Analysis of the Relationships between VDML and BPMN

This document was prepared by Fred Cummins and Henk de Man in response to questions regarding the relationship between VDML (Value Delivery Modeling Language) and BPMN (Business Process Model and Notation). It presents an analysis of the relationship between VDML and BPMN. This mapping as well as the transformations and possible BPMN extensions are not specifications but are a basis for feasibility analysis and further development by VDML implementers.

The mapping reveals the similarities and differences between BPMN and VDML. While both BPMN and VDML represent business processes, VDML is at a higher level of abstraction. The difference in viewpoints might be compared to the differences between a circuit diagram for an electronic appliance, and a component assembly drawing for the same appliance. While they represent the same appliance and have some common elements, you cannot derive one from the other.

Some fundamental differences in addition to level of abstraction:

- BPMN does not address organizations, including ad hoc and informal collaborations that are essential to the operation of the business and determine responsibility for resources and business operations.
- BPMN does not address business capabilities and their relationships to processes and organizations.
- BPMN does not address either the creation of value or the exchange of value with customers or other recipients or providers.
- BPMN focuses on the flow of control for execution of activities—resources and deliverables are, for the most part, implicit. VDML focuses on statistical measurements of the performance of activities and the management and flow of resources and deliverables that are consumed and produced by activities—flow of control, for the most part, is implicit.
- BPMN has multiple models for participant interactions and flow of control (process, collaboration, choreography, conversation). VDML has one paradigm for the interaction of participants and flow of deliverables (collaboration of roles performing activities linked by deliverable flows to create value).
- BPMN defines roles in Processes; VDML supports roles in Collaborations with recursive roles within roles. For example, John participates in Role A and Role A participates in Role B, so John indirectly participates in Role B because he participates in Role A.
- BPMN provides operational detail of activities, decisions and events to deliver a desired result; VDML provides an abstraction of the work performed, including anapplication of a capability, to produce a deliverable and contribute value with measurements for multiple contexts and scenarios.
- BPMN represents formal actions and transfers of control; VDML represents both formal and informal actions and transfers of deliverables and other value contributions.

These differences account for most of the differences and ambiguities in the mappings presented in this paper. Some of these ambiguities may be resolved in implementations of transformations by tool vendors, but many will be left to the discretion of the individual user to determine the appropriate model in the particular circumstances.

The relationships between BPMN and VDML are presented in five sections that follow:

(1) The complementary roles of BPMN and VDML

(2) A tabular mapping of BPMN to VDML concepts,

(3) Transformation of a BPMN model to a VDML model,

(4) Transformation of a VDML model to a BPMN model,

(5) Feasibility of defining VDML as an extension of BPMN,

(6) Feasibility of implementing VDML and BPMN in the same tool

These are followed by a brief summary.

The Complementary Roles of BPMN and VDML

I this section we consider the complementary roles of BPMN and VDML. While they represent some similar concepts, they address different business problems and areas of concern. They provide different viewpoints; they address the concerns of different stakeholders.

The intent of VDML is to address the needs of business leaders to define, manage and transform the design of the enterprise. This requires a broad perspective to incorporate multiple aspects of the enterprise, such as the design of roles and collaborations. The focus of BPMN is defining and managing repeatable, reliable processes with an emphasis on automation. The focus is much more specific but much deeper in detail addressing many exceptions and variations. The community of concern is primarily managers, business analysts and systems developers closer to the business operations. In terms of the draft MDA Guide, VDML supports a business model and BPMN supports a logical system model.

The primary mode of application of VDML is “forward engineering” as described by the draft MDA Guide. This involves development of a business solution by starting at an abstract level of design and developing in stages of increasing detail. VDML enables business leaders to define and agree upon a high-level business design that can then be used to guide more detailed solutions that may be delegated to multiple business units. BPMN represents a primary tool for development of the next level of detail.

However, not all applications of VDML fit that pattern. The manner in which VDML is used and its relationship to BPMN will vary depending on the nature of the business problem being solved. A number of different modes are discussed in the following paragraphs.

Business network analysis

VDML Business Networks models can be developed to represent relationships with customers, suppliers and others, typically in the context of certain collaborative activities such as a line of business, product development relationships, regulatory compliance relationships, etc. The collaboration structure and exchange of deliverables helps analysis focus on both tangible and intangible deliverables and the less obvious but important exchanges. This may lead to changes in business relationships, clarification of roles and deliverables, and development of improved internal business processes.

Customer value analysis

Customer value analysis typically will be based on value stream analysis. A basic value proposition must be defined to identify values of concern to a customer or customer community. Then the value proposition must be linked to collaboration and activity value contributions. In the absence of an existing VDML model, development of the value stream will typically be performed top-down, considering high-level activities or stages of the business and breaking these down into a hierarchy of activities and deliverable flows. The values of a customer value proposition are traced back up the value stream to identify sources of value and potential improvements.   The level of detail will be driven by the level of confidence in the measurements and the need to drive down to specific capabilities and responsible organizations. This could start with BPMN processes, but typically this requires digging through too much detail. The higher level analysis offered by VDML offers a stepping stone that will ultimately lead to focused consideration of process improvements (BPMN).

The analysis of customer value will lead to identification of activities might be improved to improve customer value. This may also lead to a “capability map” that provides a visual breakdown of the capabilities required by the activities with highlights to identify those capabilities that need improvement.

Capability analysis

Capability analysis can go beyond the identification of critical capabilities in a value stream. In a large enterprise, multiple product lines or lines of business may require some of the same capabilities. Capability analysis with VDML provides a way to identify similar capabilities, consider ways they can cooperate to improve their performance, and assess the potential to consolidate for economies of scale. This may drive process improvements or development of processes for a shared capability to meet the needs of multiple lines of business.

When capabilities are consolidated as shared services, there is a need to understand their performance in multiple contexts. Changing a capability for one line of business could adversely affect another. In addition, analysis of investments in improvement of capabilities should consider how each capability affects all of its uses to provide an enterprise-level perspective on allocation of investments. The implementation of improvements will likely involve process improvements based on the capability analysis.

New product/LOB analysis

Development of business operations for a new product or line of business should involve leveraging existing capabilities. VDML supports modeling the business at a level of abstraction that identifies capability requirements without requiring a lot of detailed process analysis. Capability requirements can be developed at an appropriate level of detail for strategic planning, to identify existing capabilities and any changes to their requirements along with requirements for new capabilities. This provides a framework for more detailed analysis, such as process modeling, to focus on the areas the need development. Business leaders can more quickly assess both the cost and time required to realize the needed production capability and make an informed business decision on the potential success of the product.

Merger or acquisition

Mergers and acquisitions, if they are to realize any synergy, require some consolidation of capabilities. VDML models can be used during due diligence to get a better understanding of the similarities and differences of the enterprises and the potential for sharing capabilities. While the enterprises may have similar products, that does not mean they have similar ways of doing business.

Here a transformation of BPMN to VDML, assuming processes are specified with BPMN, may be an expeditious way to develop VDML models for comparison. If transformation is not possible, or if starting at business processes is too big an undertaking, VDML can still support value stream models at levels of detail appropriate to confirm similarities or expose differences so that much of the modeling does not reach the business-process level of detail. These models will provide the basis for consideration of to-be models to realize expectations of synergy in the merger.

Strategic transformation

Strategic transformation potentially involves substantial change to the way the business operates. Starting with business processes will bias the result by making it difficult to see the forest for the trees. VDML supports a higher level view of the business where sufficient detail can be developed to realize a meaningful assessment of the scope and duration of the change, the performance expectations and competitive position as well as the framework for organization of initiatives and development of business processes and systems. Multiple VDML models can be used to define stages of transformation and define incremental change rather than one, long-term undertaking with no benefit (or failure) until the end.

Accountability

VDML provides the linkage of customer value to contributing activities, to use of capabilities and to responsible organizations so that organizations can be held accountable for poor results and recognized for important improvements. VDML will drive meaningful performance measurement, and support analysis that starts from a high level of abstraction and expands levels of detail to focus on a specific area for improvement. It will also help clarify expectations and responsibility for shared capabilities.

Mapping BPMN Concepts to VDML Concepts

The different viewpoints represented by BPMN and VDML result in significantly different ways of viewing the operation of the enterprise. Not only are the models quite different, the users tend to associate different semantics with some of the same terms. The table below defines a mapping of BPMN concepts to VDML concepts. In this table class names and association names are broken with spaces for readability.

BPMN	VDML	Comment
Collaboration	Business Network	A BPMN Collaboration is restricted to the exchange of Messages between two or more business entities (Participants). The Participants may be "black-box" or Processes may be visible within them. A BPMN Collaboration involves one set of Participants. Subsets of interactions are defined with Conversations. A VDML Business Network (Collaboration) defines exchanges of all deliverables between business entities (VDML Parties), not just messages. The Business Network Deliverable Flows are the same as Deliverable Flows between internal Activities (same as for all Collaborations). Activities of Parties may perform work or delegate to supporting Collaborations of the Party, thus providing desired abstraction. A VDML Business Network can delegate to shared component Business Network(s). A VDML Business Network also exchanges defined Value Propositions that are an abstraction of the operational exchanges of value between the Parties, describing the net value effect of their relationships in the Business Network.
Process	Collaboration	A BPMN process contains a network of Flow Nodes (e.g., Activities, Gateways, Events) connected by Sequence Flows along with Data Stores, and Data Objects to achieve a desired result. Any VDML Collaboration can have an Activity network—Activities and Stores connected by Deliverable Flows to produce deliverables and value contributions. A BPMN Process cannot map to a Business Network because a BPMN Process does not include Message interaction between business entity Participants. Most VDML Activity networks that are equivalent to BPMN Processes are in Capability Methods (Collaborations) which can be engaged in multiple contexts (equivalent to a BPMN "global" Process).
Choreography	No equivalent	BPMN provides Choreography as an alternative view of a BPMN Collaboration to specify the sequence of exchanges independent of the internal operations of Participants. In VDML, a sequence of exchange and some Choreography constraints can be inferred from Deliverable Flows along with VDML constraints and planning percents on Ports and Resource Uses, but these attributes exist to support value computations. VDML is only concerned with exchanges of deliverables and

		values, not the specification of a message exchange protocol for participants.
Conversation	Business Network (conversation view)	A BPMN Conversation identifies clusters of message flows that achieve a supporting purpose between BPMN Participants within a Collaboration. A BPMN Conversation can map to a conversation view of a VDML Business Network where each conversation is an abstraction of the exchanges between the Parties as defined by a component Business Network.
Participant	Party	In BPMN a Participant is a business entity that plays a role, or it is the role of a typical business entity (the specification is ambiguous). In VDML, an explicit distinction is made between Roles and Participants. A Party is a business entity that plays one or more roles in a VDML Modelm, typically in a business network. . A Role can be filled by an Actor, a Party, a Collaboration or another Role.
Resource Role	Role	BPMN defines Resource Role as the use of a Resource as a performer that can be specified in the form of a specific individual, a group, an organization role or position, or an organization. A BPMN Resource Role may be associated with a Process or an Activity. This is distinct from a BPMN Participant in a Collaboration. VDML adopts a universal Role concept, whereby a Role is always contained in a Collaboration and determines the manner in which the Participant contributes to the Collaboration (by performing Activities). A Role is filled by a Participant that may perform multiple Activities within the Collaboration. A BPMN Resource Role maps best to a VDML Performer (a sub-type of Role) in a Capability Method, and a BPMN Participant maps best to a VDML Party (a sub-type of Role) in a Business Network. Resource Role in BPMN is not recursive: there is no concept of “Roles of Roles” nor “Roles of Collaborations” as in VDML.
Resource	Participant, Actor	BPMN uses Resource, to define, in the abstract, a performer in the form of a specific individual, a group, an organization role or position, or an organization. Typically a person as Resource is assigned to a Resource Role. In VDML, the Participant in a Role may be an Actor, a Collaboration (Participants working together) or another Role that is expected to directly or indirectly engage a Collaboration or an Actor. Collaboration, Actor and Role are all distinct sub-types of Participant.

Partner Entity	Participant	Partner Entity is a sub-type of BPMN Participant that represents a business entity in a BPMN Collaboration. In VDML, a Role refers to a Role Definition that describes the nature of the Role in a Role Library. The Participant in a Role may be a specific business entity or other form of Participant (Role or Collaboration).
Partner Role	Role Definition	Partner Role is a sub-type of Participant that characterizes the nature of participation of a business entity such as a buyer or seller. In VDML, a Role refers to a Role Definition that describes the nature of the Role in a Collaboration. The Participant in a Role may be a specific business entity or other form of Participant. The BPMN Partner Role would translate to a VDML Role with the business entity as Participant.
Process elements "cluster"	Activity	In VDML, an Activity applies a business Capability, so the scope of an Activity aligns to the scope of the associated Capability. If the Activity engages a Collaboration to perform the Capability, then the scope of the Collaboration aligns to the scope of the associated Capability. BPMN does not recognize Capabilities. In a transformation from VDML to BPMN, this capability alignment is not a problem since it is resolved in the source VDML model. However, when translating from BPMN Processes and Activities, there is no basis in BPMN for alignment with Capabilities. A BPMN Process cluster of flow elements can be specified to correspond to a business Capability that consists of 1 or more Activities, zero or more Gateways and zero or more Intermediate Events, connected through zero or more Sequence Flows and/or zero or more Data Associations.
Sequence Flow	Deliverable Flow	A BPMN Sequence Flow initiates a receiving element as a result of a source Event, Task, Gateway or Activity. It does not convey information or a deliverable, so the availability of relevant data and deliverable(s) is implicit. A BPMN process has a data context that may be referenced by any element within the Process. VDML does not define a Collaboration data context, and explicitly delivers relevant deliverables via Deliverable Flows to enable action by an Activity. Not all deliverables may be explicitly delivered, but those that have associated business value and are depended upon by a receiving Activity or Collaboration are explicit.
Message Flow	Deliverable Flow	If a Deliverable Flow is between independent entities (i.e., in a VDML Business Network) it must be represented in BPMN as a Message Flow in a BPMN Collaboration.

		In BPMN, a Message Flow connects to a process within each Participant. A Message Flow cannot connect to a Data Store but a Message Flow can connect to a Send/Receive Task which may reference a Data Store within the sender/receiver process, so that data may be sent and received as a Message. In a VDML Business Network, deliverables are exchanged via Deliverable Flows. Each Participant, through execution of one or more roles, has Business Network Activities to receive and/or send deliverables. These Activities delegate to internal Capability Methods of the Participant to apply the relevant internal capability of that Participant. A deliverable may be received into a Store of the Capability Method that allows for buffering of the asynchronous actions of Participants and can account for delays.
Message	Business Item	A BPMN Message conveys data to another business entity Participant. It is associated with an Item Definition. A VDML Deliverable Flow conveys a Business Item that is associated with a Business Item Definition. A Business Item not only represents a deliverable between business entities, but is used consistently within VDML as a deliverable between Activities and Collaborations, as well as deliverables held in Stores. A Message in BPMN embodies information that is exchanged between Participants in a Collaboration. The PBMN Item Definition in the Message may indicate if the Message content refers to an independently-delivered deliverable or if the message content is the deliverable (e.g., a purchase order). In VDML a Business Item, carried by a Deliverable Flow, represents the actual deliverable which may be information (such as a specification or a contract) in the Message or physical, monetary or other things of value, tangible or intangible delivered by other means.
Data Object	Business Item	A BPMN Data Object is associated with an Item Definition and a VDML Business Item is associated with a Business Item Definition. A Data Object can flow from or to Activities, but such flow does not define a constraint on execution. BPMN allows for a visualization that shows a Data Object on a Sequence Flow when a Sequence Flow and Data Object flow have the same source and target giving the appearance of a Deliverable Flow. In BPMN, a Data Object can only "flow" inside a Process (i.e., the data is local to the Process). Data may be delegated and

		received by a Call Activity where the data is transformed through an Input Output Specification. Otherwise, Data flow between processes must occur through a Data Store. In BPMN, if data is to be communicated to an external entity, it must be transformed to a Message by the sender in a Send Task, and the Message must be transformed back to a Data Object by the receiver in its Receive Task. It is expected that the Item Definition associated with a Data Object would be the same as that for the Message sent or received. The life cycle of a Data Object is the life cycle of the Process in which it is contained, i.e. it is created when the Process instance is created and destroyed when the Process instance is over. Data that must flow between processes within the same business entity must be passed through either a Data Store or Call Activity (delegation). In VDML a Business Item can be exchanged between Activities of any Roles within a Collaboration, including Parties in Business Networks.
Data Output Association, Data Input Association	Deliverable Flow	Data Associations “move data between Data Objects …and inputs and outputs of Activities, Processes, and Global Tasks.” Similarly, Deliverable Flows move deliverables between Activities and Stores. In addition, “Tokens do not flow along a Data Association, and as a result they have no direct effect on the flow of the Process.” [Quotes are from the BPMN specification.] A Data Association is NOT equivalent to VDML Port as a basis for capturing measurements, e.g. delivery time. A Data Object flow may link many sources to many targets, so the source of measurements (e.g., Activities) would be ambiguous. A BPMN Data Object with only one Data Association (e.g. related to one Activity) would not be represented in a VDML model unless it represents resources being held In a Store for consumption by an Activity. In transforming from BPMN to VDML, a Data Object suggests that an Activity needs a Business Item for input or produces a Business Item as output, but it may not be clear how that corresponds to the dependencies defined by Sequence Flows. This ambiguity must be resolved by the user. A BPMN Data Object may have multiple Data Output Associations and Data Input Associations. It is ambiguous

		which targets are dependent on which sources.
Item Definition	Business Item Definition	A BPMN Item Definition defines the type of thing referenced, used or conveyed by an Item Aware Element. This includes Activity, Data Store, Data Object, Message and others. In VDML, a Business Item Definition defines a type of deliverable and is classified In the Business Item Library. Business Items are deliverables in the VDML model and each references their associated Business Item Definition.
Data Store	Store	There is some similarity between these concepts, but their functions are quite different. A Data Store is an Item Aware Element and can be referenced by a Data Object. While a Data Object may represent physical things, and a Data Store may have information about physical things. It does not hold non-information things. A BPMN Data Store does not have input or output flows, but it can be shared by different processes to support shared data. A VDML Store holds Business Items pending input to Activity(s). It has input and output Deliverable Flows and captures measurements regarding delay, number of Business Items held in the store, etc. Essentially, a VDML Store is an element that can be in the flow of deliverables in multiple Collaborations, while a BPMN Data Store is a shared place to hold information. Consequently, there is not a direct translation between these.
No equivalent	Pool	A VDML Pool specializes Store to provide a source of reusable Business Items tracking their availability, assignment and release..
No equivalent	Assignment	The VDML Assignment element links a Role to its Participant. Assignment is context-dependent. This supports the VDML multiple context capability where the same Collaboration (Process) may be used in different contexts with different Role Assignments. In some cases, a Business Item identifies a Role as input to a delegation. The Role may identify a subject for action by the Collaboration, and it may also be assigned as a Participant in a Role within the Collaboration (such as a patient who also takes a role in defining the treatment plan).
Call Activity	Activity	A BPMN Process may engage a sub Process (global Process) via a Call Activity. Multiple Call Activities in the same or different Processes may engage the same global Process. While the Call Activity may specify different inputs and outputs, the called Process specification is the same for all uses of the Process.

		In VDML, an Activity may engage a Collaboration that functions as a sub-process to deliver a Capability. Each such Activity that engages the same Collaboration creates a Delegation Context. Each Delegation Context may identify a different set of measurements and Role assignments for the engaged Collaboration and its elements. This is essential for analysis of performance and value contributions in these different contexts as well as consideration of different scenarios (see below).
No equivalent	Scenario	A VDML Scenario defines the set of measurements associated with VDML elements within a particular operational context such as a particular product mix. It is beyond the scope of BPMN.
Input Output Specification	Port Delegation	BPMN defines the inputs and outputs of a Call Activity and a Process (as well as some other elements) with Input Output Specifications. These define the Data Linkage for engaging a Process. VDML uses Port Delegation to explicitly map Ports of a parent Activity to Ports of the Collaboration to which it delegates thus identifying the specific Activity or Store that receives or produces the input or output deliverable. It is essential to VDML that these are explicit objects, as Ports of the sub Collaboration are differently mapped in different Delegation Contexts. Value Add aggregation and related Measurements rely on Port Delegation.
Gateways	Port conditions, planning Percentages or offsets and Resource Use	Gateways define branch and merge of alternative and concurrent paths in BPMN. In VDML, all paths are determined by Ports and their conditions and planning Percents on Port Containers (e.g., Activities) as well as Resource Use elements linking Activity inputs to outputs. Ports are integral parts of their Port Containers as they define how inputs are received and outputs are produced. An Intermediate Port defines the point in time between start and end when the Port is active. Allocations to paths are measured statistically as defined by conditions and planning Percentages on ports, and the association of inputs to outputs within an Activity through Resource Use. A Resource Use may also define when multiple Business Items are consumed from a Deliverable Flow for a single unit of production (e.g., four wheels for a car). It would otherwise not be possible to do meaningful analysis of measurements of performance and value, as well as analysis of resource requirements, for a Deliverable output. Packaging Ports and Resource Uses with their Port Container is also

		important for the VDML abstraction (reduction of model complexity) and for alignment of Activities with Capabilities. Consequently, Ports and their constraints that determine the inputs and outputs of a Port Container (e.g., Activity) will translate to BPMN Gateways and Timer Events linked to the Activity through Sequence Flows to be part of an implicit "process elements cluster."
No equivalent	Org Unit	This is beyond the scope of BPMN. VDML is about business design, not just process design. It is essential to VDML to associate resources, personnel and capabilities with organizations to support an understanding of accountability and operation of the business.
No equivalent	Value Proposition	Beyond the scope of BPMN. Value Proposition is fundamental to the ability of VDML to represent the delivery and exchange of value and the contributions to value by the activities and capabilities engaged in a value stream.
No equivalent	Value Add	Beyond scope of BPMN. Value Add elements are essential for support of Value Propositions and assessment of the sources of value in a value stream. Value Add elements are associated with the Output Ports on Port Containers (Activity, Store, Collaboration) and are linked through Port Delegations for collaborations as sub-processes. Value aggregation in BPMN would require reconciliation of value Add associations with Data Flows, Sequence Flows, Message Flows and Call Activity Input Output Specifications.
No equivalent	Capability	Beyond scope of BPMN. BPMN does not define Capabilities nor provide any indications of the use of capabilities or the Processes/Activities that apply Capabilities. Furthermore, existing business processes may not be well aligned with Capabilities, so a Process may incorporate multiple capabilities and/or provide part of a capability complemented by other Process(es). There is no BPMN construct defined for aligning a capability and its inputs and outputs with the associated cluster of BPMN process elements.
No equivalent	Capability Offer	Beyond scope of BPMN. VDML identifies a Capability Offer for each Org Unit that provides a Capability. BPMN does not address organization structure nor capabilities.
No equivalent	Business Item Library, Value Library, Capability Library, Practice Library, Role Library, SMM Library	Beyond scope of BPMN. These VDML Libraries are hierarchical structures to define, name and classify concepts that may occur multiple times in a model (or in different models) so they can be consistently and uniquely named and defined. An SMM library defines a collection of shared measures used in VDML. Industry libraries should be developed and shared. The Capability Library is particularly important for identification of multiple occurrences of the same Capability implementation that may be considered as alternative sources or candidates for

		consolidation.
No equivalent	Measured Characteristic	Beyond scope of BPMN. Measured Characteristics are defined as required on Measureable Elements to support analysis using VDML. They provide different measurements on a Measureable Element for each Analysis Context (e.g., Scenario) in which the element is used. The metamodel details for measurements are defined by SMM (Structured Metrics Metamodel).
End Event	Output Port of Collaboration	While End Events of BPMN and Input/Output Ports of a VDML Collaboration correspond to the end of a Process or Collaboration, the semantics of the BPMN End Event are quite different from those of a VDML Port. A BPMN End may define one of alternative ends of a Process. In VDML, multiple Output Ports define the possible output deliverables. Output Ports define output paths and output Ports define associated value contributions by reference to Value Adds. BPMN determines when an End Event is reached; VDML defines statistically, how often and with what values the deliverable is output.
Error Event, Compensation Event, Cancel Event or Escalation Event	No Equivalent	Error, Compensation, Cancel and Escalation events represent technical mechanisms for resolving exceptions. VDML does not deal with these events since they are not part of the business-level abstraction. In VDML the handling of business exceptions fits in the regular structure of the model and are simply represented as alternative paths. Nevertheless, the statistical frequency of occurrence of these Events should be reflected in losses associated with the affected Collaborations or Activities.
Start Event	Input Port of a Collaboration	A BPMN Start Event triggers the beginning of a Process. Different types or Start events are triggered by different circumstances: a “None” Start Event occurs at the top level Process or a global Process. The occurrence of a specific or repeating time, the satisfaction of a condition or the arrival of a message can trigger a Start Event. Other Start Events occur when they catch an event with the same Event Definition as an event thrown by an intermediate or End Event. In VDML, Deliverable Flow to an Input Port defines the start of Port Container (i.e., an Activity, Collaboration or Store). A delay before starting can be statistically represented by an Input Port offset or the duration of a Store with a Deliverable Flow to the delayed Activity.
Intermediate catch event in normal flow	Store or intermediate Input Port	In BPMN, normal flow can be delayed by a catch Event waiting for an event thrown by an intermediate or End Event for the same Event Definition. For Timer Events and Condition Events, the throw is implicit and is “thrown” whenever the specified time or condition occurs.

		In VDML, the delay before starting an Activity that occurs while waiting for a message, a condition or a specified time can be statistically represented by the duration of a Port offset or a Store with a Deliverable Flow to the delayed Activity. However, if the catch Event occurs between Activities that are both part of a cluster that translates to one VDML Activity, then the delay must be represented by an offset delay for an Input Port of the VDML Activity for the deliverable where the delay may represent, not only the delay for the event, but also the duration of work within the VDML Activity that occurs before the catch Event is encountered.
Intermediate Boundary Event	Intermediate Output Port	A BPMN Boundary Event is attached to the boundary of an Activity. A Message, Signal, Timer or Conditional Event will “catch” the arrival of a message, a signal, the occurrence of a specified time or the occurrence of a condition and initiate another flow before completion of the attached Activity. In VDML, an intermediate Output Port corresponds to the BPMN boundary event and it outputs a Deliverable Flow corresponding to the alternative flow initiated by the boundary event. The Intermediate Output Port specifies the offset from the end of the associated Activity representing the statistical occurrence before the alternative end of the Activity. Other measurements will reflect the statistical frequency of occurrence of the event. These boundary events can result in parallel flows or they can cause the associated Activity to terminate. The statistical frequency of occurrence of the event must also be factored into the performance/value contribution computations for the Activity and subsequent Activities in the resulting Deliverable Flow(s).
Intermediate throw and End (throw) Events	Store	In BPMN, a “throw” event can occur as an intermediate or end Event. When it occurs, it throws an event with a specific Event Definition to be caught by a counterpart catch Event waiting for an event of the same Event Definition. The catching event(s) may be Start Events or intermediate Events and will initiate alternative flows. The counterpart in VDML is a shared Store that may receive input in one Collaboration and provide output in another Collaboration.
Send Task	Output Port	In BPMN a Send Task exists for the transformation of an internal Data Object to an external Message and Message Flow. The Data Object may not be explicit in the BPMN model. The Send Task will specify the transformation with Input Output

		Specifications to create a Message. The message will specify the Item Definition (deliverable). For VDML, we may infer that the source of the deliverable information is a Data Object, if present, or the source of the Sequence Flow that initiates the Send Task. There is no transformation of deliverable to message since the output is still a Deliverable Flow.
Receive Task	Input Port	In BPMN a Receive Task exists for the transformation of an external Message from a Message Flow to Internal data. The Receive Task will specify the transformation with Input Output Specifications to create a (possibly implicit) Data Object from a Message. In VDML, a Deliverable Flow corresponding to the BPMN Message Flow will deliver the Business Item (Message) to the Input Port of the Port Container corresponding to the BPMN element that receives input from the Receive Task. The Deliverable Flow will typically be directed to a Store since the sender and receiver have asynchronous operating environments and the Store will define the delay or other characteristics of an input queue.
Choreography Activity	No Equivalent	A Choreography Activity represents a step in a message interaction protocol between business entities. VDML does not address interaction protocols, only dependencies on receipt of deliverables.
Group	No equivalent	A BPMN Group is a visual separation of elements that are members of the same category. There is no similar concept in VDML. A BPMN Group might be used to mark up a BPMN specification in order to align process elements (a cluster) to a business capability.
Link	No equivalent	A BPMN Link is a mechanism for visually connecting a Sequence Flow in one diagram to a Sequence Flow in another diagram. There is no equivalent in VDML and only a need for transformation if there were an attempt to align BPMN displays with VDML displays.
Loop Characteristic (Activity, Task or Sub-Process)	Activity	In BPMN, Loop Characteristics determine the manner in which an Activity, Task or Sub-Process is executed multiple times. The presence of Loop Characteristics is represented as an icon with a circular arrow within an Activity, Task or Sub-Process element for repeated performance, or an icon of parallel lines within an Activity, Task or Sub-Process if multiple instances are to be created and executed. If multiple instances are created, they may be specified as sequential or concurrent. In addition, BPMN may specify that events are fired for completion of individual instances, after all are completed, or other variations.

		In VDML, control of the individual instances or iterations is not at issue. The measurements and deliverable flows for a single Activity represents the net effect of the BPMN "loop." If the instances are sequential, then the duration of the Activity is the statistical sum of the durations of the instances; if they are concurrent, then the duration is the statistical average. Other measurements must be appropriately adjusted to represent the value contributed to the deliverable of each instance.
Business Rule Task, Script Task	Port	In BPMN, these tasks perform an automated action. Generally, VDML will not represent this level of detail except that if the result is input to a Gateway, then it may be the basis for a constraint or planning percentages on Input Ports, or distribution among alternative Output Ports.
Service Task	Activity	A BPMN Service Task indicates that a service is engaged to perform the desired work. A VDML Activity may represent the application of a shared Capability that may be internal or external to the enterprise and may be engaged through delegation or left undefined (black box).
User Task	Activity	In BPMN, a task performed by a person or group. In VDML this is an activity with a role filled by a person or a Collaboration (group of persons).

Transformation of a BPMN Model to a VDML Model

We expect that transformation of a BPMN model to a VDML model will not occur as often as transformation from VDML to BPMN, but we start with that transformation because it highlights the degree of abstraction represented by VDML.

The transformation to VDML will be useful to some users who want to model a higher-level abstraction of their current business processes and apply value stream analysis, value network analysis, capability analysis or other techniques supported by VDML. The transformation will be helpful but the result will be far from complete as a VDML model. A BPMN to VDML transformation depends on certain assumptions of the tool vendor as well as user design decisions and additions to the VDML model that are beyond the scope of BPMN.

This transformation is more complex than the transformation of a VDML model to BPMN because BPMN makes many distinctions that are not recognized in the VDML abstraction. For example, a BPMN Process with a variety of Activities, Tasks, Gateways, Events, Data Objects, and Sequence Flows translates to a VDML Collaboration with Activities and Stores, each with Input and Output Ports connected by Deliverable Flows. The same VDML element types apply to the exchange of messages between business entities.

A small example

We will start to illustrate these differences and the challenges of transformation with a small example. We will focus on a Process cluster that translates to a single VDML Activity. Later, we will consider the reverse transformation, taking the VDML Activity and translating it back to BPMN.

Figure 1, below, depicts a sample Process cluster that applies the capability of one performer. If more than one performer were involved, then the cluster would translate to a collaboration of those performers engaged by a delegating Activity. This cluster is fairly complex because the Gateways and Events that control the inputs and outputs of a cluster will be incorporated into a single VDML Activity.

Here, an Invoice is received as a Message to a Receive Task. The Receive Task creates an internal Invoice Data Object. If the shipment also has arrived, the process proceeds through the Exclusive Gateway to the Authorize Payment Activity that uses the Invoice and Shipment Data Objects. If the shipment has not arrived, a Timer is set and the performer waits for the shipment. Receipt of the shipment occurs elsewhere in the enterprise and results in a Signal. The Signal Event terminates the Wait for Shipment Activity and control proceeds through the Exclusive Gateway to the Authorize Payment Activity. If the shipment has not arrived before the Timer expires, the invoice is rejected. The example illustrates a message Receive Task, two Data Objects, two Activities, branch and merge Gateways, a boundary Timer Event and a boundary Signal Event.

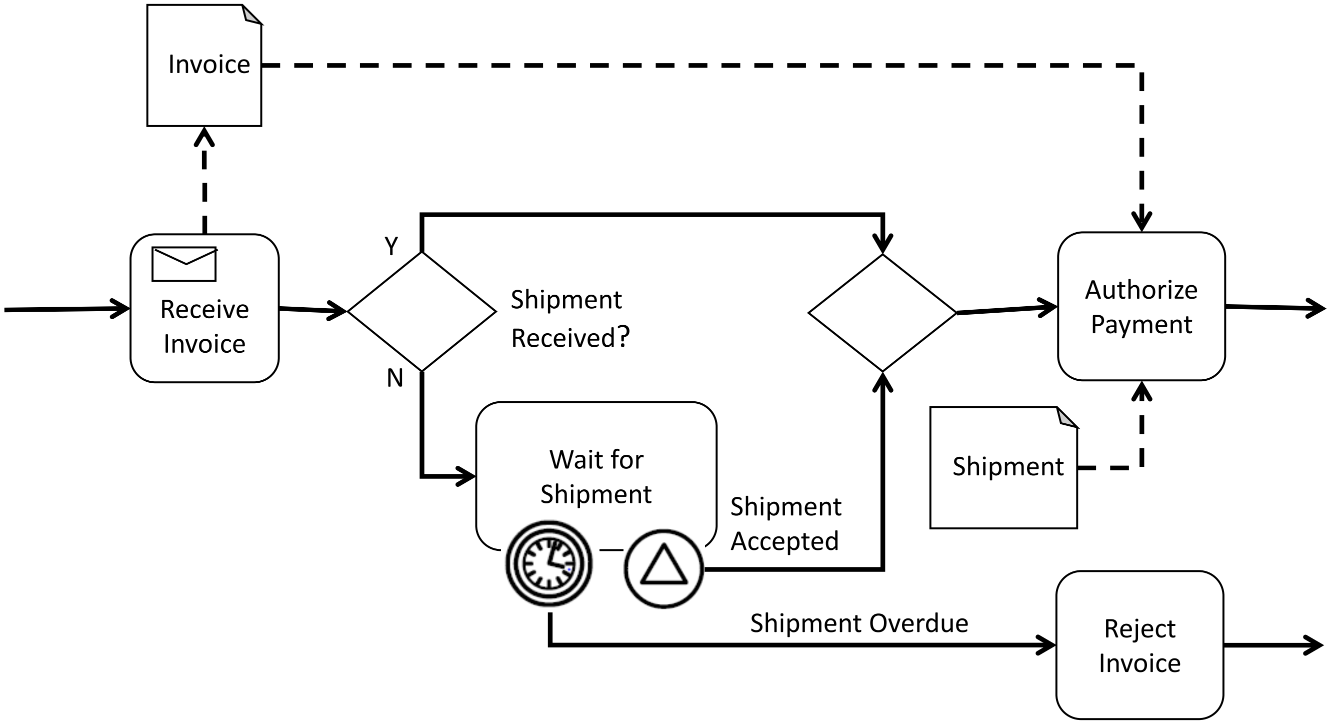

Figure 1, Example Process Cluster

Figure 2, below, depicts a single VDML Activity that might be the result of transforming the Process Cluster of Figure 1. The Output Ports have conditions as indicated by the bar through the middle. The Output Ports also have associated Value Adds, not shown, as indicated by the black fill of the Ports. The Value Adds may capture measurements such as cost and duration associated with each of the Output Ports. A planning Percentage on the Output Port specifications will indicate that Payment Authorization occurs most of the time and a planning Percentage on the Invoice Rejection will indicate that it occurs the rest of the time.

The Activity has two inputs: the Invoice and the Shipment. Both are received from Stores since they are received from independent Collaborations—the invoice from a Business Network exchange, and the Shipment from a shipment receiving operation elsewhere in the enterprise. Resource Use elements (not shown) may link the two inputs to the Payment Authorization output although the Shipment Data Object is not explicitly linked to the Shipment Event in Figure 1. The Shipment Input Port has an offset to reflect an arrival delay (an Input Port attribute).

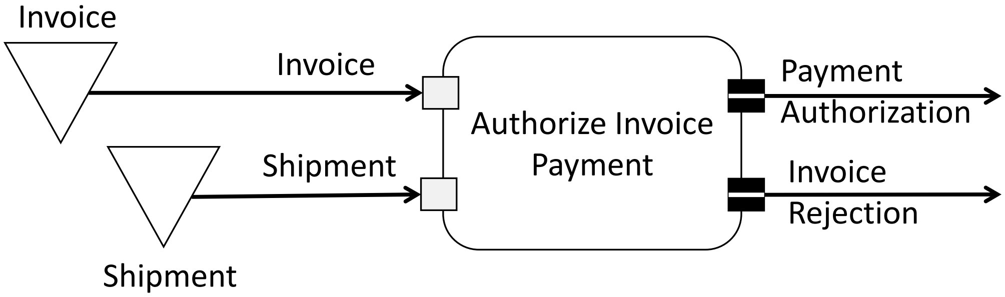

Figure 2, VDML Activity Translated from Figure 1

This transformation highlights the loss of information that is primarily a result of the VDML abstraction. There is no need to be explicit that the outputs are mutually exclusive since the statistical measurements are used independently, but the measurements should be complementary (entered by the user). The distinction between a Message for the Invoice and a Signal for the Shipment is not relevant in VDML and both are represented as Deliverable Flow inputs. The Timer is reflected in the offset of the Shipment Input Port and the associated duration measurement for the Invoice Rejection Output Port. There is no need to reconcile each Invoice with each Shipment since we are only interested in the statistical measurements associated with the Activity and its Ports over time—multiple invoices and shipments.

Transformation requirements

We avoid repeating here the mapping information in the above table, and we will not define a mapping process in detail. Instead, we will outline requirements to be considered in a BPMN to VDML transformation. Details of a transformation will be left to each implementer who will determine the computational techniques, transformation defaults and user participation requirements of their modeling tool.

The transformation requirements outlined below are general approaches. Implementers may find that they require refinements or they represent one of alternative approaches.

- The Message Flows of a BPMN Collaboration translate to Deliverable Flows within a VDML Business Network, with the Item Definition of each Message translating to the Business Item Definition associated with a Business Item (deliverable) for the Business Network Deliverable Flow. A BPMN Conversation translates to a VDML sub-Business Network. In the VDML model, the parties of the parent Business Network must be assigned to the corresponding Parties of the sub-Business Network.

- Generally, VDML Business Network Deliverable Flows will not be linked directly to the source and destination Capability Methods within the network parties. Business Network Activities of each Party will input and output Deliverable Flows exchanged with the other Parties. These Activities will delegate to Capability Methods internal to the associated party. The Business Network Activities or the Capability Methods they engage may use Stores to buffer inputs and outputs.

- A BPMN Data Store may be a bridge to other BPMN processes. A BPMN Data Object is confined to the scope of its Process. Data may be posted in a Data Store for access by other Process(es). Therefore, when a Data Store is associated with a Data Object it must be considered as a potential "channel" for translation to Deliverable Flows. VDML uses Stores in a similar manner. However, the semantics are somewhat different. A VDML Store is a buffer, a place to hold deliverables (Business items) pending acceptance by a recipient Activity. So flow of a deliverable out of a Collaboration will typically appear as a Deliverable Flow input to a Store where the same Store provides an output Deliverable Flow in another (receiving) Collaboration.

- Generally, a BPMN Process translates to a VDML Capability Method. However, BPMN does not define capabilities. A Process translation to VDML requires alignment with a Capability. A BPMN Process will typically align with a more general capability. If it does not, there may be a need to do some refactoring either with the related BPMN processes or with the Capability Methods that are generated from the Processes.

- Within a Process, clusters of BPMN elements must each be identified as the application of a capability. This clustering must be performed by a user. Here we propose that the user use BPMN Groups to define the boundaries of clusters aligned with Capabilities. Then where Sequence Flows or Data Object Flows cross Group boundaries, they represent inputs and outputs to the VDML Activity for the associated Capability. The Group Category might be used to identify the Capability name. In defining Process clusters, every BPMN element of a Process must be included in a cluster along with the Activity(s) that do the related work.

- If the Activities in a cluster are all performed by the same Performer, then the cluster can be transformed to a single VDML Activity. If a cluster involves multiple Performers, then it should be translated to an Activity that engages a Capability Method.

- Where the cluster includes more than one Activity performed by the same Resource Role, it can be represented by a VDML Activity with intermediate Input Ports or intermediate Output Ports to segment the work of the VDML Activity to correspond to the work of the component BPMN Activities.

- A cluster may have Data Object inputs and/or outputs that translate to deliverables on Deliverable Flows, but a transformation mechanism will not be able to distinguish Data Objects that must be delivered for an Activity to start as opposed to supporting data.

- Most Deliverable Flows will correspond to Sequence Flows. Since Sequence Flows do not identify deliverables (Business Items) either these must be inferred from Data Objects and Messages or identified by the user. The BPMN model may have Item Definitions as a basis of VDML Business item Definitions, but it is likely that more Business item Definitions will be required for the VDML model since BPMN focuses on flow of control, not flow of deliverables.

- The user will need to define the Capability Definitions and the Capability hierarchy for the Capability Library. The user will also need to define the Org Units and organization structure, and link Capability Offers and Capability Methods to the Owner and Provider Org Units.

- Where a Data Object is specified as an input to a BPMN Activity, it is likely to indicate the need for a deliverable on an input Deliverable Flow to the VDML Activity. However, it may be associated with a Sequence Flow to the Activity, or it may have another source and is assumed to be available for that Activity. The user must validate Business Item assignments, reconcile redundancies and define Business Items for Deliverable Flows created from Sequence Flows.

- Item Definitions are associated with Item Aware Objects—this includes Activities, Data Objects, Data Stores, Messages and Tasks. These may be collected as Business Item Definitions for the VDML Business Item Library. The Business Item Library will have a flat structure to be organized into a hierarchy/taxonomy by the user.

- An embedded Process should be translated to a Capability Method assuming it represents the application of a Capability. The user will need to define the input and output deliverables since the embedded Process relies on the data context of the parent Process. A Call Activity engages a global Process. The global Process has explicit inputs and outputs, but they are not expressed as deliverables. Again, the user will need to define the input and output deliverables. For both the embedded Process and the global Process, the Input and Output Ports of the Capability Method must be linked to the input ports and output ports of the delegating VDML Activity.

- A BPMN Resource Role can have a Resource Assignment Expression that evaluates to a Resource (e.g. an Actor). If the Actor is provided as input to the Process from a Call Activity, then the Actor should be passed in a Business Item in the VDML delegation. It likely will require user participation to determine the source of the Actor and if it is passed to the process from the Call Activity.

- Gateway branches and merges translate to constraints and planning Percentages on the Input or Output Ports of Activities or Stores. A branch will become alternative Ports on the output of the source of the Sequence Flow that is input to the Gateway. The user must determine the "planning percentage" (a statistical allocation) between the branches to support value contribution analysis. A merge will be represented as alternative or concurrent Ports on the input of the target Activity. If the alternative inputs are matched (e.g., match invoice to shipment), then the input is to a Store where there may be a measurement for delay introduced by the matching. The user must determine the delay measurement.

- BPMN defines "catch" and "throw" events. These catch-throw links must be translated to Deliverable Flows. Generally, the Event that throws is in a different Process from the corresponding Event(s) that catches. In most cases, this mechanism should translate to a VDML shared Store that receives an input in the throwing Collaboration and provides an output in the catching Collaboration.

- Timer and Conditional Events are distinct. A Timer will translate to a delay that may be reflected in an Activity measurement or in the duration a Business Item is in a Store. A Conditional Event will require user participation to determine if a Deliverable Flow is required, or only a measurement reflecting the frequency or timing of the condition of interest.

These requirements are not exhaustive nor precise, but are provided for insight on the nature of a BPMN to VDML transformation, and as a starting point for development of a transformation capability.

BPMN Elements Not Translated

The following BPMN model elements will not be translated to VDML elements since they represent concepts that are not relevant to the VDML abstraction.

- Group (except as used for defining capability clusters)
- Link
- Choreography
- Choreography Activity

In addition, many specializations of BPMN elements are not distinguished in a corresponding VDML model.

Transformation of a VDML model to a BPMN Model

Transformation of a VDML model to a BPMN model is more straightforward than transformation from BPMN to VDML because the VDML model is a simpler abstraction of the operation of the business, and much of the VDML model consists of concepts that do not appear in a BPMN model, particularly value contributions, value propositions, organizations, capabilities, Scenarios and measurements.

A BPMN transformation will be a useful starting point for development of BPMN processes for a new business design; however much of the detail necessary for effective process control and optimization cannot be derived directly from the VDML model. Some of this can be generated based on defaults defined by the tool vendor but most of it will require detailed business process design by the user.

Extending the small example

We will start to illustrate this transformation, by building on the small example, from above. In Figure 3, below, we show a BPMN Process cluster that might be generated from the VDML example of Figure 2.

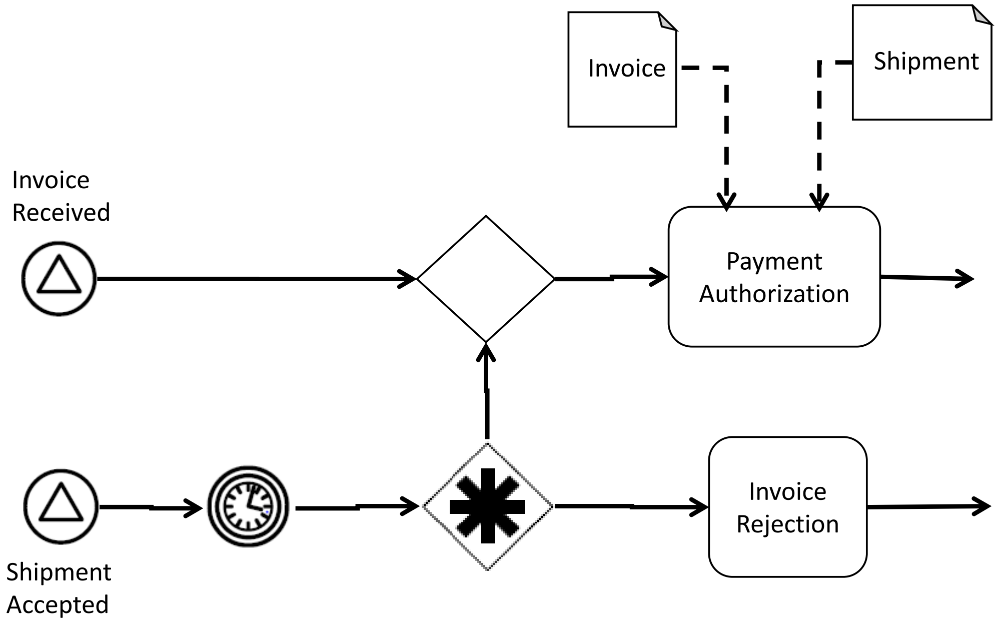

Figure 3, BPMN from the VDML of Figure 2

In Figure 3, both the receipt of an invoice and the receipt of a shipment are shown as Signal Events since there is no distinction between message and signal in VDML, in Figure 2, receiving an external message and the internal shipment receiving event; both are represented as shared Stores. In Figure 3, the shipment event is followed by a timer representing the offset of the shipment receipt Input Port of Figure 2. The Complex Gateway represents the fact that payment authorization requires both the invoice and the shipment, but the shipment does not always arrive in time so the invoice is only paid part of the time as indicated by the planning percentages on the Output Ports of Figure 2.

Clearly this process cluster does not have the same effect as the Process cluster of Figure 1. It further illustrates the loss of information as a result of the VDML abstraction. What it does not show is the loss of information from the VDML model that would result if the user had made a more complete and useful VDML model with value measurements, value proposition, organization structure and specification of capabilities.

Transformation requirements

We will avoid repeating here the mapping information in the above table, and we will not define a mapping process in detail. Instead, we will outline requirements to be considered in a VDML to BPMN transformation. Details of a transformation will be left to implementers who will determine the computational techniques, transformation defaults and user participation requirements of their modeling tool.

The transformation requirements outlined below are general approaches. Implementers may find that these require refinements or that they prefer alternative approaches.

- Each top-level VDML Business Network will translate to a BPMN Collaboration. Each sub-Business Network will translate to a Conversation among the BPMN Collaboration Participants. In VDML, an Activity of a Party sends and receives Deliverable Flows, and that Activity engages a Capability Method that will originate and consume the input and output of the Party Activity. Each Deliverable Flow translates to a BPMN Message with an Item Definition that corresponds to the Business Item of the Deliverable Flow.

- The Activities of the VDML Business Network engage VDML Capability Methods. The Deliverable Flows of the Business Network link to Ports of internal Participant Capability Methods through delegation from their Business Network Activities. Each Input Port should link to the Input Port of a Store within the Capability Method. Each Capability Method Output Port should be linked to the output port of the Activity that produces the Capability Method output. For BPMN, the Message Flows should link to Send and Receive Tasks that correspond to the Output Ports and Input Ports of the Capability Methods.

- Each VDML Capability Method will translate to a BPMN Process. Most Capability Methods will be engaged through delegation from a VDML Activity. The input and output Ports of the delegating Activity must be translated to the input Output Specifications of the corresponding BPMN Call Activity that engages the Process. This is based on the translation of VDML Business Item Definitions in the VDML Business item Library, to BPMN Item Definitions.

- In general, a VDML Role that is not filled by a Collaboration (through delegation) will translate to a BPMN Resource Role. VDML can have Roles in Roles; BPMN cannot. Where the VDML model assigns a Role to a Role, if there is a specific Actor, then the Actor should be assigned to the primary BPMN Resource Role, otherwise the participant of the Primary Resource Role should be left open. For example, in VDML, if there is a committee of managers, and if John is a manager, then John is in the manager Role and the manager Role is in the committee member Role. For BPMN, John would be in the manager Resource Role and, independently, in the committee member Resource Role—the Resource Roles are independent. Roles in Roles are most likely to occur in organizational relationships that will not be translated to BPMN.

- A VDML Business Item can identify a Role of an Actor that is the subject of Activities such as a patient. The Role can then be assigned to a Collaboration Role to be the Performer of Activities in that Collaboration. In BPMN, a Resource Role can have a resource Assignment Expression that evaluates to the Resource (i.e., Actor) to fill the Role. The translation to BPMN should define an expression appropriate for dynamic assignment of the participant in the VDML Business Item Role to the BPMN Resource Role.

- A VDML Activity (the application of a Capability) with one Input Port and one Output Port is straightforward. The VDML Activity translates to a BPMN Activity, the input Deliverable Flow translates to an input Sequence Flow with the same source, and the output Deliverable Flow translates to an output Sequence Flow with the same destination. Data Object flows can be defined in parallel with the Sequence Flows if desired to capture the Item Definitions corresponding to the Deliverable Flow deliverables (Business Items), or Data Objects for these Business Items can be shown as input or output of the Activity.

- An Activity that delegates will translate to a Call Activity. The Business Items that are input and output of the VDML Activity translate to Item Definitions associated with Data Input and Data Output elements of the Call Activity and the global Process. Details of the Input Output Specification must be developed by the user.

- Translation of a VDML Activity with multiple inputs or outputs can be very complex. An individual VDML Activity will represent a more complex BPMN element cluster. In particular, where Ports control the inputs and outputs of an Activity, these must be translated to Gateways. The Gateways must be inferred from examination of Input Ports, Output Ports, Resource Uses and their associated attributes. In addition, offsets on VDML Ports will require Timer Events in the BPMN translation. Since Gateways do not receive or send Data Object flows, the Gateways must be linked using Sequence Flows. It may be appropriate to generate “parallel” Data Object flows to link Data Objects to the sources and destinations of the Gateway Sequence Flows, bypassing the Gateways. This translation likely will require the use of a pattern matching and transformation technique, and it likely will require some assumptions and defaults by the translation tool. In addition, the user will need to resolve the need for Data Object flows and will need to evaluate and refine the Gateway and Timer Event flows as implied by the obvious limitations of the example transformation in Figure 3, above.

- Generally, VDML does not model iterations. Where an Activity may be performed multiple times to achieve a result, the statistics will reflect the typical number of occurrences, but will not specify that it occurs through concurrent or repeated operations. VDML will indicate where multiple inputs are consumed for a single output (e.g., four wheels on a car) and this may be used as the basis for representing concurrent flows or iterations in a BPMN translation.

- A VDML Store is translated to a throw Event if it only has an input Deliverable Flow, and to a catch Event if it only has an output Deliverable Flow. If the Store has both inputs and outputs, then it may be translated to a Timer Event, but if the delay is inherent in the business operation, a Timer Event is not appropriate. In other words, the delay associated with a Store in VDML may simply be a reflection of an operational delay, not an intentional delay that must be explicit in a BPMN Process.

These requirements are not exhaustive nor precise, but are provided for insight on the nature of a VDML to BPMN transformation, and as a starting point for development of a transformation capability.

VDML Elements Not Translated

The following list highlights the VDML classes that would not be included in a transformation to a BPMN model. These reflect the difference in scope of VDML modeling compared to BPMN modeling, and the information excluded when translating from VDML to BPMN.

- Value Add
- Value Proposition Component
- Value proposition
- Value Library
- Capability Library

- Business Item Library
- Practice Library
- Role Library
- SMM Library
- Org Unit
- Capability Offer
- Assignment
- Scenario
- Delegation Context
- Pool
- Resource Use
- Measured Characteristic (and associated SMM elements)

In addition, there will be business properties on VDML elements that have no counterparts in BPMN.

Feasibility of Defining VDML as an Extension of BPMN

Both BPMN and VDML model business operation "processes." Consequently, the following paragraphs discuss the implications of extending BPMN to support VDML modeling. Implementers of BPMN might be able to leverage their investment in their BPMN modeling tool to offer a VDML modeling tool. Users could then invest in a single tool to do both BPMN and VDML modeling. However, as this discussion hopefully clarifies, it is not possible to define a single metamodel that extends the BPMN metamodel to incorporate a VDML metamodel.

The following sub-sections outline the implications of using the BPMN meta-model as the basis for the VDML meta-model—essentially extending/specializing BPMN classes to support VDML modeling.

Specializations

The following bullets outline how a number of VDML classes might be defined as specializations of similar BPMN classes. These specializations require additional analysis and modeling for validation, but they provide a reasonable starting point for consideration of a BPMN extension.

- Specialize BPMN Flow Element Container for VDML Collaboration and its sub-classes.

- Specialize Item Aware Object for Deliverable Flow.

- Specialize Data input Association for Input Port and Data Output Association for Output Port.

- Specialize Data Object for Business Item.

- Specialize Item Definition for Business Item Definition (in a taxonomy).

- Specialize Activity (BPMN) for Activity (VDML) Store and Pool.

- Specialize Libraries from Definitions Element

- Specialize Base Element for Value Add, Value Proposition Component and Value Proposition
- Make Base Element a Measurable Object
- Specialize Resource Role for VDML Participant and its specializations (i.e., Role, Collaboration, Actor).
- Specialize Base Element for Analysis Context (and Scenario and Delegation context).
- Specialize Base Element for Port Delegation (and sub-classes)
- Specialize Base Element for Capability Offer.

Barriers

The following bullets outline VDML requirements that cannot be met by these specialization of similar BPMN classes even if some BPMN attributes and associations were ignored

- Collaborations, Activities and Stores are all Port Containers. This enables the consistent use of Deliverable Flows and Value Contributions. There is no common abstract class in BPMN to support this.
- The delegation mechanism of BPMN is incompatible with the delegation mechanism of VDML. In addition to support for aggregation of value measurements, VDML supports different delegation contexts, and thus independent measurements, for different uses of the same Capability Method (i.e., Process).
- VDML defines an Assignment element to link a Role to a Participant in the Role. This is essential for Role assignments to be context dependent, i.e., to be defined differently in different contexts in which a collaboration is used, and to enable a Participant in a role to also appear as a Business Item in the same collaboration (e.g., a patient as subject of services and as participant in service planning).
- BPMN is based on a shared data context within a Process such that flow control prevents premature action by an Activity. There are no deliverable flows to or from Events and Gateways. VDML is focused on the flow of deliverables to define Activity dependencies and statistical measurements rather than the precise flow of individual transactions.
- Control-based flow does not support the association and flow of value contributions with deliverables.
- VDML incorporates the concept of a value stream to produce an end deliverable and for aggregation of measurements to feed value propositions. This requires, not only the specification of Value Add, Value Proposition and Value Proposition Component classes, but addition of the mechanisms for delegation and aggregation that link the Value Proposition to the end deliverable and link the Value Adds to the activities and collaborations that contribute to the end product. BPMN Call Activities and Input Output Specifications are not compatible with these mechanisms.
- The operational semantics of BPMN and VDML flows and activities are inconsistent. VDML flows represent continuous flows of deliverables over time while BPMN flows represent controlled execution

of individual business transactions. As an analogy, BPMN defines delivery routes through a city while VDML defines statistical traffic flows through intersections. Alternative VDML paths represent statistical percentages of deliverable flows over time, while alternative BPMN flows represent a choice of one path or the other for each transaction.

- There is no common abstract class (and thus a lack of consistency) for BPMN Process, Collaboration, Choreography and Conversation and thus no root for VDML Collaboration and its specializations (i.e., Org Unit, Community, Business Network and Capability Method).

- A BPMN-based VDML metamodel would provide no compatibility between VDML with CMMN (Case Management Model and Notation).

- There is no common abstract class for Participant, Resource Role, Collaboration, Process and Resource to enable the recursive Role structures of VDML. This is important for full representation of collaborations in an organization where individuals participate in multiple roles directly and indirectly as a result of their roles in other collaborations.

- BPMN does not consider business capabilities and thus provides no mechanism for alignment of Processes and their Activity networks with capabilities.

- VDML allows a Business Network to be composed of sub-Business Networks, recursively, rather than a single level Collaboration with Conversations.

- The delegation mechanisms of BPMN and VDML are incompatible. BPMN defines data transformation between a Call Activity and a global Process, whereas VDML defines the Business Item transfer between the port(s) of a calling Activity and the Ports of a called Collaboration.

- Collaborations, Activities and Stores all have Ports that are an integral part of Deliverable Flows and delegations. In VDML they are all specializations of Port Container, but there is no common abstract class for BPMN Flow Container and BPMN Activity.

- While a Data Object can represent a Business Item/deliverable, BPMN does not provide a mechanism for explicitly defining the flow of deliverables between Activities, nor associating Activity value contributions with deliverables.

- A VDML Store is a holder of business items pending acceptance by target Activity(s). It is part of the Deliverable Flow the same as Activities, but it can be shared by different Capability Methods as a bridge for Deliverable Flows. A BPMN Data Store can hold data about deliverables and share that data across Processes, but does not participate in BPMN Sequence Flows or Data Object Flows. A BPMN Signal Event participates in Sequence Flows, and can provide a link between Processes, but it does not convey business Item Definition information and thus does not support the flow of deliverables between Processes (I.e., Capability Methods).

Achieving a shared metamodel between BPMN and VDML would require substantial re-factoring of the BPMN metamodel. This would certainly be opposed by current BPMN vendors. While this would enable both BPMN and VDML modeling to occur in the same modeling environment, it would not enable a single model to support both a BPMN viewpoint and a VDML viewpoint.

BPMN Elements Not Used

The above approach to extending the BPMN metamodel to define the VDML metamodel uses a limited number of BPMN classes as the basis for specification (specialization to) VDML classes. The following BPMN classes are not involved, directly or indirectly, in the resulting VDML metamodel. Consequently, a BPMN model could not be developed by building on a VDML model.

- Process
- Call Activity
- Callable Element (and its specializations)
- Sequence Flow
- Gateway (and its specializations)
- Data Store
- Collaboration
- Conversation
- Choreography
- Message
- Message Flow
- Event (and its specializations)
- Task (and its specializations)
- Participant
- Partner Entity
- Partner Role
- Input Output Specification (and related elements)
- Expression

This is, at least in part, a result of the additional level of detail incorporated in BPMN and BPMN models.

Feasibility of implementing VDML and BPMN in the same tool

A BPMN vendor may anticipate a competitive advantage from implementing a VDML modeling capability in a BPMN modeling tool. This may provide some leverage in reuse of supporting software, and it may be appealing to customers to purchase one tool to address both modeling needs.

There are three potential approaches: (1) fully integrate BPMN and VDML so that they are alternative viewpoints on a shared model, (2) specialize core modeling elements of BPMN to define the VDML modeling elements and share user interface technology to provide the VDML views, or (3) maintain two distinct models in the same environment and link the related concepts. We will explore each of these in the sub-sections that follow.

Shared Model

A shared model approach would require that common abstract classes be discovered for all of the BPMN and VDML elements. For the most part, VDML elements and BPMN elements become views of these common elements. The following observations identify some of the key requirements of such an approach.

- In order for the common model to be compatible with both BPMN and VDML, the BPMN Processes and Activities (or Process clusters) would need to be aligned with business Capabilities as they are in VDML. A VDML Activity is then a view of a BPMN Process cluster.

- In order to support the VDML viewpoint, deliverables would need to be explicit, at least where they are inputs and outputs of a cluster, are passed by delegation or are passed in message exchanges.

- In order to support the BPMN viewpoint, the constraints and planning percentages of VDML Ports would need to be expressed as gateways with statistical flow measurements. This would require the ability to dynamically translate or propagate changes to a VDML Activity with its Ports and Resource Uses to corresponding BPMN Activities with Gateways and Events to maintain consistency of the model in support of users doing VDML modeling.

- A VDML shared Store would appear in the BPMN viewpoint as Events with Inputs as catches and outputs as throws to communicate across processes/collaborations. A Store or Pool that holds aninventory would not appear in the BPMN viewpoint except where there is an intentional delay that would appear as a BPMN Timer Event.

- A Deliverable Flow must be interpreted in context for BPMN since message flows in a BPMN Collaboration are different from Sequence Flows in a Process.

- A Sequence Flow must essentially be a Deliverable Flow without a deliverable. Where an output contributes to the value stream a deliverable is required. Where an Activity is dependent on the arrival of a deliverable, a deliverable is required. Inputs and outputs of a Store or Pool require deliverables. Sequence Flows (without deliverables) may be used where the ordering of Activities is arbitrary or reflects some physical ordering that does not involve the flow of deliverables.   A BPMN view can express all Deliverable Flows as Sequence Flows.

- Delegation would require both specifications of the data input and output and the deliverables input and output. Essentially, the user for each viewpoint would need to specify the delegation details to support their viewpoint.

- A Data Object is a view of a Deliverable Flow where the deliverable is data/information

- An Assignment element would link a Role/Resource Role to its Participant.

- Process, BPMN Collaboration, BPMN Conversation and VMDL Collaboration (and its sub-types) would have a common abstract class. A BPMN Collaboration would be a view of a VDML Business Network.

Capability Method and Process would be alternative views of a sharable method/process. Org Unit and Community would not be shared.

- Scenario and Delegation Contexts would apply to all measureable elements but the BPMN viewpoint would not expose the measurements except as extensions to BPMN. However, the BPMN viewpoint would need to be aware of scenarios where there are other differences in different contexts such as Role Assignments.

These observations are over-simplifications of defining a shared model. Assuming a sufficient model and associated views can be defined, there is a concern that users would not be able to deal with the complex interactions that would arise when defining or changing one viewpoint. Furthermore, changes to the BPMN or VDML specifications will likely affect the implementations of both languages. This approach is not recommended.

Shared Components

In this approach, a tool would support the definition of separate VDML and BPMN models. The models would share generic components both for core modeling concepts and for the user interface, and links could be maintained for alignment of BPMN Processes, Collaborations and Conversations with VDML Collaborations.

The following paragraphs outline some potential abstract classes that might be shared between the BPMN metamodel and the VDML metamodel.

- A common abstract class for BPMN Process, Collaboration, Choreography and Conversation and VDML Collaboration.
- A common abstract class for Sequence Flow, Message Flow, Event throw/catch and Deliverable Flow
- A common abstract class for port and Data Association
- A common abstract class for Message, Data Object and Business Item
- A common abstract class for BPMN Call Activity, Send Task, Receive Task and VDML Activity and Store.
- A common abstract class for Events, Data Store, and VDML Store as shared across Processes/Collaborations

Most of the other element types would be specialized from a common abstract root much as they are in the current metamodel specification.

This is a preliminary outline that would require considerable further analysis and design. It is not clear if development of the common abstract classes is worth the effort since it probably will not simplify transformation between the models.

The benefit of implementing the two languages in the same tool are

- Users can buy one tool to do both BPMN and VDML modeling

- Transformations can be implemented in the same environment
- Generic user interface and model management facilities can be shared.
- Changes to the BPMN or VDML specifications will not be interdependent except to the extent transformations are affected.

This approach should be considered by vendors that intend to implement VDML and currently have or intend to have a BPMN implementation.

Linked Models

In this approach, a tool supports both the VDML and the BPMN metamodels to develop separate VDML and BPMN models. However, the models are linked so that changes in one model can be coordinated with the other, and analyses in one model can be supported with additional information and perspective from the other model. Implementers can determine which links are beneficial. The following paragraphs are suggestions to be considered.

- **Capability alignment**. BPMN processes may be linked to VDML Capability Methods. Furthermore, selected BPMN Groups or Activities might be associated with VDML Activities (one-to-many). This promotes both alignment with VDML and with service oriented architecture (SOA) and capability analysis.
- **Business Item alignment**. BPMN might be linked to the VDML Business Item Library as its source of Item Definitions.
- **Organizational responsibility**. BPMN processes may be linked to responsible organizations directly, or they may be linked to Capability Offers that associate the Processes with Capability Methods as well as both the organization that provides the capability and the organization that owns the Capability Method (manages the development and maintenance).
- **Deliverable flows**. The lack of identification of deliverables is a gap in business process design. Links might be included to associate inputs and outputs of BPMN Activities with VDML Business Items .

In general, the above links will not be different for different VDML scenarios. However, implementers will need to address alignments between different versions of the BPMN model and/or different versions of the VDML model.

Summary

The relationship between BPMN and VDML is not straightforward. They are different ways of looking at the design of the business. VDML is broader in scope and focuses on statistical measurements of both formal and informal business operations. BPMN is more narrowly focused on formal processes and the flow of individual transactions. Note that case management activities, as defined by the CMMN (Case Management Model and

Notation) specification, will also be represented in VDML models and will require a mapping that is quite different from that for BPMN.

We have explored, above, both transformation between the models and the potential ways of integrating BPMN and VDML. The following paragraphs summarize the observations.

Translation

In summary, parts of a VDML model can be translated to a corresponding partial BPMN model, and parts of a BPMN model can be translated to a more abstract, partial VDML model. A model in one is inadequate to generate a model in the other, and there is information loss in translation in either direction. Different tools may make different assumptions, and the user will be required to make a number of model design decisions and complete the development of the resulting model.

Nevertheless, these transformations have value for reducing the work to create a BPMN model from a VDML model or a VDML model from an existing BPMN model. In either case the result should be a useful starting point for development of more a detailed model.

“Round-tripping” is not reasonable as there is user input required and information lost in either direction of translation, and compensating for these losses would outweigh the value of the transformation. However, it should be useful for a VDML model to include references to identify corresponding BPMN Processes.

Extension

Extension of the BPMN metamodel to define the VDML metamodel is not practical for several reasons:

- Since relatively few existing classes would be specialized and many would be specialized from Base Element, little of the BPMN semantics would be inherited or consistent. Almost all of the VDML elements are either missing from BPMN or have different relationships and semantics.

- As outlined above, there are several important VDML constructs that cannot be duplicated using specializations of BPMN. These are highlighted in the “Barriers” of the BPMN Extension section. In addition, the existing BPMN classes would bring attributes and relationships that are not relevant to a VDML metamodel.

- In order to overcome the barriers to extension, the BPMN metamodel would need to be refactored to introduce additional abstract classes and model elements in BPMN. These would require modifications to existing BPMN implementations and, along with the VDML extension elements, would substantially increase the complexity of an already complex BPMN metamodel. The programming of implementations would become more complex for both BPMN and VDML, and future versions of the metamodels would be very difficult to manage, requiring collaboration of a diverse group of participants.

While these issues are barriers to extending the BPMN metamodel for VDML, they are not barriers for a BPMN tool vendor to implement VDML and BPMN in the same tool environment.

Linked Models

The differences in semantics and structures of the metamodels means that it would not be possible to create a single, integrated metamodel for both VDML and BPMN. However, an implementer could realize benefit from implementing both in a single modeling environment. This would enable some core modeling facilities and user interface facilities to be shared, and mechanisms for transformations between VDML and BPDM models could be implemented in the same environment. Furthermore, for some purposes, links between BPMN and VDML models could improve the consistency of design and coordination of changes. However, users would need to be cautious about maintaining too many links that could make the combined model increasingly complex for larger scale applications.