

# New Capabilities for Process and Interaction Modeling in BPMN 2.0

**Stephen A. White PhD, International Business Machines, and Conrad Bock, National Institute of Standards and Technology, USA**

## 1 INTRODUCTION

This paper provides a high-level introduction to new features in processes and interaction diagrams in the Business Process Model and Notation (BPMN) Version 2.0 [1]. BPMN 2 expands the capabilities of BPMN 1.x Process and Collaboration diagrams, and adds Choreography diagrams for business interaction modeling. Half of the paper covers new elements in Process diagrams, including non-interrupting Events and Event Sub-Processes. The other half covers new capabilities for modeling interactions, including Conversations, and interactive Processes in Collaborations. The paper assumes familiarity with earlier versions of BPMN.

BPMN provides a view of processes (how things get done) with flow charts tailored for business processes and interactions. In addition, BPMN 2 has two ways to model business interactions: Collaborations and Choreographies, the first emphasizing participants, and the second the sequence of interactions. Choreographies are new to BPMN 2.

The primary purposes of BPMN 2 are threefold:

- First, to provide a notation that is readily understandable by all business users, from business analysts creating initial drafts of the processes, to those performing processes or implementing technology to automate them, and finally, to business people who will manage and monitor those processes.
- Second, to support the notation with an internal model that has formal execution semantics enabling process model execution, as well as declarative semantics to relate processes and interactions.
- Third, to provide a standard interchange format for transfer of process and interaction models, and detailed visual information, between modeling tools.

These features of BPMN 2 create a standardized bridge between the business process design and process implementation.

### **1.1 The Origins of BPMN**

The BPMN 1.0 specification was developed by the Business Process Management Institute (BPMI), now merged with the Object Management Group (OMG), and released to the public in May, 2004. BPMN was adopted as an OMG standard in February, 2006. Work on BPMN continued within the OMG and BPMN 1.1 was completed in June, 2007, BPMN 1.2 was completed in June, 2008, and BPMN 2.0 was completed in June, 2010. The term “BPMN 1.x” is used in this paper to represent the 1.0, 1.1 and 1.2 versions of BPMN.

### **1.2 New BPMN Modeling Capabilities**

The basic look-and-feel of BPMN has not significantly changed in BPMN 2, especially for process modeling. The updates to process and interaction modeling are described in the following sections.

In addition to process and interaction enhancements, BPMN 2 defines four new Conformance Levels to support different modeling requirements:

- **Process:** includes Collaboration, but not Choreography and Execution.
- **Process Execution:** for execution engines.
- **Business Process Execution Language (BPEL) Execution:** for BPEL execution engines [2].
- **Choreography:** for Choreography tools. Process and Execution are not required.

Furthermore, the Process Conformance level is divided into three sub-levels:

- **Descriptive:** elements for high-level modeling.
- **Analytic:** elements consistent with the Department of Defense Architecture Framework [3].
- **Common executable;** elements for models that can be executed.

## 2 PROCESS MODELS

A common use of BPMN is for modeling business processes, sometimes in the context of collaborations. BPMN 1.x diagrams are basically the same in BPMN 2, but BPMN 2 adds significant support for more advanced process modeling patterns. The major areas updated include: Activities, Events, Gateways, and Data.

### **2.1 Updated Activities**

BPMN 2 Activities are updated in the following ways:

- Markers for Task types
- A new Business Rule Task
- Changes in Multi-Instance markers
- New Global Tasks
- New Call Activities
- New Event Sub-Processes

The sections below will describe these changes.

#### Markers for Tasks

BPMN 1.x was developed with various types of Tasks (atomic Activities). These provide a set of predefined Tasks, such as sending or receiving a message. BPMN has a User Task, Service Task, Receive Task, Send Task, Manual Task, and Script Task, as well as an undefined Task. In BPMN 2, a Business Rule Task is added (see next section).

In BPMN 1.x the Task types were a part of the BPMN model, but there was no visualization to distinguish the Task types. In BPMN 2, distinguishing markers are added to the Task types (see Figure 1) in the upper left corner of the shape, except for the undefined Task (now called an Abstract Task).

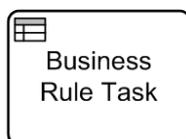


**Figure 1: Task types and their markers**

Note that the Receive Task has two variations. A standard Receive Task has a marker in the shape of an envelope. A Receive Task that is used to start (initiate) a Process has a marker that looks like the Message Start Event.

**Business Rule Task**

A new Task type is added in BPMN 2—the Business Rule Task (see Figure 2). This Task represents an Activity in the Process where a business rule engine evaluates Process data and returns the results. Process data can be updated based on the results, which can then affect the flow of the Process through a Gateway, for example.



**Figure 2: A Business Rule Task**

**Multi-Instance Markers**

BPMN 2 adds notation to distinguish between the two types of Multi-Instance Activities: sequential and parallel. The parallel Multi-Instance Activity maintains the BPMN 1.x notation of three vertical lines (see the Activity on the left in Figure 3). The sequential Multi-Instance Activity now uses three horizontal lines (see the Activity on the right in Figure 3).



**Figure 3: Parallel and sequential multi-instance Task markers**

**Global Tasks**

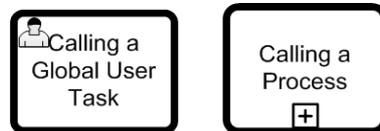
Global Tasks are new elements in BPMN 2 that enable modelers to create libraries of reusable atomic Tasks that can be called into a Process through a Call Activity (see the next section). Global Tasks are not graphical elements themselves, but are reused by graphical elements. Of the types of Process (embedded) Tasks (see Figure 1, above), only Business Rule, Manual, Script, and User types can be Global Tasks.

**Call Activities**

BPMN 1.x provided the capability of reusing one Process in the flow of another. BPMN 2 modifies and expands this capability. The Sub-Process types *reusable*

and *reference* in BPMN 1.x are replaced with a Call Activity in BPMN 2. The remaining BPMN 1.x Sub-Process type *embedded* becomes the BPMN 2 Sub-Process element.

The Call Activity is an Activity that reuses either a previously defined Global Task (see previous section) or a Process. Call Activities are distinguished from other “local” Activities by their thick border (see Figure 4). When a Call Activity reuses a Global Task, the marker for that type of Task will be visible in the upper left corner. When a Call Activity reuses a Process, the plus sign marker of a Sub-Process will be visible in the bottom center.



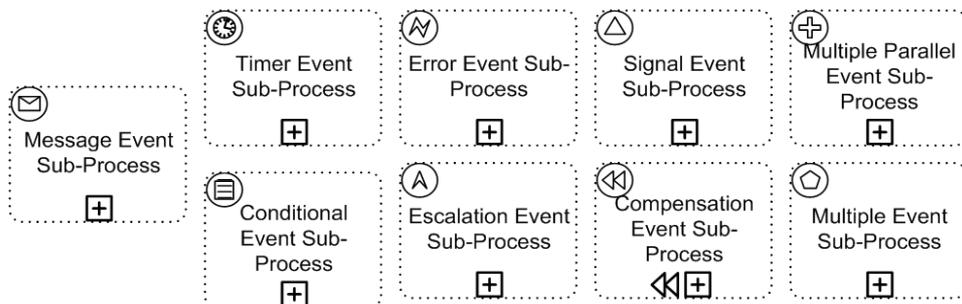
**Figure 4: Call Activities**

A notation pattern introduced in BPMN 2.0 is that thin boundaries represent elements local to the diagram and thick boundaries represent elements that reuse global elements. The pattern is used in Collaboration and Choreography diagrams (see section 3.1.3).

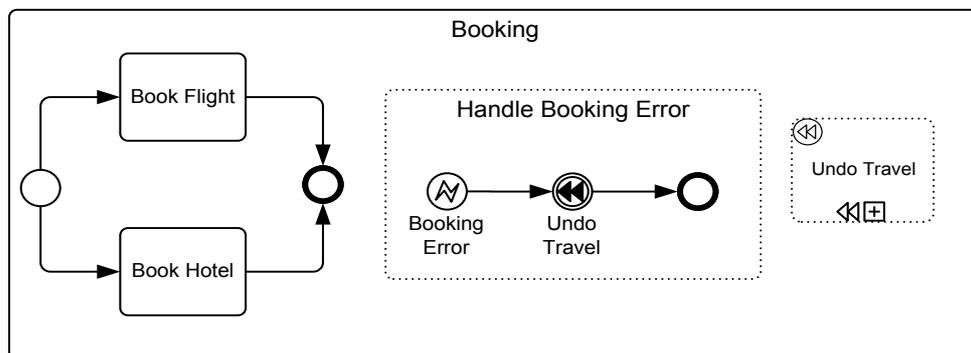
Event Sub-Processes

Event Sub-Processes are a new element in BPMN 2 that combine the characteristics of boundary Events and Sub-Processes. They are similar to boundary Events, except are placed inside the Activity (a Process or Sub-Process—they cannot be used for a Task). They are Sub-Processes in that they are a compound Activity—an Activity that has lower-level Activities as part of its definition.

The Event Sub-Process is distinguished from a normal Sub-Process by its dotted line border and the specific Event type that triggers it shown in the upper left corner of the shape (see Figure 5). If the Event Sub-Process is expanded, then the marker is not shown (because the Start Event is visible), but the dotted border remains (see Figure 6).



**Figure 5: Event Sub-Process types and their markers**

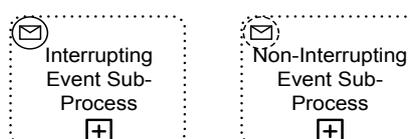


**Figure 6: An expanded Event Sub-Process**

An Event Sub-Process is a Sub-Process that is initiated only when its Start Event occurs. An Event Sub-Process is contained within a Process, but it is outside the main flow of that Process. That is, as the Process flows from its normal Start Event to End Event, the Event Sub-Process will not be initiated. The Event Sub-Process can only be initiated if its Start Event is triggered. Unlike a normal Sub-Process, an Event Sub-Process can only have one Start Event and that Start Event must have a trigger. As with boundary Events, Event Sub-Process can be set to either interrupt its parent Process or not.

Event Sub-Processes set to interrupt will stop its parent Process, when the triggering event occurs, and begin performing the internal Activities of the Event Sub-Process. While the main flow of the Process has been stopped, the Process as a whole will not have completed until all the active Event Sub-Processes have been completed. Event Sub-Processes that do not interrupt will run in parallel with the main Process flow and can be triggered multiple times during the lifetime of the main Process.

Visually, interrupting and non-interrupting Event Sub-Processes are distinguished by the border of the marker for collapsed Event Sub-Processes or the border of the Start Event for expanded Event Sub-Processes. Non-interrupting Event Sub-Process markers have a dashed border (see Figure 7).



**Figure 7: Interrupting and non-interrupting Event Sub-Processes**

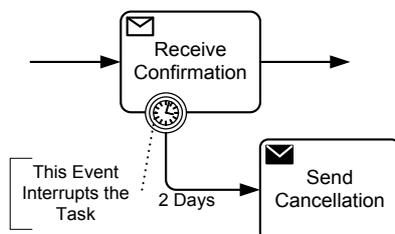
As with boundary Events, not all kinds of Event Sub-Processes can be non-interrupting. The Event Sub-Processes that do not have non-interrupting options are Compensation and Error. The remaining Event Sub-Processes that do have non-interrupting options are Conditional, Escalation, Message, Multiple, Multiple Parallel, Signal, and Timer.

## 2.2 Updated Events

A major enhancement to the behavior of Events in BPMN 2 is boundary Events that trigger without interrupting the Activity. In addition, two new types of Events are introduced: Escalation and Multiple Parallel.

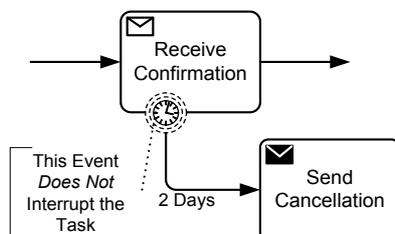
### Non-Interrupting Events

One of the unique characteristics of BPMN 1.x was the innovation of placing Events on the boundaries of Activities (boundary Events) to show that those Activities might be interrupted during their performance (see Figure 8).



**Figure 8: Use of interrupting boundary Events**

However, some business process patterns require the boundary Event to be triggered without interrupting the original Activity. Thus, the Activity would continue being performed while a parallel path is triggered. With BPMN 1.x there are alternative ways of modeling this behavior, but they are complex and require many additional modeling elements. In BPMN 2, the capabilities of boundary Events are upgraded to enable triggering without interrupting the source Activity. An interrupting boundary Event maintains the BPMN 1.x double lined border (see Figure 8, above). A non-interrupting boundary Event has dashed lines for its border (see Figure 9).



**Figure 9: Use of non-interrupting boundary Events**

Some boundary Events do not have a non-interrupting option. These Events are the Cancel, Compensation, and Error Events. The boundary Events that do have a non-interrupting option are Conditional, Escalation, Message, Multiple, Multiple Parallel, Signal, and Timer.

### Escalation Events

Escalation Events are added in BPMN 2. In concept, they are similar to Errors, but are generally less critical. They usually represent a situation requiring human intervention. Unlike Error Events, they do not need to interrupt when attached to Activity boundaries. Escalation Events can be used for Start Events (for Event Sub-Processes), catch and throw Intermediate Events in the main Process flow, boundary Intermediate Events, and End Events (see Figure 10).



**Figure 10: Escalation Events**

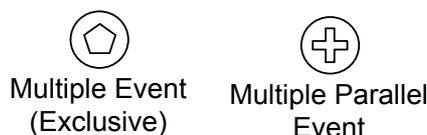
### Multiple Parallel Events

Multiple Events can respond to more than one Event type for a single Event on the diagram. For example, there might be multiple ways to start a Process. Instead of creating a separate Start Event for each of these ways (which is also possible), the modeler might want to have less clutter and combine them into a single Start Event on the diagram. The Multiple Event can respond to any combination of valid Event types, including multiple definitions of the same type (for example, multiple unique Messages).

The Multiple Events behave exclusively. That is, if any one of the defined Event types for the Event is triggered, then it is the only one responded to, and the Process flow will continue from the Event. Any triggering of any other of the Event types would be independent. In the case of Start Events, any new trigger would create a separate Process instance.

In BPMN 2, the Multiple Event capability is expanded to support parallel dependency between the types of Events defined for a single Multiple Event. In this case, *all* of the Event types defined for the Event must be triggered before the Process flow can continue. This variation of the Multiple Event is called the Multiple Parallel Event.

The Multiple Parallel Event marker is distinguished from the Multiple Event and looks like the Parallel Gateway marker (a plus sign—see Figure 11). The plus sign is unfilled to indicate that it is used in “catch” situations.



**Figure 11: Multiple Event types**

Multiple Parallel Events can be used for Start Events, catch Intermediate Events in the main Process flow, and boundary Intermediate Events. They are not used for throw Intermediate Events or End Events, since these types of Events already throw all of the defined Event types in parallel.

### 2.3 Updated Gateways

Most of the changes to Gateways in BPMN 2 are for the Event Gateways. In BPMN 1.x Event Gateways could be set to initiate a Process or not. In BPMN 2 there is a visual differentiation between Event Gateways that initiate a Process and those that do not. The normal Event Gateway, one that does not initiate a Process, maintains the original internal marker that looks like a Multiple Intermediate Event (see left Gateway in Figure 12). The Event Gateway that does initiate a Process now has an internal marker that looks like a Multiple Start Event (see middle Gateway in Figure 12).



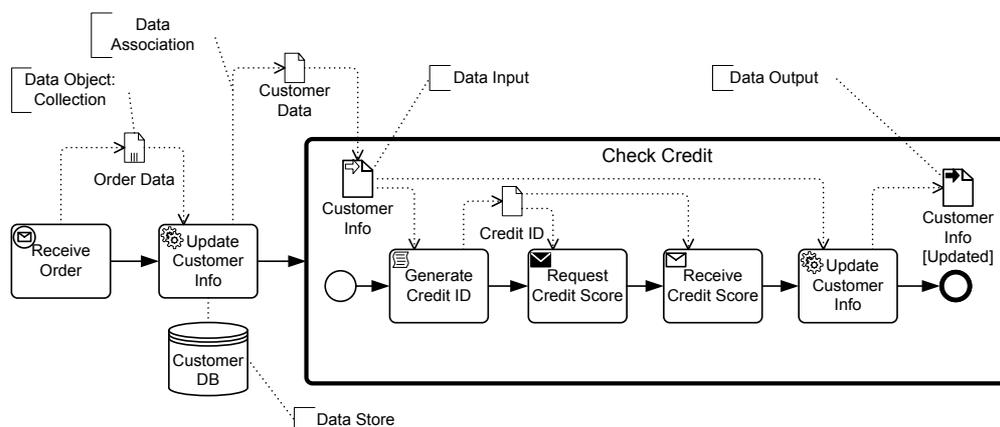
**Figure 12: Event Gateway variations**

Both the initiating and non-initiating versions of the Event Gateway are exclusive. This means that of the multiple Events that follow the Gateway, only one of them

can be triggered each time the Gateway is used. However, to fill the requirements of some business process patterns, a new variation of the Event Gateway is added in BPMN 2—the Multiple Parallel Event Gateway. This Gateway requires that *all* of the Events that follow the Gateway must be triggered before the Process can be initiated (this Gateway is only used for initiating a Process). The internal marker for this variation looks like the new Multiple Parallel Start Event (see right Gateway in Figure 12, above).

## 2.4 Updated Data Elements

In BPMN 1.x data was considered an Artifact and not a main part of the Process flow. While the flow of data is still separated from Sequence Flow, data becomes first class elements in BPMN 2. Many technical and graphical changes are made to how data can be modeled. The technical changes are mainly of interest to the tool implementers and advanced modelers, but there are new graphical data elements, including: Data Input, Data Output, and Data Store (see Figure 13).



**Figure 13: A Process with changes to BPMN data elements highlighted**

### Data Associations

In BPMN 1.x a connector between a Data Object and an Activity was an Association (a dotted line). BPMN 2 transforms these lines into new elements called Data Associations. Associations are still used for connecting Artifacts to elements, such as a Text Annotation to an Activity, but Data Associations are used exclusively between data elements and Activities or Events. They also add mechanisms for the transfer and transformation of data as it passes through the Process. Both Associations and Data Associations use the same dotted line—thus, BPMN 1.x and BPMN 2 diagrams will look the same in this regard (see Figure 13, above).

### Data Inputs and Outputs

Data Inputs and Data Outputs were a part of BPMN 1.x for Activities and Processes. These elements were hidden attributes of the model. In BPMN 2 the Data Inputs and Data Outputs for Processes can be visualized (there is no place to visualize the Data Inputs or Outputs of Tasks). Data Inputs and Data Outputs share the same basic shape as a Data Object, but there are markers in the upper left corner of the shape to distinguish them. A Data Input has an unfilled arrow marker (see the “Customer Info” element in the middle of Figure 13, above). A Data Output has a filled arrow marker (see the “Customer Info [Updated]” element at the right side of Figure 13, above).

### Data Stores

Data Stores are a new element for BPMN 2. They represent basic organizational data stores, repositories, or databases. They are represented in a diagram with the traditional database cylinder shape (see the “Customer DB” element at the left side of Figure 13, above).

### Collections

It is often the case that data elements come grouped together using the same basic structure. For example a set of orders for the same product may be sent at the same time. This is known as a collection, and BPMN 2 adds an attribute to data elements to enable modelers to define them. If a Data Object, Input, or Output is defined as a collection, then the shape will include the same marker that is used for Multi-Instance Activities (see the Data Object “Order Data” at the left side of Figure 13, above).

## 3 INTERACTION MODELS

Modern businesses typically provide services along with products (“solutions”), and partner with other businesses in delivering solutions.<sup>1</sup> For example, producers of cell phones provide mobile services, many products are paired with maintenance contracts, and shipping is usually provided by separate businesses. Services can increase potential markets by providing multiple services on a single product. New services can help differentiate businesses from competitors. They also enable businesses to focus on core value, while partnering for non-core services.

Business services are characterized by interactions between businesses and their customers and partners that are usually agreed or assumed ahead of time, and do not detail the internal activities of the participants. The interactions might be very short from start to end, or take place over a long period. Agreements determine what information, goods, or personnel are needed by whom and at what time, how complaints and unusual situations are handled, and so on. Interactions might be completed in advance, or have details worked out during the interaction, as in case management. Capturing service interactions in diagrams gathers requirements and solution expertise in one place, and facilitates coordination between the parties when the interaction is carried out. This helps lower costs for providers and consumers, reduce unnecessary or unsatisfactory interactions, and identify new areas for service development, including combining and adapting existing services.

This section describes interaction diagrams in BPMN 2, including the new Choreography diagram, improvements to the Collaboration diagram from BPMN 1.x, and enhancements to interactive process modeling. Section 3.1 covers interaction models in BPMN 2. Section 3.2 describes capabilities added for interactive processes needed to deploy interactions.

### 3.1 Interaction Diagrams

BPMN 2 has two diagrams for interactions: Collaboration and Choreography. The first is available in BPMN 1.x, and enhanced in BPMN 2, while the second is in BPMN 2 only. The diagrams show different aspects of interactions, sometimes

---

<sup>1</sup> The term “service” in the section is used in the business sense, rather than a web service or other software operation. Business services can be complicated to deliver, and require more information to specify than simpler software services such as getting stock quotes. Antoine Lonjon provided some of the points in this paragraph.

using different notations for the same concepts, or highlighting some concepts over others. Section 3.1.1 covers the concepts in common to both diagrams, while Sections 3.1.2 and 3.1.3 describe concepts available in only one of the diagrams.

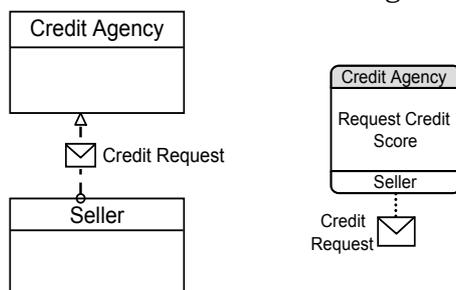
### 3.1.1 Interaction Basics

Interaction diagrams in BPMN 2 have these elements in common:

- Participants are the interacting agents. These might be businesses, departments, or people, for example, or automated agents in software or hardware.
- Messages are sent between Participants. These can be informational or physical, including physical things that do not carry information, such as cars or furniture.
- Messages Flows occur at certain points during the interaction, between particular Participants. The same Message can be carried by more than one Message Flow.

Figure 14 shows the notations for these three basic concepts. Participants in Collaboration diagrams on the left are shown with rectangles (called “pools”), while the Choreography diagram on the right shows them as bands inside a rounded rectangle, called a Choreography Activity. Collaboration diagrams show participants much more prominently than Choreography, so are useful when relationships between Participants are the primary concern.

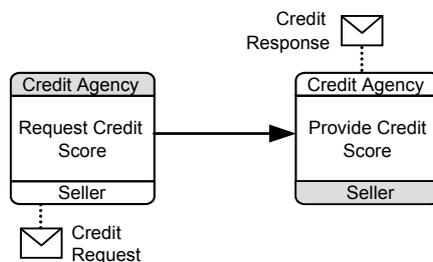
Messages are shown as envelopes on both interaction diagrams, with a label naming them. Message Flows show which Participants exchange the messages. In Collaboration diagrams Message Flows appear as dashed arrows with Messages optionally overlaid on them. In Choreography, Message Flows are shown as Choreography Activities, with Messages linked to them by dotted lines called Associations. The unshaded bands of Choreography Activities are Participants sending the Message, and shaded bands are the ones receiving them.



**Figure 14: Basic Interaction Elements**

### 3.1.2 Message Flow Sequence

Messages usually flow between Participants in a particular order. For example, in many retail purchasing interactions, payment is made before the product is delivered. Choreographies capture this most directly, as Sequence Flows between Choreography Activities (see Figure 15). Sequence Flow arrows indicate that the Message in the Choreography Activity at the tail of the arrow flows before the Message in Choreography Activity at the head. In this example, the Message requesting a credit score is sent before the one providing the credit score. The sending Participant in the first Message Flow is the receiver in the second, capturing a simple request-response interaction. The response may be a very long time after the request, or a very short time, depending on how much processing occurs within the receiver of the first Message.



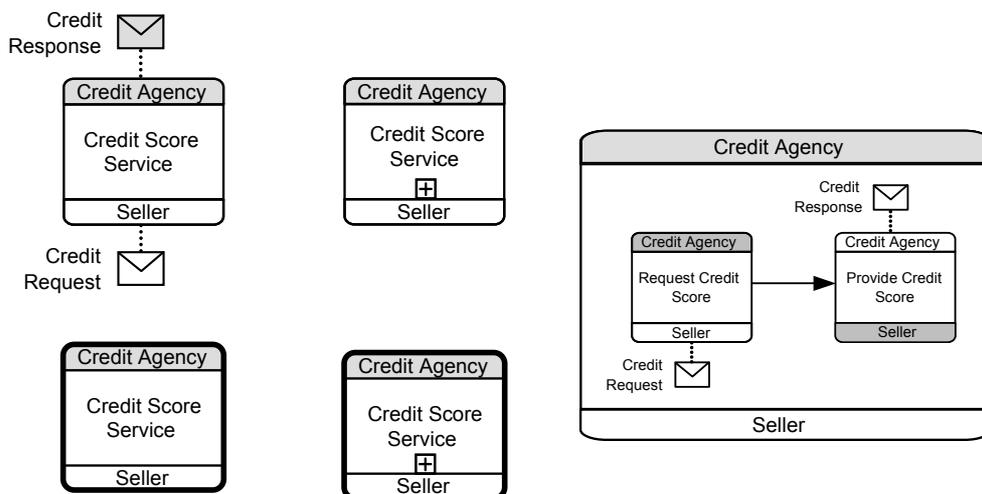
**Figure 15: Message Flow Sequence in Choreography**

Choreography diagrams can use Gateways to split and merge Sequence Flows in similar ways as BPMN Process diagrams, but with some restrictions due to limited Participant visibility. The rule above about Participants in sequences of Choreography Activities applies even when Gateways are present. The sender in a Choreography Activity must also be a Participant in the immediately previous Activity, either as sender or receiver, even if Gateways are interposed between the Activities. For split Gateways, which have a single Sequence Flow coming in and multiple going out, all senders in Activities after the Gateway must participate in the Activity before the Gateway. For merge Gateways, which have multiple Sequence Flows going in and a single one going out, the sender in the Activity after the Gateway must participate in all the Activities before the Gateway. The notation for Gateways in Choreography diagrams is the same as in Process diagrams.

### 3.1.3 Grouping Message Flow

Grouping Message Flows helps manage complicated interactions by gathering multiple flows together under a single element. Choreography diagrams support this with Activities representing multiple Message Flows (see examples on the left in Figure 16). These have two Message Flows, as indicated by the Message icons linked to the Participants sending them on the upper right. In this example the Seller sends a Credit Request Message, and the Credit Agency sends back a Credit Response. The thickness of Choreography Activity borders indicates whether the grouped Message Flows can be nested by multiple Choreography Activities, either in the same Choreography diagram or different ones. A thick border indicates they can be used in multiple Activities, while a thin border indicates they cannot. Thick-bordered Activities are Call Choreography Activities. The one on the lower left in Figure 16 is “calling” a Global Choreography Task that groups Message Flows for nesting by multiple Activities and diagrams. The thin-bordered Activity on the upper left is a Choreography Task, which groups Message Flows without a making them available for multiple Activities or diagrams (they are local to the diagram). Message icons can only be linked to Choreography Tasks. Message icons for receiving Participants are shaded to match the bands to which they are linked.

The two Choreography Activities on the left in Figure 16 do not explicitly capture sequencing of the Message Flows they group, because they do not expand to another Choreography diagram, as indicated by the absence of small plus-sign markers. However, they can represent two Message Flows at most, and the first Message to be sent is identified by linking its icon to the unshaded Participant for Choreography Tasks. These two Activities combine request and response flows between the two Participants, with the request happening first because it is the initiating Participant, and the response happening next, because it is the only one left.

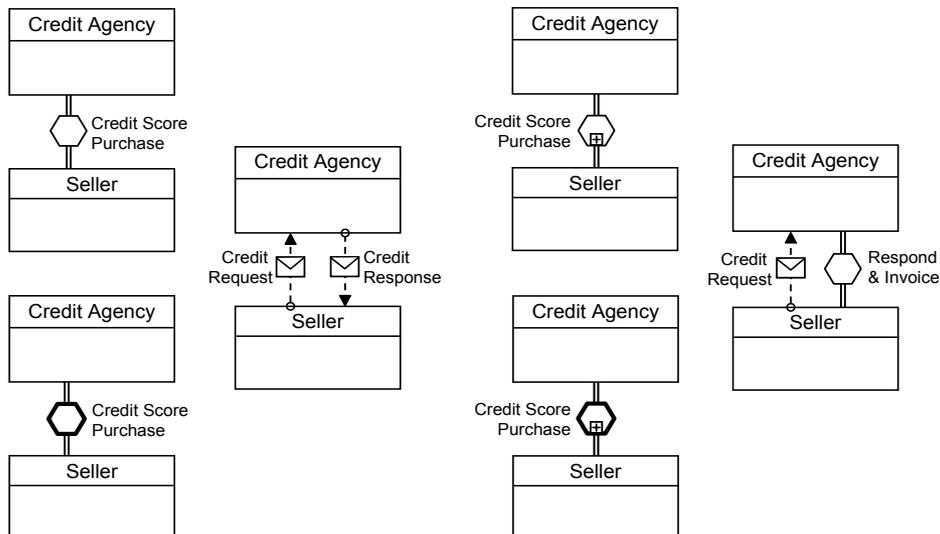


**Figure 16: Grouping Message Flow**

To capture sequencing of grouped Message Flows, Choreography Activities can expand to entire Choreography diagrams, as illustrated in the middle and right in Figure 16. The middle shows the collapsed forms, as indicated by the small plus-sign markers. In the expanded form shown on the right, they contain a full Choreography diagram, including Sequence Flows between nested Choreography Activities. The Activity in the lower middle is calling the Choreography diagram on the right, which is available to be reused in multiple Activities and diagrams. The Activity in the upper middle is a Sub-Choreography, which nests the diagram on the right without making it available for multiple Activities or diagrams.

Collaboration diagrams also support grouping Message Flows to manage complicated interactions, using Conversations to stand in for multiple Message Flows (see examples on the left in Figure 17). Conversations appear as hexagons with solid double lines (“pipelines”) connecting Participants. The examples expand to two Message Flows shown in the middle left, the Seller sending a Credit Request message to the Credit Agency, which sends back a Credit Response. The thickness of Conversation borders indicates whether the grouped Message Flows can be reused in multiple Conversations, either in the same outer Collaboration diagram or different ones. A thick border indicates they can, while a thin border indicates they cannot. Thick-bordered Conversations are Call Conversations. The one on the lower left in Figure 17 is “calling” a Global Conversation that groups Message Flows for reuse by multiple Conversations and diagrams. The thin-bordered Conversation on the upper left groups Message Flows without making them available for multiple Conversations or diagrams. These Conversations do not have a special name, but are analogous to Tasks in Processes and Choreography in the way they group elements.<sup>2</sup> The Conversations on the left in Figure 17 can only group Message Flows, not other Conversations, as indicated by the absence of a small plus-sign adornment.

<sup>2</sup> The hexagon notation in general is technically called a Conversation Node.



**Figure 17: Conversations**

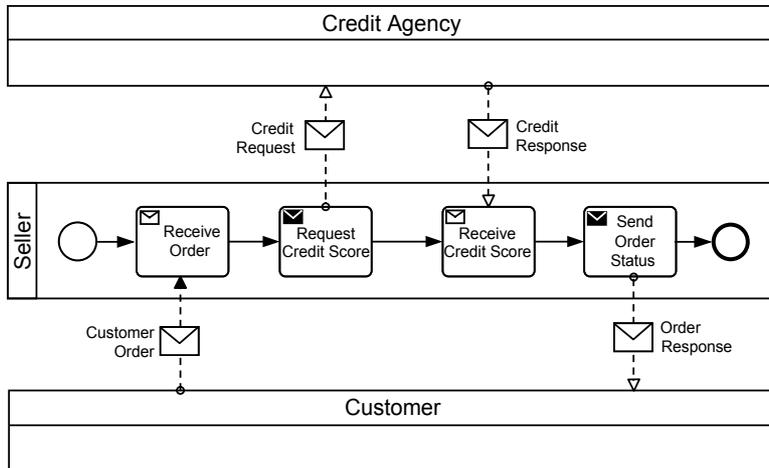
To help manage complex interactions, the expansion of Conversations can include other Conversations as well as Message Flows (see examples on the right side of Figure 17). In collapsed form, these Conversations have a small plus-sign adornment. In expanded form, they appear as Message Flows and Conversations, shown on the far right. The Conversations in the lower middle right is calling the Collaboration diagram on the right, which is available to be nested in multiple Activities and diagrams. The Conversation in the upper middle right is a Sub-Conversation, which nests the diagram on the right without making it available for multiple Activities or diagrams.

### 3.2 Interactive Processes

Interactive Processes send and receive Messages to and from outside Participants. They are needed to deploy the interaction diagrams discussed in the previous section. Section 3.2.1 describes how Collaboration diagrams include interactive Processes within Participants. Section 3.2.2 covers interactive Processes as they appear to other Participants and to the Participants carrying them out.

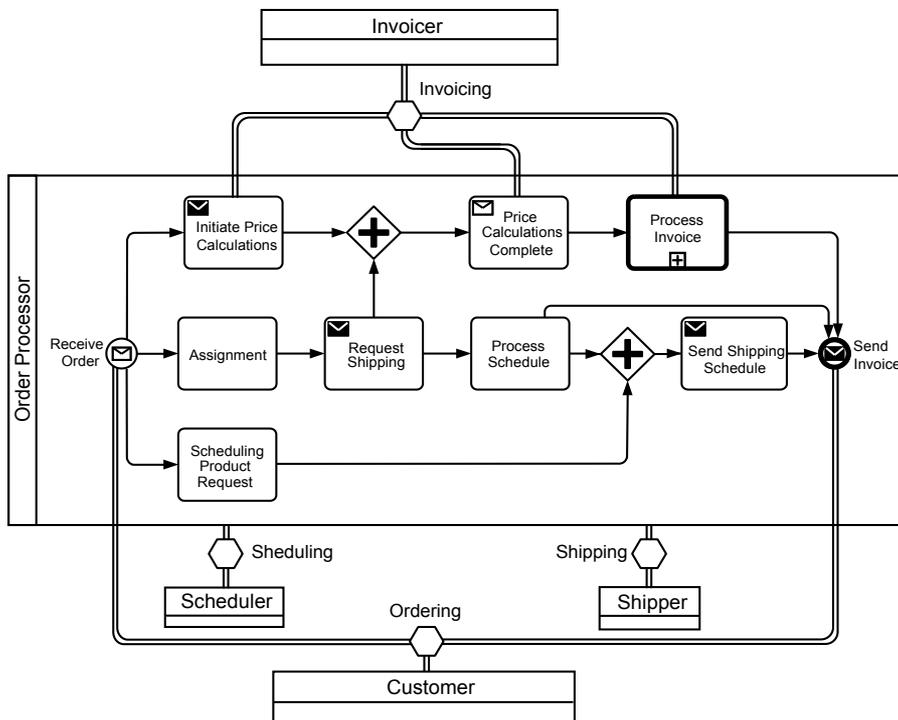
#### 3.2.1 Processes in Collaboration Diagrams

Collaboration diagrams can show how Activities in a Process interact with other Participants, by sending and receiving Messages. Processes usually interact with multiple Participants, serving or producing a product for at least one of them, with assistance from the others. Figure 18 is an example Process where a service is provided to a Customer, with assistance from a Credit Agency. Message Flows show which Tasks interact with which Participants.



**Figure 18: Process in a Collaboration**

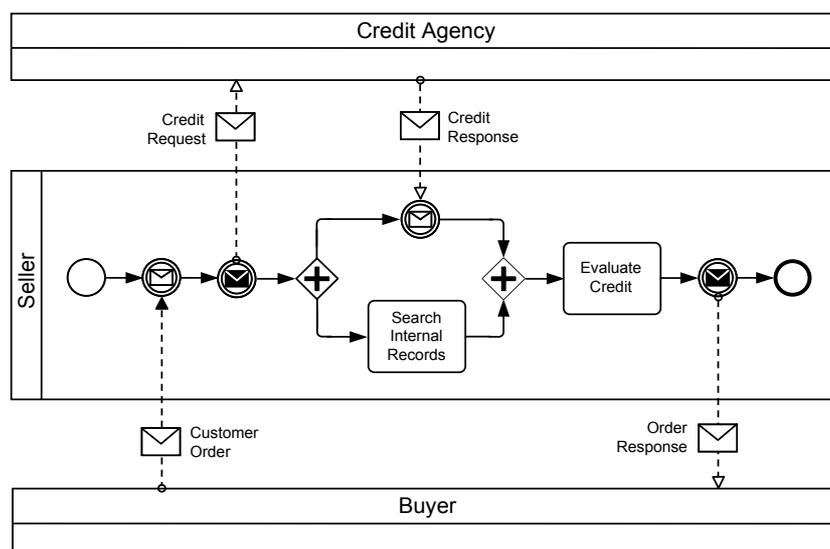
Conversations can link Process Activities and Participants by grouping many Message Flows at lower levels of Process nesting. For example, the Invoicing Conversation in Figure 19 links the Process Invoice Activity to the Invoicer Participant. The Activity calls another Process that exchanges potentially many Messages with the Invoicer (not shown for brevity). Similarly the Scheduling and Shipping Conversations might group many Message Flows to many Activities, but in this case they simplify the diagram by linking directly to the Process Participant, rather than identifying the Messages and Activities involved.



**Figure 19: Process with Conversations**

## 3.2.2 Public and Private Processes

Businesses usually do not show each other their Process models, even interactive ones, unless the models only describe interactions with other businesses. In BPMN 2, these are called public Processes, while those internal to a business are private.<sup>3</sup> For example, Figure 18 could be a public Process, because it only sends and receives Messages, which are Activities other Participants can detect. Figure 20 shows a possible private Process for Figure 18 containing Activities that are not interactive and usually not shown to other Participants. It uses Event notation for receiving and sending Messages, which has the same effect as the Receive and Send Tasks in Figure 18. While the private Process waits for the Credit Response Message, it searches internal records for credit information. A Parallel Gateway is used to receive the Credit Response Message even if it arrives while the internal search is underway. When the Credit Response Message arrives and the internal search is completed, the Process can continue. When the private Process is carried out it will appear to external Participants as if it were the public one in Figure 18, because the same Messages are sent in the same order to the same Participants.



**Figure 20: Private Process for Figure 18**

A private Process supports a public one by interacting the same way as a public one, as described above. The *supports* relationship does not have a notation, but is accessible in other ways with tools supporting Process diagrams. Modelers can specify which private Processes support which public ones, for example, to declare a private Process they developed will cover the public Processes agreed to with partners. Tools might check these declarations, but it is not required for BPMN 2 compliance. The same private Process can support multiple public ones, each showing different aspects of the Private process. For example, two other public Processes can be defined from Figure 18 by removing either Customer or

<sup>3</sup> Public Processes are called “abstract” in BPMN 1.x. The name is changed in BPMN 2 to avoid confusion with Processes that act as templates for further development. Public Processes might be used as templates for private Processes, but the private Processes cannot be changed arbitrarily as those based on templates can. Private Processes must support interactions specified in public Processes (see rest of this section).

Credit Agency. These would only have Message Flows and Tasks for interacting with the remaining Participant, Credit Agency or Customer respectively. The two additional diagrams would be suitable for showing to those particular Participants. The private process in Figure 20 would support both of these public processes.

### 4 CONCLUSION

The new features for processes and interactions in BPMN 2 significantly broaden the range of applications of the language. Process Tasks and Events are more refined, including the use of business rules, and Event Sub-Processes enabling triggers of separate portions of a Process. Events themselves are more expressive, supporting Activity boundary Events without interruption if needed, and Multiple Events that combine events either exclusively or in conjunction with each other. Data becomes a first-class part of Process modeling in BPMN 2. Inputs and outputs are visually specified for Processes and Activities. Collections of Data Objects and Data Stores are also supported.

Interactions and interactive processes undergo a major upgrade in BPMN 2, with Collaborations grouping Message Flows into Conversations, a new Choreography diagram for specifying sequences and conditions for Message Flow, and explicit relationships between public and private interactive Processes. Collaborations show Participants more prominently, while Choreography shows sequencing of Message Flows directly. Message flow to Activities in interactive Processes can be grouped into Conversations, aligning with grouping of Activities in Processes. Public and private interactive processes can be linked to indicate which private processes support which public ones.

### 5 ACKNOWLEDGEMENTS

The authors thank Antoine Lonjon, Peter Denno, and J.D. Baker for their input to this paper.

Commercial equipment and materials might be identified to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the U.S. National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

### 6 REFERENCES

- [1] Object Management Group, "Business Process Model and Notation (BPMN), Version 2.0, Beta 2," <http://doc.omg.org/dtc/10-06-04>, June 2010.
- [2] Organization for the Advancement of Structured Information Standards, "Web Services Business Process Execution Language Version 2.0," April, 2007.
- [3] U.S Department of Defense, "DoD Architecture Framework Version 2.0," May 2009.