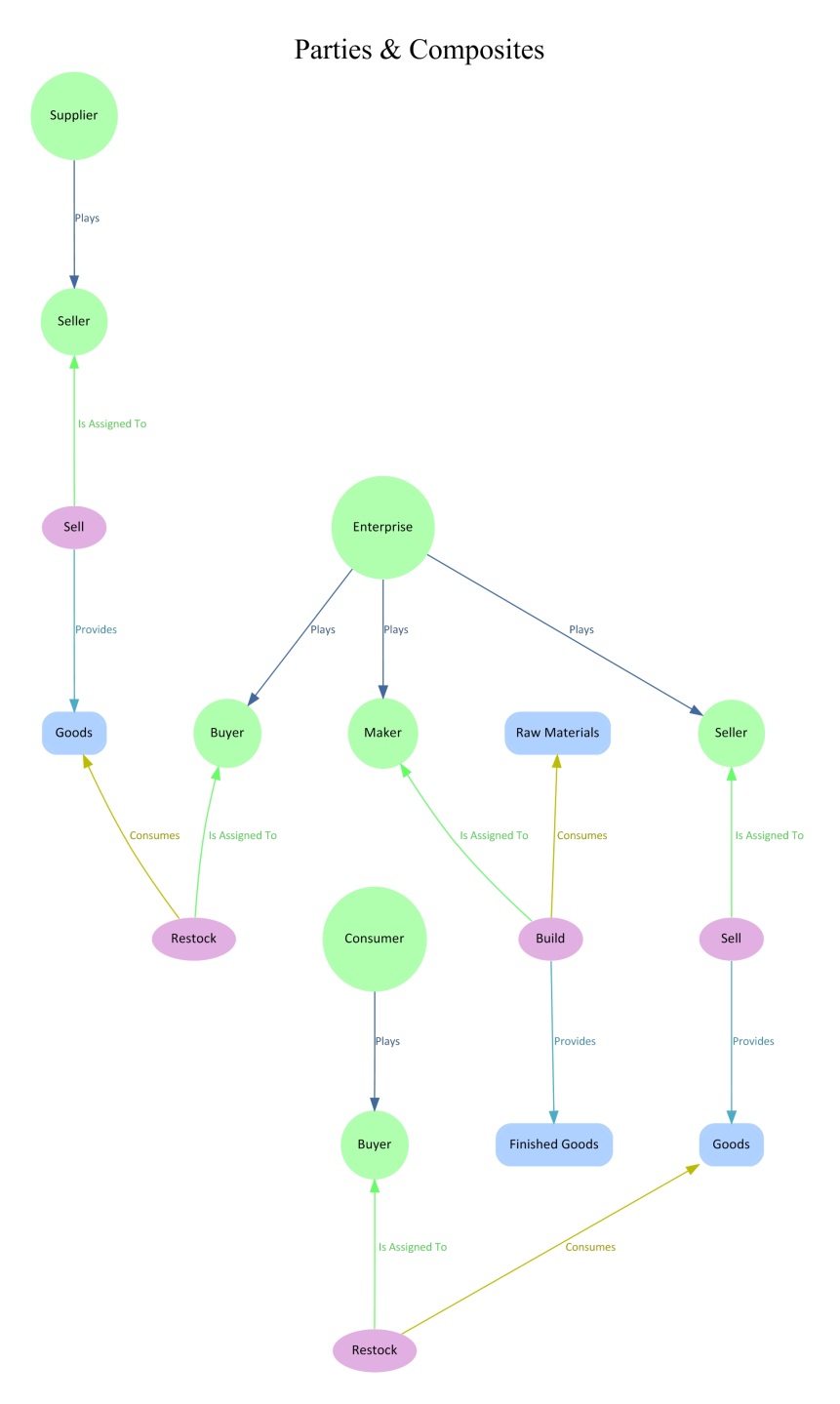


Diagram 3. (3) Simple Template Evocations, Illustrated

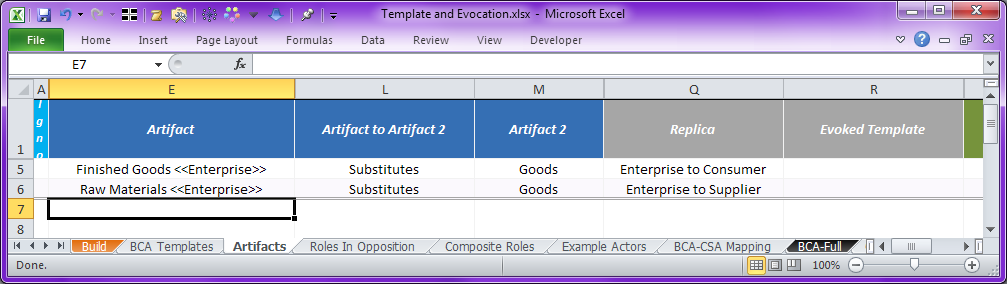
Note that the build target for this diagram is once again Parties & Composites, *with all rows of the Composite Roles tab revealed*.

## Concept Substitution in Template Evocation

Evocation can also be done with substitution. Any element of the template can be substituted for some other element. The following example shows two artifact substitutions, both of Goods, which are substituted with Raw Materials, and again with Finished Goods.

This substitution further elaborates our domain model, connecting the Maker role to the Buyer and Seller function of the Enterprise.

The following Entry Form shows the substitution of the artifacts. (Substitutions for Roles and for Responsibilities are also possible, though using a tab other than the Artifacts tab will be necessary.)



Entry Form 4. Artifact Substitution in Evocation

These facts declare the substitution of Finished Goods with Raw Materials.

To express substitution, a statement is made indicating that Finished Goods substitutes for Goods, and, another that this substitution statement itself is attached to the evocation. This entry form accomplishes this once on each of the rows 5 & 6.

Note carefully the use of differentiators in column E and lack thereof in column M.

Technically speaking, a differentiator not required for substitution to work; however, since in this case, the subject of our substitution (column E) is itself an evocation member — the result of an evocation of the Maker Template as Enterprise — a differentiator is used to refer to it.

The object of substitution (column M) is used to specify a member within the template itself — i.e. rather than within any evocation — hence the appropriate lack of use of differentiator in column M.

The following diagram illustrates the resulting model. Goods in each of the two template evocations have been replaced by a more specialize artifact giving better meaning to the model.

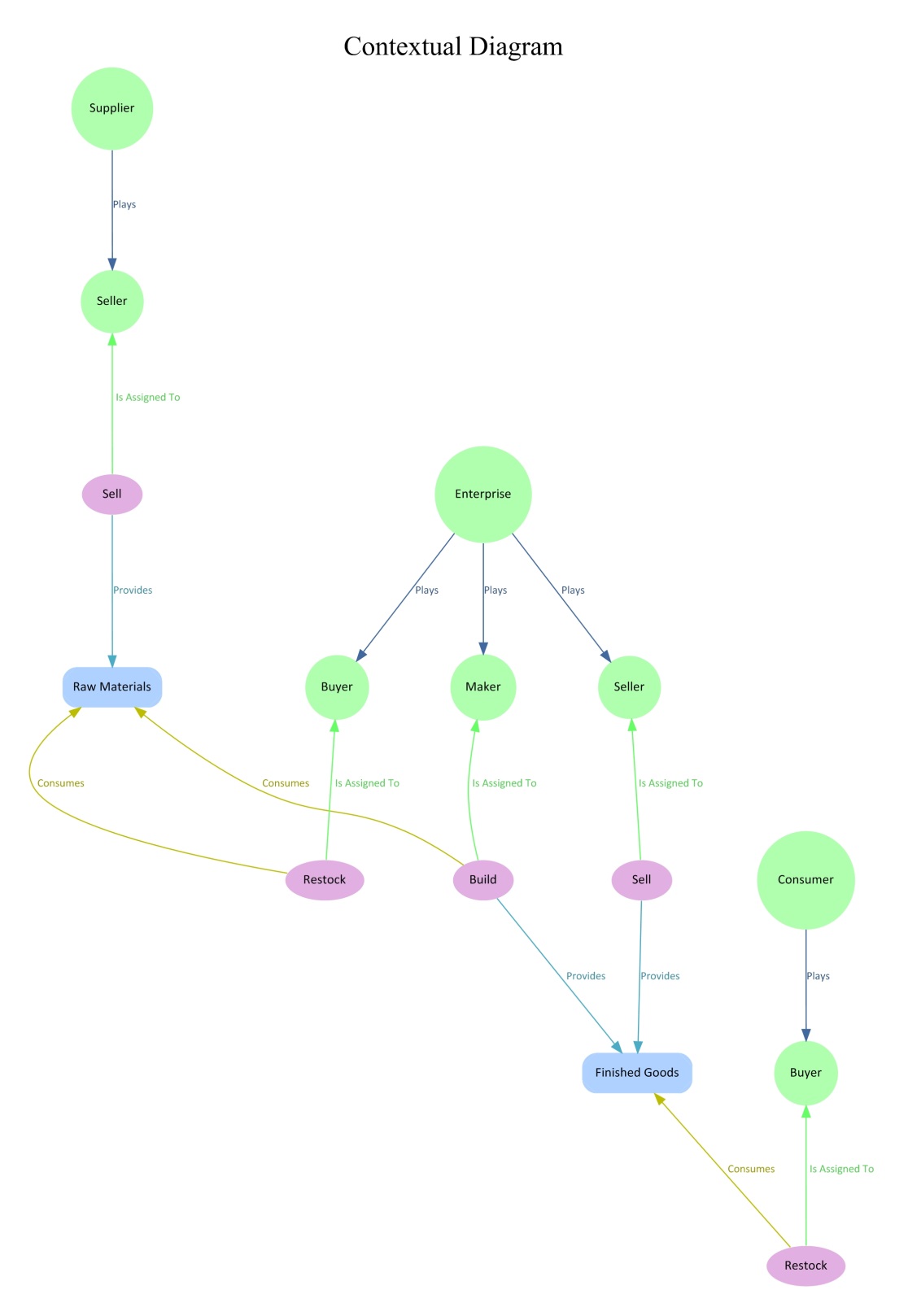


Figure 4. Template Evocation with Substitution (and Adornment)

## The Build Tab for this Example

The following picture is of the build tab from this example.

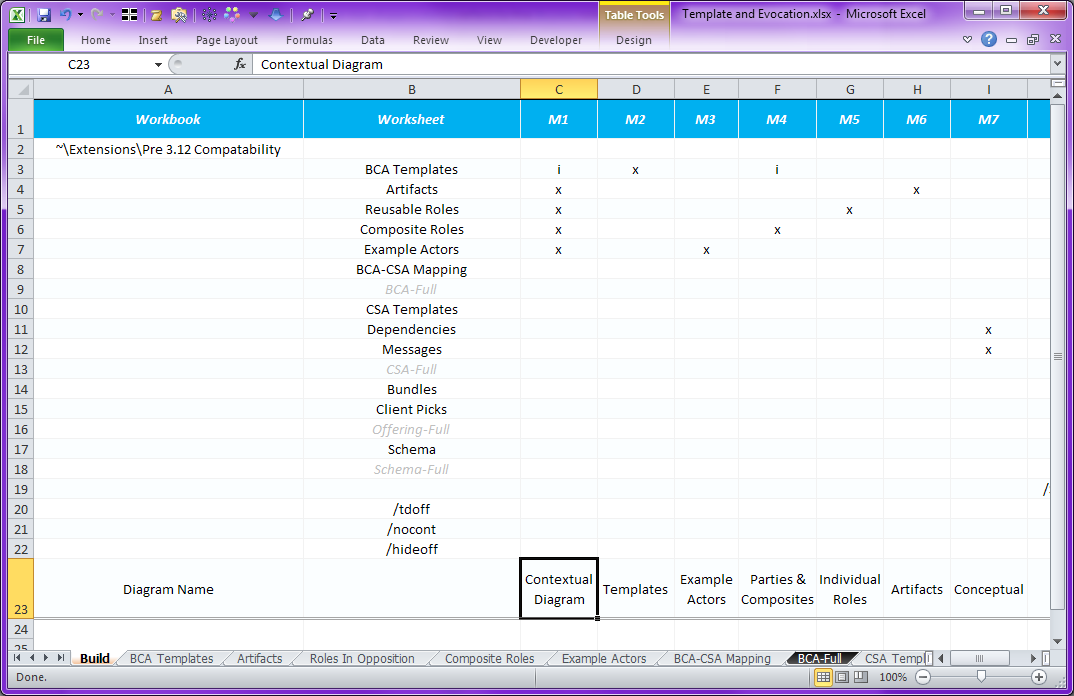


Table 5. The Templates & Evocations Build Tab

Let’s start with the Templates build target, in column D. It’s easy, simple diagramming the templates themselves, alone, using the visible inclusion into the model.

Next, Parties & Composites combines the Templates tab with the Composite Roles tab. For this diagram, it is necessary to access the modeling content of the templates since they are invoked by the Composite Roles tab. This makes the defined templates available to the tool for further processing, namely the evocations in the other tab. However, we don’t want to graph the template content itself, really just the resulting evocations. Thus, in this case, the template tab is included in the diagram using the “i” (invisible inclusion) method, which reveals the tab’s model content to the tool, but instructs for omission from the generated diagram.

This same technique is used for the Contextual Diagram, which in addition, includes the artifacts tab that describes the substitutions.

Note that evocations themselves have a form (as concepts and relationships) but are by default not shown in diagrams, in favor of showing just the consequences of evocation. Evocations themselves can be revealed by selecting the /hideoff option on any given diagram.

Further, the names of the members of evocations can be revealed using the /tdoff option. This can be useful in determining the exact name to use for adorning an evocation as well as in debugging.

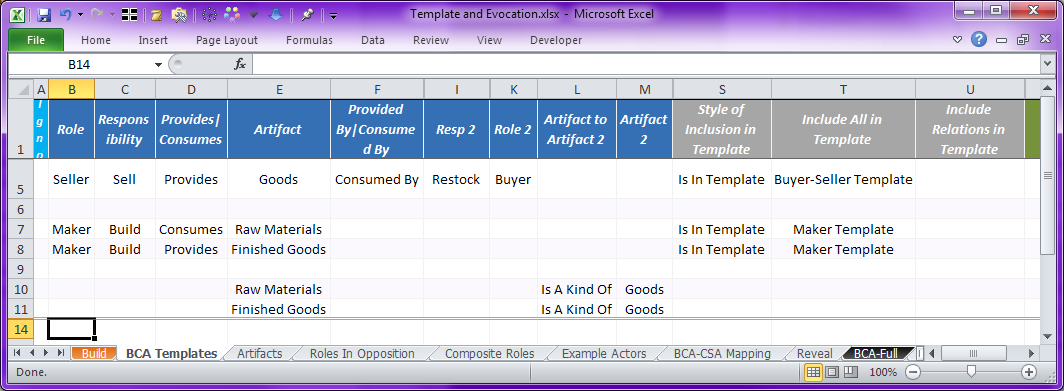
And lastly, the /nocont option can be used to show things without containment, everything will be shown as first class entities and relationships, sometimes useful in understanding what’s in a group, such as a template.

## Advanced Template & Evocation Topics

### Relationships

Relationships (between other members of the template) are themselves either in or out of the template as specified by the entry form. Most of the time, you’ll probably want relationships between template members to also be in the template, to get the expected behavior of them being evoked along with the other members.

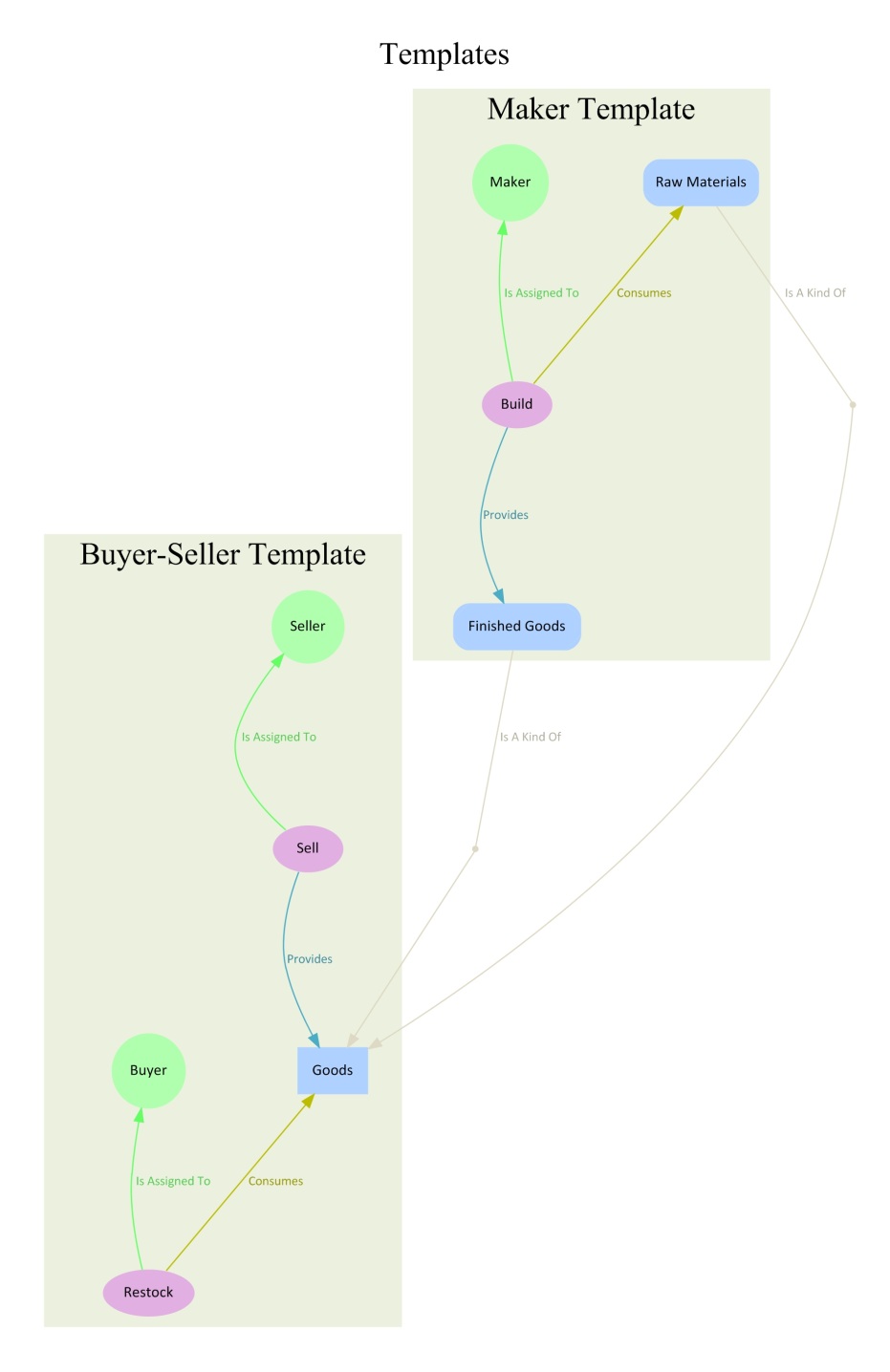
However, there are times when that isn’t always the case. Sometimes, a relationship may capture certain descriptive qualities for a member of the template without the modeler wanting that to be in the template. Below is an example of the Is A Kind Of relationship between Raw Materials and Goods, and Finished Goods and Goods on rows 10-11.



Entry Form 6. Relationships Outside of Templates

The diagram for this is shown below. In particular, note how the relationships described in rows 10-11 are specifically diagramed to show they are not part of either template — this can be seen from the location of the node for the relationship Is A Kind Of.

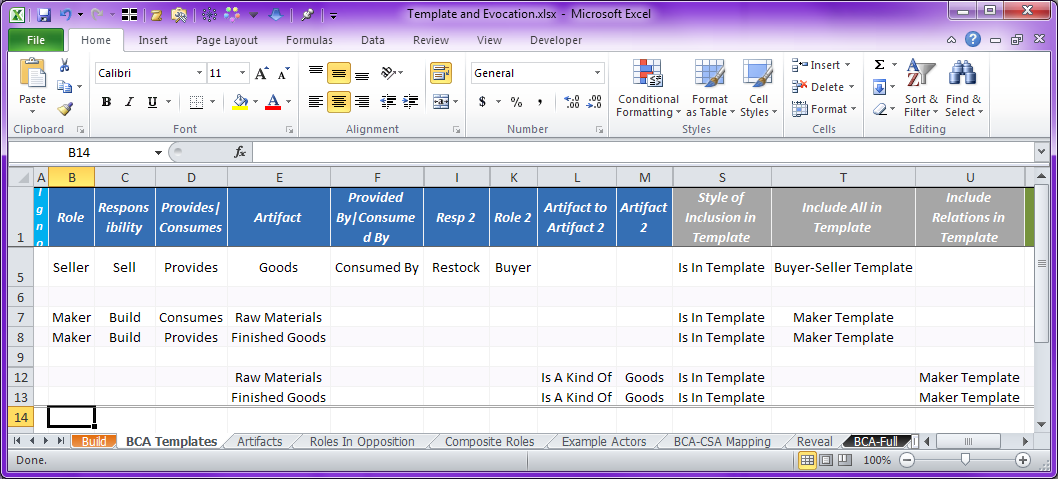
*Also note that when a relationship connects two nodes, and, the relationship and both nodes are in the same grouping (a template in this case), then the graphic rendition omits the explicit relationship node. Conversely, omission of the explicit node for relationships should be viewed as an indication that the relationship itself is in the same group (in this case, a template) as the nodes it connects — such is the case with the Is Assigned To, Provides, and Consumes relationships shown here.*



*Explicitly Graphed Relationship Nodes*

Diagram 5. Relationships Outside Template, Illustrated

Other times, when multiple templates are being defined we may want to include the relationships between template members in only one of the templates. Below, we have the same relationships (now on rows 12-13) describing that Raw Materials Is A Kind Of Goods (and the same with Finished Goods). In addition, below, we’ve chosen to add these relationships to the Maker Template.



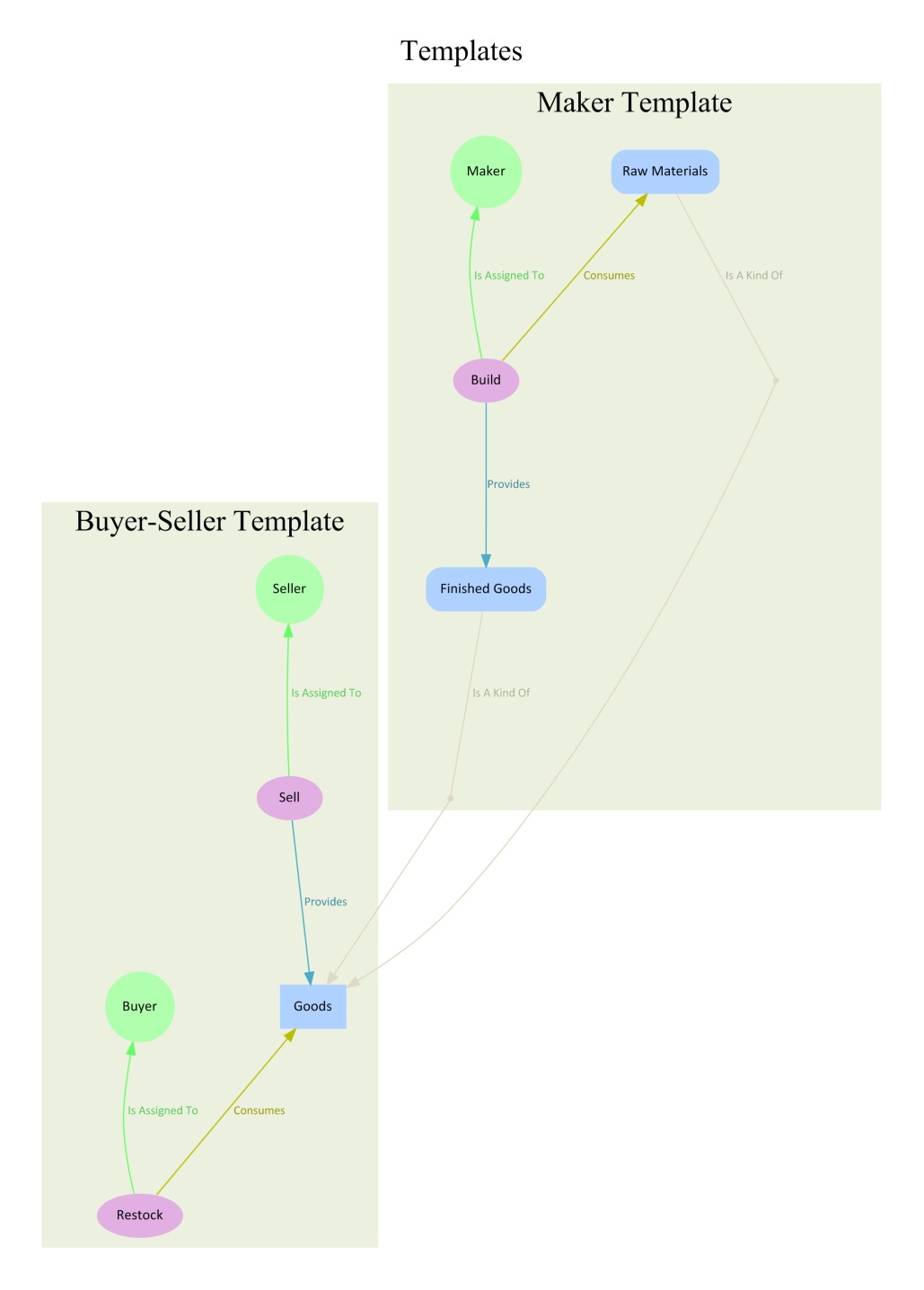
Entry Form 7. Relationships Inside Template

*It is relatively simple to place simple concepts in the template without the relationships they are involved in — just mention that concept alone on the row that also specifies inclusion in a template.*

*However, the opposite — placement of a relationship in a template without also adding the subject and object of the relationship — requires the use the secondary column U, “Include Relations in Template”, for specifying the template name instead of the primary column T, “Include All in Template”.*

Note in the above example, the use of Maker Template in column U, which specifies adding only the relationships mentioned in columns B thru M, not the nodes as well.

The graphic rendition for this is shown below. In particular, note that the graphic rendition for the Is A Kind Of relationships places the explicitly graphed relationship nodes within the Maker Template; it does this because even though the relationships are in the same template as their subjects, the relationships are not in the same template as their objects. Graphing a relationship that simply connected the subjects to the objects would leave ambiguous where the relationship was actually located (whether in the Maker Template, the Buyer-Seller Template, or in neither template).



*Explicitly Graphed Relationship Nodes*

Diagram 6. Relationships Inside Template, Illustrated

### Nested Evocations

Evocations within templates are possible; such evocations are delayed until evocation of the outer most enclosing template. This results in a nested naming scheme for template members, which needs to be understood if the evocation is to be augmented with additional concepts or relationships.

*$$$ TBD Finish/Edit this section.*

*It is reasonable that template member participate as subjects and objects in relationships that are themselves either in or not in the template. For example, a template may define a responsibility, Report Weekly Status, and a relationship, Is Assigned To, such as: Report Weekly Status Is Assigned To Employee. Presuming that Employee is outside of the template, yet the Is Assigned To relationship is within the template, then evocations of the template will replicate the relationship with variable subjects pointing at a common Employee as object.*

*$$$ EXAMPLE*

*By contrast, if the Is Assigned To is not in the template, then evocation of the template will replicate the responsibility but not the relationship; only the template member itself will participate in the relationship. This is useful in some cases, for example, when providing information about the template member that shouldn’t be replicated with evocation (e.g. constraints not yet implemented), or, when the relationships add (class) type information to the concepts, which does get replicated with the member concepts.*

# References

##### QuickRDA: Introduction & Overview

##### QuickRDA: RDA Domain Language

##### QuickRDA: RDA Contextual Modeling

##### QuickRDA: RDA Conceptual Modeling

##### What Makes Good Architecture