Exploring VDML with Spreadsheets

Fred A. Cummins

December 26, 2014

# Table of Contents

[Table of Contents 1](#_Toc408322073)

[1 Introduction 3](#_Toc408322074)

[2 Approach Highlights 3](#_Toc408322075)

[2.1 Variances from the VDML specification 3](#_Toc408322076)

[2.2 Linked cells 4](#_Toc408322077)

[2.3 The Example Model Fragment 5](#_Toc408322078)

[3 VDML Spreadsheet Tables 5](#_Toc408322079)

[3.1 Activities 5](#_Toc408322080)

[3.1.1 Roles 7](#_Toc408322081)

[3.1.2 Capability Method/Business Network 7](#_Toc408322082)

[3.1.3 Activity 7](#_Toc408322083)

[3.1.4 Deliverable Flows 9](#_Toc408322084)

[3.2 Business Item Definitions 9](#_Toc408322085)

[3.3 Role Definitions 11](#_Toc408322086)

[3.4 Capabilities 11](#_Toc408322087)

[3.5 Organization 12](#_Toc408322088)

[3.6 Value types 13](#_Toc408322089)

[3.7 Measurements 14](#_Toc408322090)

[3.7.1 Role Names 14](#_Toc408322091)

[3.7.2 Method/Business Network 15](#_Toc408322092)

[3.7.3 Activity Name 15](#_Toc408322093)

[3.7.4 Input number 15](#_Toc408322094)

[3.7.5 Input Mode 16](#_Toc408322095)

[3.7.6 Activity Use Factor 16](#_Toc408322096)

[3.7.7 Output Deliverable Flow 17](#_Toc408322097)

[3.7.8 Planning Percentage 17](#_Toc408322098)

[3.7.9 Output Net Use Factor 17](#_Toc408322099)

[3.7.10 Activity Value Contributions 17](#_Toc408322100)

[3.7.11 Cumulative Value Contributions 18](#_Toc408322101)

[3.7.12 Delegation Group 18](#_Toc408322102)

[3.7.13 Notes 19](#_Toc408322103)

[3.7.14 Example cost measurement computations 19](#_Toc408322104)

[3.7.15 Value aggregations in delegation 20](#_Toc408322105)

[3.7.16 Value aggregations for shared stores 21](#_Toc408322106)

[3.8 Value propositions 22](#_Toc408322107)

[3.9 Objectives 24](#_Toc408322108)

[4 Applications of a Spreadsheet-based Model 24](#_Toc408322109)

[4.1 Capability Analysis 25](#_Toc408322110)

[4.2 Value Contribution Analysis Modeling 25](#_Toc408322111)

Exploring VDML with Spreadsheets

Fred A. Cummins

January 6, 2015

# Introduction

This paper describes a technique the application of the VDML (Value Delivery Modeling Language) metamodel using spreadsheets. The technique is illustrated in a companion Excel workbook that is the source of figures in this document. It was developed to explore the application of VDML, particularly the aggregation of value measurements to support value propositions and objectives, and it provides a way of exploring VDML analysis in lieu of the availability of a full VDML modeling tool. It does not include some of the more complex and sophisticated aspects of a robust VDML modeling tool.

VDML was recently adopted by OMG (Object Management Group), see <http://www.omg.org/spec/VDML/1.0/Beta1> for the specification. See also <http://fredacummins.blogspot.com/2013/02/value-delivery-modeling-language-vdml_4878.html> for an overview of VDML. VDML-compliant tools are currently under development. In the meantime, business analysis and design based on VDML can provide significant value to business leaders.

While VDML-compliant tools will be important for realization of the full value of VDML modeling, conventional spreadsheet tools can be used to realize significant benefit and capture small-scope business models that may later be transformed to use a VDML-compliant modeling tool. This technique has been developed based on the functionality of MS Excel.

In this article, we will start with highlights of the general approach using spreadsheets. Next we will discuss the multiple spreadsheets that represent a VDML scenario (the model of a particular state of the business over a stable operating period). Finally, we will consider potential applications of such a model.

# Approach Highlights

This section presents highlights of the approach as a foundation for understanding the spreadsheet discussions as well as the limitations of the approach.

## Variances from the VDML specification

Spreadsheets restrict the ability to fully represent VDML models. Consequently, this approach simplifies certain aspects of VDML. The primary limitations are outlined, below.

* There is no recognition of a capability method owner as distinct from a capability method provider. In full VDML, this allows a single method specification (owner) to be offered by multiple organizations (providers).
* Resources are only considered as business items that occur in deliverable flows or may be held by a store. In full VDML, resources may be managed as reusable and are owned by an organization.
* Only business networks and capability methods are modeled with Activity networks. In full VDML, any collaboration can have activities.
* Measurements of shared capability methods are the same for all uses of the shared method. Full VDML manages measurements based on the context of each use.
* The spreadsheet model has cells for “use factors.” In a full VDML model, these measurements are implemented as a specific value type. They are treated differently in the spreadsheet model due to their unique role in the computation of other value type measurements.
* The Resource Use element is not represented, so a user must set appropriate use factors for input flows, activities and output flows. In full VDML, the Resource Use element controls the selection of inputs, input quantities and the association of inputs with certain outputs.
* Business networks and capability methods must have unique names for delegation and for distinguishing the same capability provided by different organizations. Deliverable flows are uniquely identified within a collaboration by the business item name, source and recipient. Full VDML represents associations/relationships that are not dependent on identifiers.
* The spreadsheet model does not provide a mechanism for passing a role assignment through delegation nor treating a role participant as a business item. For example, a primary care physician has a role in a treatment method, and the same primary care physician should be the participant in sub-methods for the patient’s treatment. A patient has a role in treatment decisions but is also the subject business item of treatment activities.

For the most part, these limitations will not be significant to early VDML users.

## Linked cells

In order to maintain consistency of the model involving multiple spreadsheets, some cells of a spreadsheet are linked to obtain their values from cells in a source spreadsheet.

The following table lists the spreadsheet tables and cells that are the master values for certain modeling elements, and it identifies the tables that reference them. Note that some spreadsheets contain multiple tables.

The tables that contain these master cells cannot be sorted because the links will be corrupted. Tables that do not contain any master cells might be sorted without corruption of links to other spreadsheets, but the spreadsheet format may otherwise prohibit sorting. Cells in any table can be cut, pasted, inserted or deleted without corrupting links.

|  |  |  |
| --- | --- | --- |
| **Source Table** | **Source Cells** | **Used by** |
| Roles | Role Name | Activities, Measurements |
| Capabilities | Method/Business Network name | Activities, Measurements |
| Activities | Activity Name | Measurements |
| Activities | Business Item name, Planning Percent | Measurements |
| Business Item Definition | Business Item Name | Activities |
| Value Types | Value Type Name, Measurement | Measurements, Value Propositions, |
| Measurements | Cumulative Value Contribution (measurements) | Value Proposition, Objective |

## The Example Model Fragment

The discussions of the spreadsheet tables in the next section incorporate a simple example, business model fragment. The example has two parts: a business network exchange with a customer for receipt of orders, and a fragment focused on activities that assemble a table from four legs and a table top, then package the table for shipment. The assembly operation produces some defective tables that are passed by delegation to a repair service. Defective tables are returned as repaired or scrap.

# VDML Spreadsheet Tables

In the following paragraphs we will discuss the structure and content of the following tables:

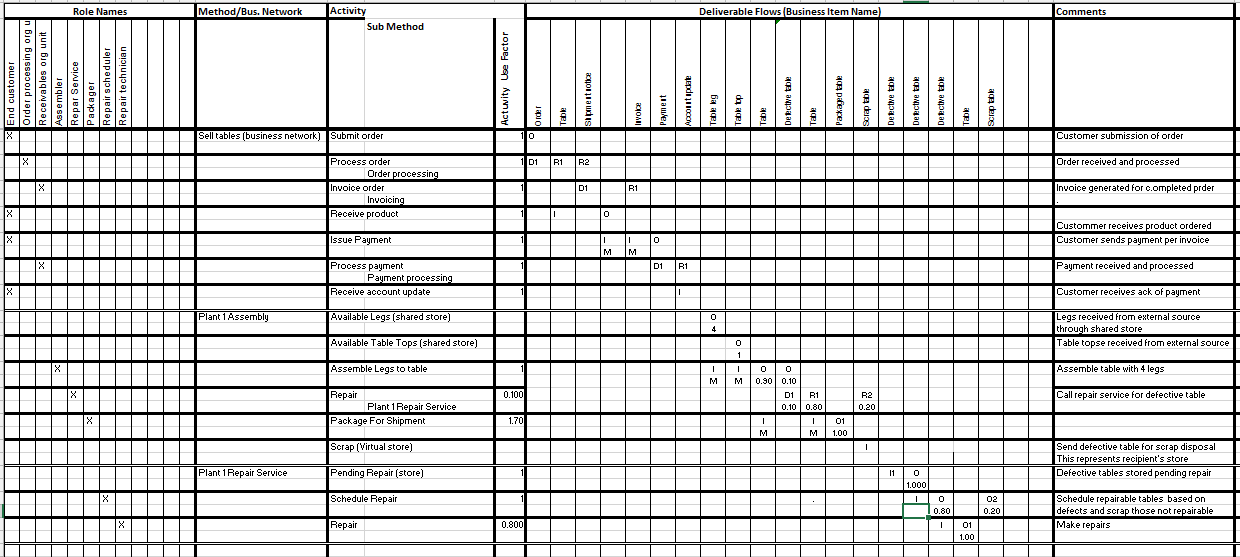
* Activities
* Business Items
* Role Definitions
* Capabilities
* Organizations
* Value Type Definitions
* Measurements
* Value Propositions
* Objective

## Activities

An Activities table, in the Activities and Business Items spreadsheet, is depicted in Figure 1. It captures information about sequences of activities within collaborations. Each activity has two rows. The purpose of the two rows depends on the intersecting columns, discussed below.

The Activities table captures the basic structure of the model. It identifies (1) capability methods and business networks (both are collaborations), (2) the activities within each collaboration, (3) the activities that delegate to other capability methods that are engaged through delegation, and (4) the flow of deliverables between activities, stores and collaborations.

The content of the Activities table is repeated in the Measurements table, but the Measurements table is more complex. The Activities table is easier to use and to observe the flows, and it is sufficient for capability analysis as discussed in the later section on Applications of the Spreadsheet Model.



**Figure 1, Activities Table**

Although the VDML model would allow any collaboration to have activities, the spreadsheet model only addresses activities of capability methods and business networks. Capability methods are abstractions of business processes, and business networks are abstractions of exchanges between independent business parties.

For ease of use, the activities should appear in the order in which they are performed. However, the order will vary when representing concurrently executing threads, and the delegation of activities to sub-collaborations. The actual sequence of execution is determined by input and output deliverable flows, so the sequence of activities in the table is for ease of use, and is not required for proper specification as long as activities are shown within the rows of their containing collaboration.

The activities table has four major columns: Roles, Capability Method/Business Network Name, Activity, and Deliverable Flows. These are discussed in the paragraphs that follow.

### Roles

The Roles group consists of one column per role with vertically aligned names in the heading. An “X” appears in the role column for the first row of each activity (two rows per activity) in which the role is performer.

Note that a role may perform multiple activities within a collaboration, and each activity may require different capabilities from the same performer (i.e., the same person or organization).

Roles are defined in the context of a collaboration as identified in the Capability Method/Business Network name column. Role names need not be unique within a model, so the name in the role column may be reused to reference roles in different collaborations. Additional role information is captured in the Role Definition table which encourages the use of a role name to have consistent meaning across the different collaborations in which it is used. Sometimes when the same role name is used, it is intended that the role in one collaboration should be assigned to the same participant in a sub-collaboration (such as a doctor participating in different collaborations for the same patient. This should be indicated in the Roles table, discussed later.

### Capability Method/Business Network

An entry in the Capability Method/Business Network (collaboration) name column defines a specific collaboration as the context for the associated group of activities and roles. Collaboration groups are separated by a double-line border. Most often, the collaboration name defines a capability method (a capability offered by an organization). A collaboration name along with an activity name uniquely identifies each activity (activity names need only be unique within a collaboration).

Business networks should have “(business network)” after the name. The activities of a business network will all engage sub-methods by delegation. These sub-methods are not detailed in the example. In general, the sub-methods of external partners will not be known. This is illustrated in the example, so the customer sub-methods are not included in the example.

### Activity

The Activity group includes columns for Activity name, Sub-Method, and Activity Use Factor.

The Activity column lists activities by name allowing two rows for each activity with the name in the first row. Activity rows occur within the group of rows headed with the name of the containing Method/Business Network (collaboration).

Where an activity engages a sub-collaboration, the second row of that Activity pair, is indented with the second column and contains the name of the sub-method (a capability method identified in the Capabilities table). The sub-collaboration detail occurs elsewhere in the Activities table. See the activity “Repair” of the “Plant 1 Assembly” method in the example, which delegates to the “Repair Service” method.

Note that an activity is typically enabled by the execution of the containing collaboration and occurrence of an input deliverable (if any). An input deliverable occurrence is a function of the execution of a deliverable source activity (where the deliverable flow is output) or availability of a business item in a supplying store. The deliverable planning percentage for that source activity determines the frequency of occurrence of the deliverable. Within a collaboration, all activities that do not have input deliverable flows are enabled when the collaboration is engaged.

The Activity Use Factor column specifies the fraction of the unit of production of the collaboration for which the associated activity is executed. If the activity has no inputs, the activity use factor defaults to 1.0 (once each time the collaboration is executed). Otherwise, the use factors of inputs will determine the activity use factor.

Stores are represented as activities in this table. A store represents a business item holding facility such as a bin, a warehouse or input queue. Stores are also important for synchronizing exchanges between independent collaborations. A collaboration may have “side-effect” output(s) that are in addition to any delegation-return outputs. Side-effect outputs to be delivered to another collaboration(s) are passed through a shared store. The sending collaboration has a virtual representation of the store. There may be multiple sources of input to a shared store (i.e., multiple virtual stores in different collaborations). Conversely, there may be multiple consumers from a shared store, thus each consumer has a virtual store representing the shared store as its source.

The store name should include “(shared store)” or “(virtual store)”. A store may appear as a virtual store in a recipient or sender collaboration. Thus the store name must be unique within the VDML model—all virtual stores for that store will refer to that name. There must in fact be only one concrete occurrence of the shared store that actually receives and sends deliverable flows, and all deliverables associated with that shared store must be the same business item type. Typically we will assume that the units received will equal the units consumed, and it may be unnecessary to explicitly make the occurrence of the shared store.

A virtual store should represent the shared store in a (double) row for each of the sharing collaborations. The “shared store” may appear in a receiving collaboration if that is the only receiving collaboration, otherwise each receiving store should be identified as a virtual store.

A unit of production received or provided by a store is based on the unit of measure applied to the stored business item. The inputs to a shared store may have a different unit of production from the receiving collaboration. In the example, the available legs store can be assumed to receive each leg as a unit of production but four legs are used as a unit of production for table assembly. The “resource use” of 4 for table legs is represented as the Activity Use Factor of the store.

### Deliverable Flows

The Deliverable Flows group contains one column for each deliverable flow and thus has two cells (two rows) for each activity-deliverable flow intersection. The name associated with the deliverable flow is the name of the business item type conveyed by the deliverable flow. In this table, deliverable flow is uniquely identified by the business item name, the source activity and the receiving activity. A business item may be conveyed by multiple deliverable flows such as where a sequence of activities each contribute value to the business item. In the table assembly example, the “table” business item occurs in multiple deliverable flows.

The top cell of the activity-deliverable flow intersection indicates if the deliverable flow is an input (designated with an “***I***”) or output (designated with an “***O***”) for that activity. The lower cell for an input is the input mode: an “***M***” for a merged input or “***E***” for an alternative/exclusive input. The lower cell for an output indicates the “planning percentage” for the outputs.

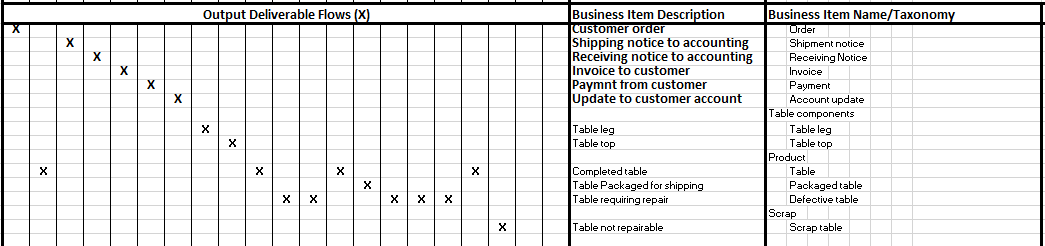
For delegation, an input to a delegating activity is indicated with a “**Dn**”, and the output of a returned deliverable is designated with an “***Rm***” where ***n*** is the number of the delegation input flow and ***m*** is the number of the delegation output flow. The numbers, ***n*** and ***m***, match numbers of the input and output flows of the sub-collaboration/method. So “Repair Method” has input “***I1***” and outputs “***O1***” and “***O2***”.

The bottom cell of the activity-deliverable flow intersection is the planning percentage of that activity output. The measurement in the cell is the percent of the activity executions for which the associated deliverable is output. This will impact the execution of the deliverable receiving activity and may affect the computation of its value measurements. See the section on Measurements for the effects of planning percentages. Activities are linked by deliverable flows. Input deliverable flow(s) enable/constrain the execution of an activity.

## Business Item Definitions

Business items are things that may be created, conveyed, stored, modified, consumed, etc. They include information records/messages/forms as well as physical things. Business items are the subject matter of activities. Deliverables are business items associated with deliverable flows. A business item table can be a distinct spreadsheet, but it is convenient to create the table as an extension to the Activities spreadsheet in order to reuse the deliverables columns for cross-reference.

The Business Item Definitions table. Figure 2, uses the Deliverable Flows and Comments columns of the Activities table to reference the activity deliverables and to contain a business item description (in place of Activity comments of the above table) An additional group of columns are provided for an indented list of business item taxonomy categories and business item names.



**Figure 2, Business Item Definitions**

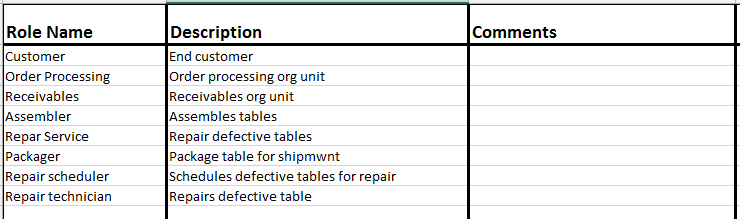
Uses of a business item in one or more deliverable flows is indicated in the intersection of the deliverable flow columns and the business item row. Deliverable flows may be “tangible” or “intangible” as defined in Value Network Analysis, see <http://www.vernaallee.com/images/VAA-VNAandValueConversionJIT.pdf> . A deliverable flow is intangible if it is not formally defined for the process. This typically represents an informal exchange of information. A tangible, formally defined deliverable flow, can be indicated by a “***X***” in the intersection of the deliverable flow column and the business item row. Intangible deliverable flows are the exception, and can be indicated by an “***I***” in that intersection.

Note that the same business item may occur in multiple deliverable flows. In the example, the “Assemble legs to table” activity produces a table, and the “Table business item is associated with the output deliverable flow and subsequent deliverable flows where the Table is input and output.

## Role Definitions

The Role Definitions table, depicted in Figure 3, contains one row for each role definition. The first column contains the role names, and the second column contains a description of each role. A third column is provided for additional comments.

In the Activities table, roles are assigned in the context of an associated collaboration, but role names need not be unique to a collaboration. A role name may be used in multiple collaborations but all uses should be consistent with the definition to avoid confusion. A role assignment (the participant in a role) is specific to each occurrence/execution of a collaboration.

**Figure 3, Role Definitions Table**

When the participant assigned to a role must be the same in a sub-collaborations, the requirement should be noted in the comments column of the role definition, and the role name should be used consistently in that manner. For example, the same primary care physician should be assigned to various treatment methods for a patient.

## Capabilities

The Capabilities table, the upper part of Figure 4, defines capabilities and associated methods, and it links the methods to the responsible organizations in the lower Organization table.

The first major column contains the capability hierarchy. The multiple minor columns support a hierarchical display.

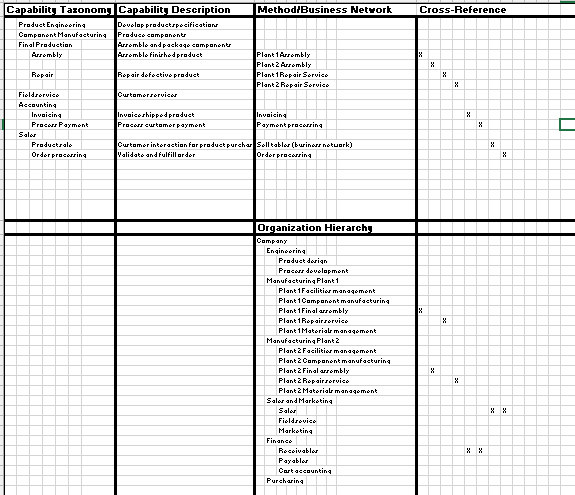
The second major column contains descriptions of the capabilities and capability categories.

The third major column defines the Method/Business Network (collaboration) names. It has multiple minor columns that are unused for capabilities but support the Organization table, below. Each collaboration name must be unique within the VDML model for reference by calling activities.

The last major column provides a cross-reference between methods/business networks, and organization units. The Capabilities table (upper) cross-reference grid flags cells on a diagonal to redirect a reference from a method/business network to an organization flag in the lower table.

## Organization

The Organization table, in the lower portion of the spreadsheet, contains the organization hierarchy in its first major column. The multiple minor columns provide for an indented list.



**Figure 4, Cross-Referenced Capabilities & Organizations Tables (Capabilities and Organizations Spreadsheet)**

The organization table has only two major columns. The Organization Hierarchy uses the multiple minor columns for indentation. To support references to specific organizations, each organization name must be unique within the VDML model.

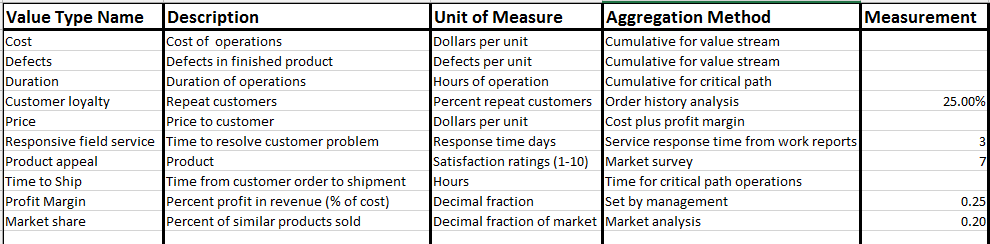
The second major column provides the cross-reference between organizations and methods/business networks. In a larger model, each organization unit will likely be responsible for multiple methods.

An organization may offer a capability without a formal method. Application of such capabilities may involve ad hoc, adaptive processes that are not formally defined. These must also have a unique method name to represent the organization’s capability and for delegation to that organization’s capability.

This composite table of methods and organizations facilitates consideration of re-alignment of capabilities and organizations.

## Value types

The Value Types table, Figure 5, defines names of value types. The first column contains the value type names. The second column is a description for each value type. Value types include all measurements of performance even though some may not be included as customer values. The third column identifies the unit of measure used for the value type, and the fourth column, Aggregation Method, is a place to capture notes on computation of the cumulative measurements. For example, costs are usually simply added. On the other hand, duration will require a weighted average by use factor from exclusive inputs or selection of the longest duration for matched inputs.

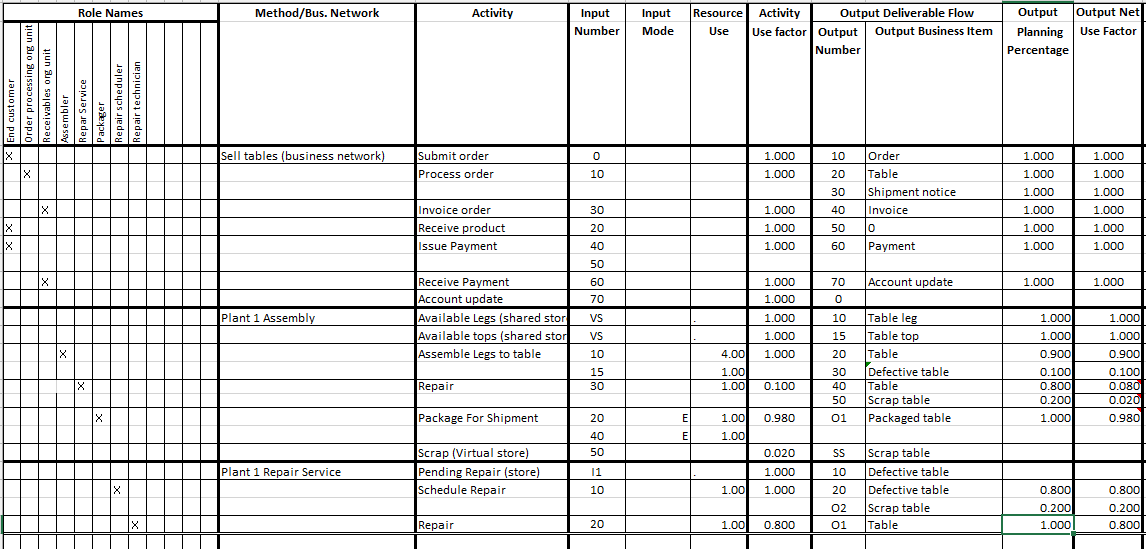


**Figure 5, Value Types Table**

The last column provides for user-assigned measurements. These are measurements that are not supported by the detail of the business model. In the example, customer loyalty, field service response time, product appeal, profit margin and market share are assigned values. Some of these may be defined by management, some are the results of research, some are computed from other measurements and some might instead be defined by the model if it were more robust. These are used as input to a value proposition. For example, price in a value proposition is computed as profit margin applied to the sum of costs.

## Measurements

The Measurements table, Figures 6a and 6b, below, performs computations to accumulate activity contributions and provide the net effect of each value type relevant to a value proposition or an objective. Figures 6a and 6b, are two halves of the table to accommodate the table width.

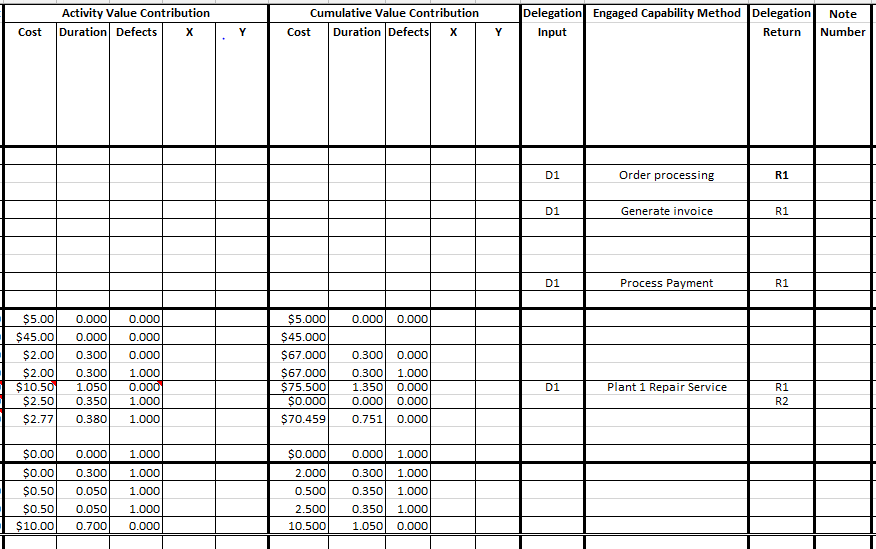
**Figure 6a, Measurements Table**

The Measurements table is the most complex. It includes all of the information captured in the Activities table plus value measurements. Users may prefer to use the Activities table for initial data capture or if they do not intend to do value analysis. In the subsections that follow we will discuss the format of the Measurements table followed by additional subsections that discuss an example cost computation, value measurements in delegation, and value measurements for shared stores.

Each activity occupies one or more rows in this table depending on the larger number of input or output deliverables.

### Role Names

This column group is a repeat of the Role Names columns of the Activities table. This is included so that all of the elements of the Activities table are included in the Measurements table for a comprehensive view. If the Activities table is included in a model, then this column group might be omitted from the Measurements table. However, note that the Activities table is still needed to support the Business Item table on the same spreadsheet assuming the user wants the mapping of all deliverable flows to each business item.

**Figure 6b, Measurements Table**

In the Measurements table, since an activity may have one or more rows, the role column for an activity is tagged in the first row of the associated activity.

### Method/Business Network

The spreadsheet-based VDML model represents activities in business networks and capability methods (both are collaborations). This column contains the name of each of these collaborations. The Method/Business Net name identifies the first row associated with that collaboration. The associated rows represent activities and their inputs and outputs. The rows associated with a collaboration are bounded by bold borders.

### Activity Name

Each activity or store is identified by a name that is unique within the collaboration. A shared store must have a name that is unique within the VDML model because it is referenced by multiple sending and/or receiving collaborations.

An activity (or store) has a number of rows that is the greater of the number of inputs or the number of outputs. The activity (or store) name appears in the first row. The activity rows are separated by thin borders.

### Input number

The Input Number identifies the input deliverable flow by reference to the output number associated with an output deliverable flow within the same collaboration.

If the input comes from a delegating activity, then the input number is ***In*** where ***n*** is the input flow identifier referenced by the delegating activity to identify a delegation input flow. If the activity delegates, then linkage is defined in the Delegation column group at the right end of the table.

In the example, the Available Legs store is shown as a shared store on the assumption that legs may be delivered from multiple sources and or from collaborations with a different unit of production (production of one leg rather than the four required for one table). A shared store receives inputs from virtual stores—representation(s) of that shared store in other collaboration(s). The shared store input number (source) is identified as ***VS*** (for “virtual store) indicating that there is one or more virtual store(s) in other collaborations that provide input to this store.

On the other hand, the Pending Repair store in the Repair Service is not shown as shared or virtual since its input is direct from delegation and the store is used only for holding tables in that collaboration, pending repair. Its input number is “I1”.

### Input Mode

The input mode column indicates if the input is to be matched with other input(s) (indicated with an “M”) or received as alternative/exclusive input(s) (indicated with an “E”). If there are some of each, then the matches should be resolved first. Most activites in the example have only one input—Packing for Shipment is the exception.

If the use factors of the matched inputs (M mode) are not equal, then the input with the largest use factor dominates. The minor input business items are implicitly applied to only some of the dominant input business items. Matching does not compute a new activity use factor; the activity use factor is the same as the dominant input (the production volume is the same). For example, one body and one chassis equals one car.

Exclusive (E) mode results in an activity use factor that is the sum of the input use factors. The activity is executed whenever any one of the inputs is received.

Use factors determine the weights given to the value measurements associated with each of the inputs. Additional variations in the translation of inputs to outputs may have additional effects on the aggregation of value measurements. These are left to the user to discover and resolve.

### Activity Use Factor

The next column shows the activity use factor. This indicates the fraction of executions of the containing collaboration during which the activity is executed. One execution of a collaboration is a unit of production in that context.

The activity use-factor is 1.0 (for each execution of the collaboration) if an activity is not dependent on any inputs. If an activity has inputs then the fraction is determined based on the use factors of the inputs that enable the activity execution. For a simple input this is the net use factor of the incoming deliverable flow. If there are multiple input deliverables, the activity use factor is determined as described for the Input Mode column, above.

In the example, repaired tables that flow with 0.08 use factor are combined with good tables with 0.9 use factor. The net activity use factor is 0.98 (the sum). However, the value measurements for the combined tables are computed as a weighted average since the repaired table values have a smaller impact on the resulting statistical value measurements.

In the example, the consumption of table legs requires 4 legs to be consumed for each execution of the collaboration. In the full VDML model, this would be accomplished through the Resource Use element associated with the receiving activity. The spreadsheet model does not support the features of the Resource Use element, so instead, the store is represented as being executed 4 times for one execution of the collaboration (Activity Use Factor). Thus each execution outputs one table leg, Note the further implications under the Output Net Use Factor discussion, below.

### Output Deliverable Flow

Output Deliverable Flow has two columns: the Output Number and the Output Business Item. The Output Number identifies an output deliverable flow for the associated collaboration. The output number is unique within the collaboration. The Business Item identifies the type of deliverable conveyed. An incoming deliverable flow is identified by an Input Number that refers to the output number of a deliverable flow in the same collaboration.

If an output deliverable flow is returned to a delegating activity, then its output number should be ***Om*** where ***m*** is the return number of the collaboration. For the activity that delegates, the output number is ***Rm*** to indicate the return flow from the delegation. The input and output numbers for the delegation are defined in the Delegation column group on the right side of the table.

### Planning Percentage

The output planning percentage indicates the fraction of production of the associated activity for which the associated output is produced. In the example, the output of the Plant 1 Assembly operation is 90% good tables and 10% defective tables.

Note that the planning percentages for alternative outputs will normally add to 1.0, representing all outputs of the activity. It will be common for the outputs to have different planning percentages and different business items.

### Output Net Use Factor

The output net use factor for each deliverable flow is computed as the product of the activity use factor and the deliverable flow planning percentage. This then is the fraction of a unit of production of the collaboration that is directed to an output. One unit of production represents one execution of the collaboration.

In the example, the store of table lets is shown with an activity use factor of 4. While the planning percentage for each execution is 1, the output net use factor becomes 4 (Planning Percentage times the Activity Use Factor).

### Activity Value Contributions

The Activity Value Contributions express the value contribution measurements of the activity for each of its outputs and associated values. The number of sub-columns will vary depending on the number of associated values of interest. In the example, Cost, Duration and Defects are shown as typical. X and Y represent other possible values.

The measurements in these cells represent the value contribution of the associated activity per unit of production. The measurements are based on one unit of production in the collaboration context. So while an activity may have a use factor of 0.5, the value measurements are still per unit of production. The use factors only determine how combinations of value measurements are weighted when combined, such as the weighted average cost of alternative inputs that occur for the Package for Shipment activity.

Some output flows may carry some of the value measurements and some may carry some of the same or other value measurements. This is the case in the example where the Repair activity outputs repaired tables and scrap tables. The costs of all input, defective tables are associated with the repaired tables while the scrap tables are all defective—defects have been eliminated from the repaired tables. Cost and duration are not relevant to the scrap tables but do not affect the good production.

### Cumulative Value Contributions

The Cumulative Value Contributions for each output express the value contribution measurements of associated values aggregated with the value measurements from the activity inputs. These are the value measurements carried forward as input to the subsequent receiving activity.

The cumulative value contribution for each relevant value type and associated output deliverable is computed from (1) the value measurement(s) received with the activity input(s) for previous activities (aggregated if there are multiple inputs), and (2) the value contributions associated with the output of the current activity. The incoming value measurements are taken from the cumulative value contributions at the source(s) of the input(s) to this activity. Note that the method of aggregation of value contribution measurements depends on the type of value.

In the example, each execution of the table leg store produces one table leg. The store contributes zero value in the Activity Value Contribution columns, but the Cumulative Value Contribution cost shows $5.00 which is thus the cost of the input(s} to the store for one leg. When the legs are input to the Assemble activity, the Output Net Use Factor is applied to the incoming value measurements, so the cost of legs is 4 times $5.00. In a model that implemented the inputs to the Available Legs store, there would be one or more virtual stores in other collaborations that receive inputs to the Available Legs store. Each of these virtual stores might receive different quantities with different value measurements. The modeler would need to compute the incoming value measurements from the appropriate aggregations of the value measurements to determine the Cumulative Value Contributions measurements in the example.

### Delegation Group

The last three columns address delegation from the activity of the same row to a sub-method. These are placed at the far right of the table because they are only relevant to activities that delegate.

The first column indicates which input of the sub-collaboration receives the input from the associated input row of the delegating activity. These are indicated with ***Dn***, where ***n*** is the input number of the sub-collaboration.

The second column is the name of the sub-collaboration.

The third column indicates which output of the sub-collaboration is returned to the activity as the output in that row of the delegating activity. These are indicated with ***Rm***, where ***m*** is the output number determined by the sub-collaboration.

The deliverable rows for the called collaboration will identify the inputs and outputs as ***In*** and ***Om*** in the input and output number columns.

Input value measurements are not passed by delegation. The value contributions of the engaged method are aggregated and provided with the return deliverable flows. These returned value measurements are then aggregated with the value measurements of the inputs to the delegating activity.

### Notes

A Notes column is included for clarification of computations, particularly the computation and/or assumptions associated with the impact of multiple inputs or outputs of an activity. It is suggested that a note number can appear in this column as a reference to a lengthy note entered elsewhere in the spreadsheet.

### Example cost measurement computations

This subsection further illustrates the computation of value measurements by tracing the development of the cost measurement through the table assembly example.

*Assemble Legs to Table* is the first activity in this value stream fragment that contributes cost. The activity use factor is 1.0—it is executed once for each unit of production of the containing capability method. It produces 90% good tables, and 10% defective tables—two outputs. The activity contributes the same cost per unit to both outputs--$2.00 per unit. For the cumulative cost for the Assemble deliverable, the activity contribution cost is added to the costs of legs (4 \* $5) and a table top ($45).

The defective tables are delivered to the *Repair* activity which delegates to the *Plant 1 Repair Service*. The repair service starts by scheduling tables for repair and in the process identifies 20% as unrepairable scrap. The cost per unit is $0.50. The scrap tables are one of the returns (O2) from the *Plant 1 Repair Service*. The repairable tables are directed to the *Repair* activity. The repair activity contributes $10.00 for the cost of repair of 80% of the tables sent to the *Plant 1 Repair Service*. This is added to the incoming cost per unit of $0.50 for a total cost of $10.50 per unit for repair of the repairable tables. Note that this cost is the cost contributed within the *Plant 1 Repair Service*, and does not include costs accumulated before the activity that called the *Plant 1 Repair Service*.

The repaired tables and the scrap tables are returned to the *Repair* activity of the *Plant 1 Assembly* method. The delegating activity takes the value measurements contributed by the called method as its own and aggregates them with the incoming measurements for good tables. Thus the activity value contribution is $10.50 for the repaired tables and $2.50 for the scrap tables. The previously defective tables represent 0.1 units of production in the collaboration context. Therefore the repaired tables represent 0.1 \* 0.8 = 0.08 units of production, and the scrap tables represent 0.1 \* 0.2 = 0.02 units of production.

The contributions of the Repair activity (received from the delegation) are added to the cumulative cost received by the Repair activity defective tables input. However, the costs of tables that became scrap should be considered part of the cost of production of the repaired tables. Consequently, the cost of the scrap tables are allocated to the cost of the repaired tables. A cost of 0.02 units of production are allocated to 0.08 units of production, so a 0.02/0.08 of the unit of production cost of the scrap tables is added to the repaired tables cost as part of the cumulative cost of production.

The repaired tables along with the good tables from the original assembly operation are inputs to the Packaging activity. These inputs are the same business items but have different costs associated with them as defined by the cumulative value contributions of the deliverable flows from their respective source activities. The original good tables represent 90% of the original unit of production of the containing collaboration, and the repaired tables represent 8% of the original unit of production. These costs are combined as weighted averages: (0.90 \* $67.00 + 0.08 \* $79.50)/0.980 since the output is a mixture of good and repaired units of production.

The packaging activity contribution of $2.77 applies to all of the output units of production, but the packaging activity only produces 98% of the original units of production of the *Plant 1 Assembly* method. 2% of the production has been lost resulting in an increase in the overall cost of a unit of production.

The user may consider alternative ways of dealing with this shortage. If the product is fungible, then this means that production schedules must be increased to cover the losses of production. If the product is specific to a customer order, then the orders for scrapped products must be re-submitted and the delivery time for those units will be delayed for the duplicated operations to re-produce the orders.

Furthermore, the scrap tables are likely delivered to a salvage operation so that some of the loss is recovered. This cost recovery should be incorporated in the final cost per unit of the product.

Note that the value measurements are always measurements per unit of production regardless of the fractional units of production of an activity or an output deliverable flow. The use factors are important for determining the relative impact of different deliverable flows on the statistical value measurements of the end product.

### Value aggregations in delegation

Input deliverable flows to a delegating activity are redirected to its sub-collaboration. The sub-collaboration is executed for each execution of the delegating activity. While the delegating activity may be performed for only a portion of the units of production of its collaboration, the sub-collaboration operates on the basis that each execution of the sub-collaboration produces its unit of production.

The cumulative value contributions of inputs to the delegating collaboration are not forwarded with the flow to the sub-collaboration. Instead, these are aggregated into the returned result of contributions by the sub-collaboration activities. The sub-collaboration may return multiple outputs with different business items and value measurements. The value measurements input to the delegating activity are then aggregated, probably selectively, to the multiple outputs.

In some cases, the deliverables returned from a delegation will represent less than a unit of production of the sub-collaboration. The use factor of the delegating activity outputshould be adjusted by multiplying the activity use factor by the returned fractional unit of production. This represents the reduced impact of this output if later combined with an alternate production thread.

In the example, not all of the tables delegated for repair are returned as repaired tables from the repair service. The fractional unit of production for repaired tables will bear the value impact of repair as input to a weighted average when combined with the measurements for tables that were not repaired. For example, the accumulated costs of the scrap tables is allocated as a cost of production of the repaired tables.

A collaboration engaged through delegation may also produce outputs that are not returned to the delegating activity. Such an output is described as a side-effect output that is directed to a virtual store representing a shared store that provides input to one or more other collaborations.

A side effect output may convey some of the value measurements that were input to the delegating activity. While the sub-collaboration does not incorporate these value measurements into the return output(s), it may be appropriate to incorporate them into the side-effect output(s), particularly if the side-effect output is the same business item as either the delegating activity input or the delegation return output.

### Value aggregations for shared stores

The primary purpose of a shared store is to provide an asynchronous flow from one collaboration to another. This represents a deliverable flow that is not returned from delegation, but represents a “side-effect” flow from the collaboration. The side-effect deliverable flow is directed to a virtual store representing a shared store from which the target collaboration(s) receives the deliverable.

A shared store may have inputs from multiple sources, including side-effects from different uses (i.e., different delegations) of a shared collaboration. Each of these sources may contribute different value measurement sets.

The side-effect deliverables of a shared collaboration may be produced by that collaboration in multiple contexts, i.e., delegations from different activities that each may have different incoming value contributions, so from the shared store perspective, these are all different sources.

The suggested approximation is to incorporate the input measurements from all of the delegating activities of the different contexts weighted by their use factors (Note that these use factors are local to each collaboration and may not be the same as their use factors a global context—context of the shared store). Depending on the nature of the measurement, this may mean computing weighted averages for the alternative measurements for each value type.

A store may have an output deliverable flow on demand. The store may introduce a delay representing a wait for response to a request, or, if the business items are uniquely identified, then the delay may be time in a queue.

In this example, a store may hold legs for assembly to a table. The unit of production of legs is one leg. The leg production feeds the table assembly store that consumes 4 legs for a table unit of production (input deliverable flow resource use = 4). The measurements associated with the inputs must be translated to their impact on the product of the consuming activity.

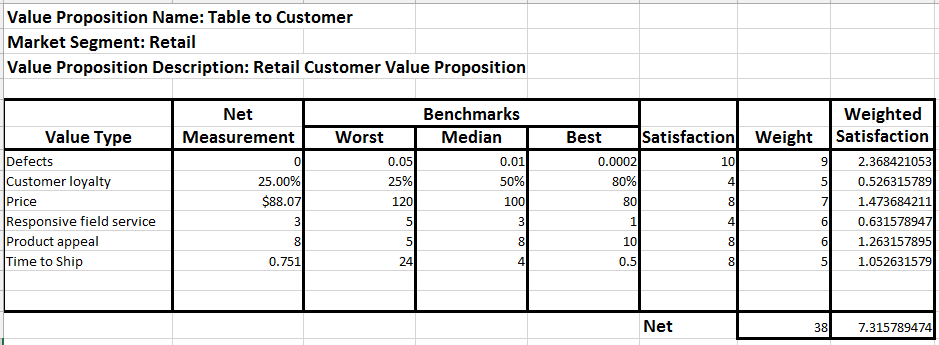
## Value propositions

A Value Proposition spreadsheet, Figure 7, may have multiple value propositions as separate groups of rows. Each value proposition should represent a different market segment and/or a different market situation.

For each value proposition, there is one row for each relevant customer value and one row for overall satisfaction.

The first column identifies the value type for that row. The next column contains the net value measurement. This net value measurement may be (1) taken from the final deliverable flow cumulative value contribution, (2) taken from an assigned measurement for the value type on the Value Types table, or (3) computed from multiple value measurements.

Some value contributions may not come from the value stream within the scope of the current VDML model. For example, product design features and product performance measures may be contributed by product development activities that are not modeled. Also, contributions such as customer loyalty or field support may not be modeled but may be important for the overall value proposition. These value measurements may be entered directly into the designated column of the Value Types table and used as input to the value proposition.

**Figure 7, Value Proposition Table**

The Benchmarks columns support an approach to determining a satisfaction measurement for the next column. The Median column should have a measurement that represents the median value for the value type within the market. The Worst and Best columns represent the extremes of the marketplace. A value measurement near the median could be interpreted as ambivalent satisfaction. A measurement near the Worst, could be interpreted as poor, and a measurement near the Best could be interpreted as good. A measurement below Worst could be viewed as unacceptable, and a measurement above Best could be interpreted as excellent.

The Satisfaction column is a rating from 1 to 10. It is computed from the net value based on customer expectations and priorities—possibly a different computation for each value type.

For example, satisfaction might be computed as a straight-line transformation from the benchmark measurements as follows:

S = (6(V-W)/B-W)+2

Where

S = The computed satisfaction level

V = The net value measurement

W = The worst benchmark measurement

B = The best benchmark measurement

However, if the net value measurement is either very good or very bad, the satisfaction measurement could be greater than 10 or less than 1.

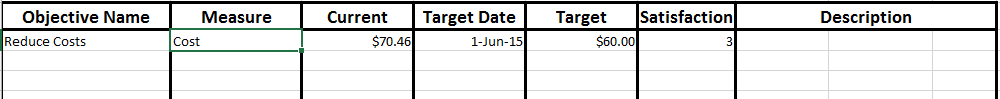
This formula aligns the worst benchmark with a satisfaction of 2 and the best benchmark with a satisfaction of 8.

The Weight column represents customer priorities for each value type. It is useful, but not necessary, for the weights to add to 1.0. The Weighted Satisfaction column is computed from the satisfaction measurements for each value, multiplied by the weight for that value and divided by the total of the weights for all of the values of that value proposition. The weighted overall satisfaction measurement is the total of the individual weighted value measurements for the individual values.

Different value propositions may be defined to address different markets where different values are relevant and different weights are assigned.

## Objectives

In various applications, it is desirable to identify objectives, set target values and monitor progress toward the objective target. These may be included as an Objective Measurements table in the Value Proposition spreadsheet as indicated in Figure 8, below.

**Figure 8, Objective Measurements Table**

An objective may be associated with a customer value, an internal performance measurement, the performance of a particular activity or capability, etc. It is up to the user to identify the measurement of interest.

The first column contains the Objective Name. The objective Measure defines the aspect being measured. The Current column contains a value measurement representing current performance. The Target Date is when the objective is expected to be achieved, and the Target column represents the desired measurement. The Satisfaction column is an indication of progress from a management perspective. The satisfaction computation could be based on the satisfaction computation used in the value proposition if the value is one that is reported for customers. The last column is for additional description of the objective that should include the source of the measurement.

If the associated VDML business model represents the current state of the business, then the Current cell may represent a measurement derived from the business model. However, if the business model represents a To-Be future state, then the target cell should be derived from measurements in the model.

The purpose of this table is to link objectives to the VDML model and the elements that can impact the measurement of the objective, thus supporting consideration of feasibility and responsibility. Supporting, more detailed objectives might target improvement contributions to particular capability methods and their performing organization units.

# Applications of a Spreadsheet-based Model

There are two general applications of the spreadsheet based VDML model: capability analysis and value creation analysis.

## Capability Analysis

The purpose of these applications is to analyze gaps or similarities of capabilities. This analysis may be performed without the Measurements table since the essential elements of the Measurements table are included in the Activities table.

The following are examples of business applications for a capability analysis.

* **Identification of opportunities for capability consolidation**. The model will help expose redundant or similar capabilities. These must then be analyzed to determine significant differences, determine if these differences are essential to the different contexts in which the capability is used, and determine the feasibility of these differences being accommodated by a consolidated implementation.
* **Analysis of the implications of outsourcing a capability**. The model will identify the primary inputs and outputs to the capability implementation that must be addressed if it is outsourced. Furthermore, the model will help identify capabilities that depend on scarce or exclusive resources that are the basis of competitive or differentiating market advantage.
* **Merger or acquisition analysis for potential synergy and economies of scale**. Value stream and capability models must be developed for current value streams that may be affected. These are then compared to models of corresponding value streams and capabilities of the business to be merged. This analysis will help identify where the two businesses are similar or different for consideration of potential consolidations. In addition to providing insight on the feasibility of the merger, the analysis can provide the basis for transformation after the merger.
* **Consideration of new product/business capability and investment requirements**. A value stream model can be developed for a potential business. The capabilities required can then be analyzed to determine if they can be addressed with existing capabilities or if new capabilities must be acquired. This is then input to development of implementation requirements.
* **Restructuring of the organization**. The model provides an understanding of the resource requirements of capabilities and the organizations responsible for providing each capability. This supports consideration of organizational restructuring to consolidate management of resources and enhance skills through specialization,

## Value Contribution Analysis Modeling

The purpose of this analysis is to identify and evaluate business changes based on the sources of value contributions and their impact on customer satisfaction.

The following are examples of applications of value creation analysis:

* **Identification of key capabilities for customer satisfaction improvement**. Value propositions can be analyzed to identify key customer values. The sources of value creation for these key values can be identified to determine where investments could improve customer satisfaction. These potential improvements can then be evaluated along with the costs of improvements to set investment priorities.
* **Analysis and design of to-be business for implementation of a business strategy**. Alternative implementations of a business strategy can be evaluated for cost and impact on customer value. The selected alternative can then be used as the basis for transformation planning, potentially using models for each phase of a transition while considering the impact on customer value at each phase and using the phase models and to-be model for planning work and setting objectives in terms of target value measurements.
* **More in-depth analysis of some of the capability analysis applications**. Each of the applications of capability analysis can be considered in greater depth using value contribution analysis.