Business Process Model and Notation 2.0

Tutorial for the BPMN 2.0 RFP Response (OMG Document bmi/2009-05-03)

June 9, 2008

Agenda

- BPMN 2.0 Background
  - Facts
  - Scope
  - New Features
- BPMN 2.0 Structure
- BPMN Modeling
  - Processes
  - Interactions
- Diagram Interchange
- Questions

What is BPMN?

- BPMN is a notation and model for defining Business Processes and Interactions

- BPMN encompasses:
  - Graphical notation
  - Semantic metamodel and accompanying interchange format
  - Visual model and accompanying interchange format
  - Refined and formalized BPMN execution semantics

- BPMN can be used for modeling at different levels of abstraction
  - From very high level, to detailed executable Processes, depending on applied methodologies
  - The BPMN specification is an reference for vendor implementations and is not written for end users

BPMN 2.0: Facts

- May 25, 2009 submission
  - Specification http://www.omg.org/cgi-bin/doc?bmi/09-05-03
  - XMI Schemas: http://www.omg.org/cgi-bin/doc?bmi/09-05-04
  - XSD Schemas: http://www.omg.org/cgi-bin/doc?bmi/09-05-05
  - XSLT Transformations: http://www.omg.org/cgi-bin/doc?bmi/09-05-06
- Submitters and supporters
  - Axway, Accenture, Active Endpoints, Adaptive, BizAgi, Bruce Silver Associates, Capgemini, Enterprise Agility, France Telecom, IDS Scheer, Insubria University, Intalio, *IBM*, MEGA International, Metastorm, Model Driven Solutions, Nortel, *Oracle*, Red Hat Software, *SAP AG*, Software AG, TIBCO Software, *Unisys*, Vangent
- IPR mode
  - Intention to contribute this work on an RF on RAND

BPMN 2.0: Scope

- Metamodel
  - Formalizes the implicit metamodel of BPMN 1.1
  - Intended to model the abstract syntax
- Notation
  - Based on BPMN 1.1, with new features
  - Based on end-user modeling requirements and needs (“Top-Down”)
  - Has capabilities to model non-executable elements or processes
  - Through a combination of graphical and supporting elements, allows a model to be populated with sufficient information to generate executable processes
- Semantics
  - Refine and formalize BPMN execution semantics
  - Semantics for interaction models and public processes are defined
- XMI-based and XSD-based Interchange formats
  - Support MDA and non-MDA tools
- Visual model and diagram interchange format
  - Based on an ongoing work at OMG (that is the Diagram Definition RFP and related proposal – to be synchronized during the FTF)
- Mapping of a BPMN subset to WS-BPEL
  - Demonstrates alignment with existing technologies and standards

Conformance

Category	Process Modeling Conformance	Process Execution Conformance	BPEL Process Execution Conformance	Choreography Conformance
Visual BPMN Diagrams	Process diagrams, Collaboration diagrams depicting collaborations among Processes and Conversation diagrams	N/A	N/A	Choreography diagrams and Collaboration diagrams augmented with Choreographies
BPMN Diagram Elements	All Process elements, Collaboration elements, and Conversation	N/A	N/A	All Choreography, Conversation and Collaboration elements.
Interchange of domain models	Process models and Collaboration models depicting collaboration among Processes.	Process models	Process models	Choreography model and Collaboration models augmented with Choreographies
Graphical syntax and semantics	Process diagrams, Collaboration diagrams depicting collaboration among Processes and Conversation diagrams	N/A	N/A	Process diagrams. Conversation diagrams, and Collaboration diagrams augmented with Choreographies.
Execution Semantics	N/A	Process semantics	Process semantics	Choreography semantics

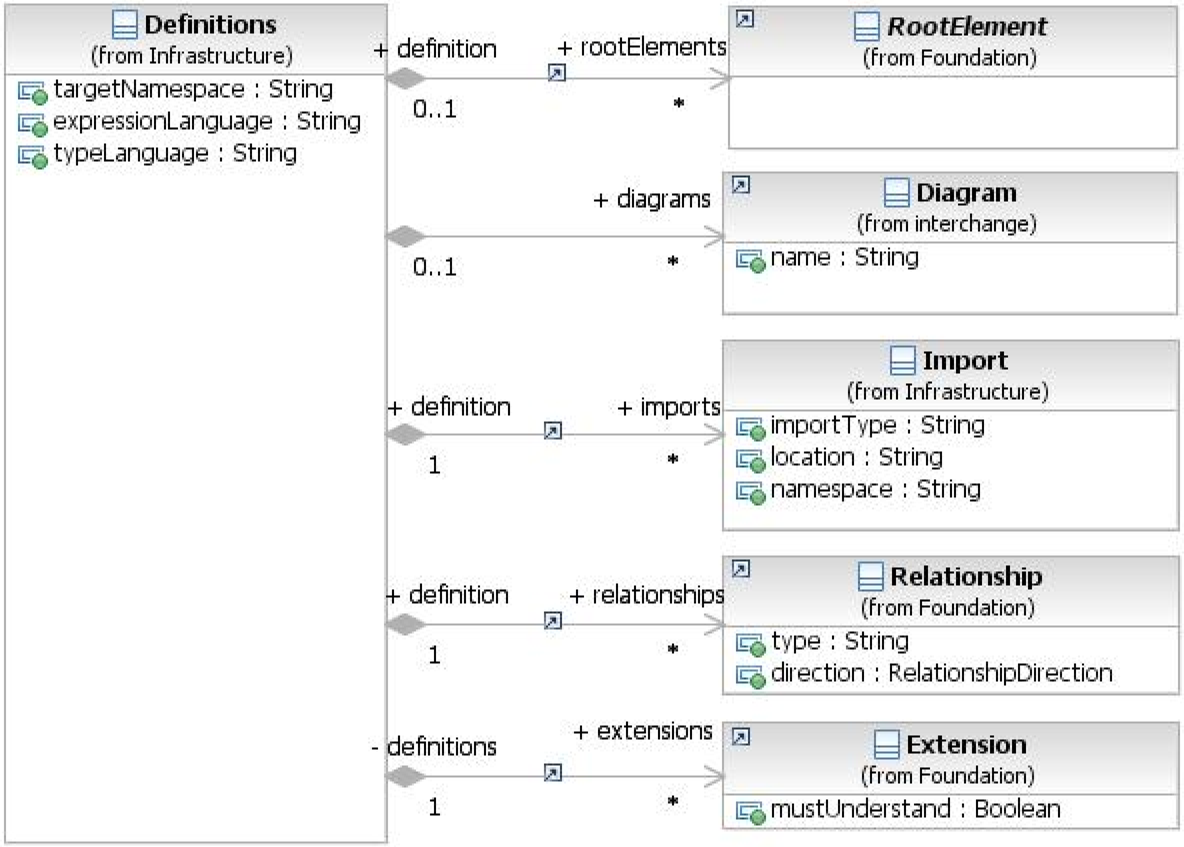
New Features for Version 2.0

- New Process Elements
  - Non-Interrupting Events
  - Event Sub-Processes
  - Markers for Tasks
  - Defined Human Interactions
  - Service Model
  - Data Inputs/Outputs/Association/Collections/Stores
  - Parallel Multiple Events
- New Interactions diagrams (Choreography and Conversation)
- Formalized Execution Semantics
- Formalized metamodel
- Semantic Interchange and Diagram Interchange
  - XML and XMI Schemas (and XSLT transformation)

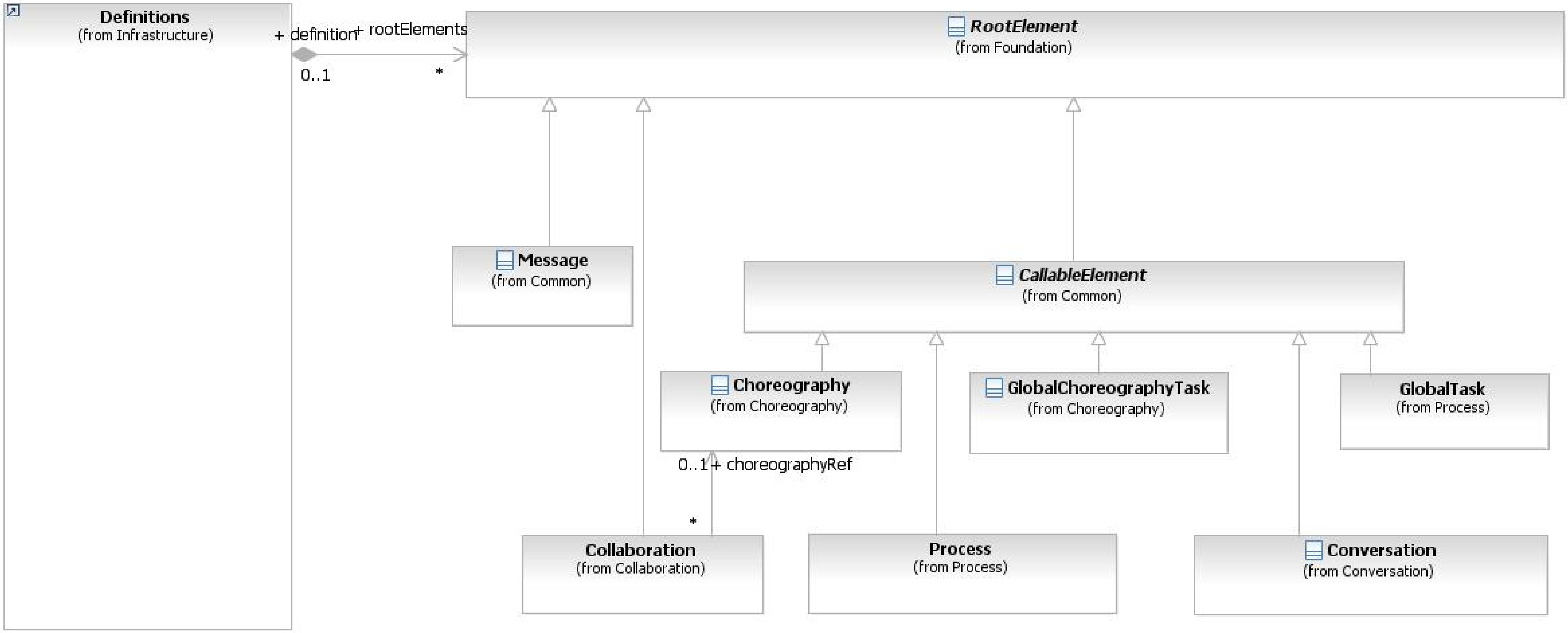
Agenda

- BPMN 2.0 Background
  - Facts
  - Scope
  - New Features
- BPMN 2.0 Structure
- BPMN Modeling
  - Processes
  - Interactions
- Diagram Interchange
- Questions

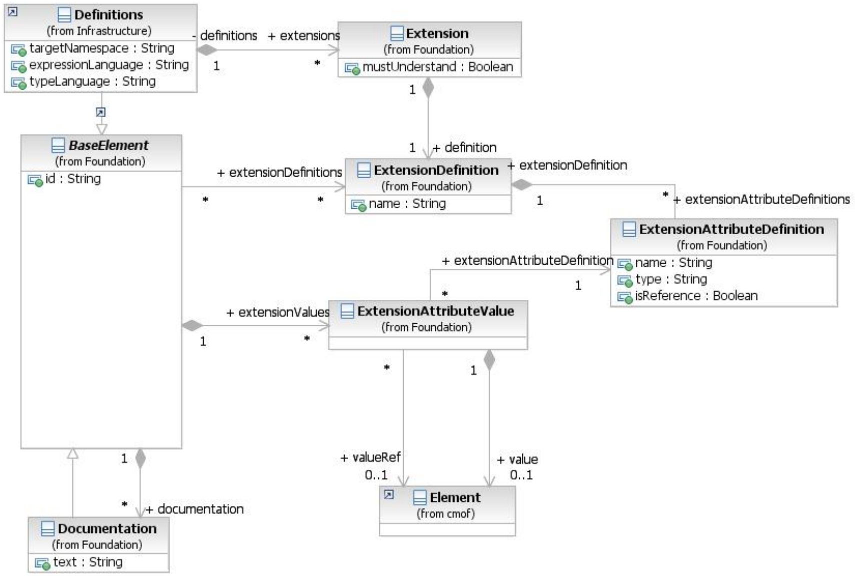
Basic Structure of BPMN 2.0 Models

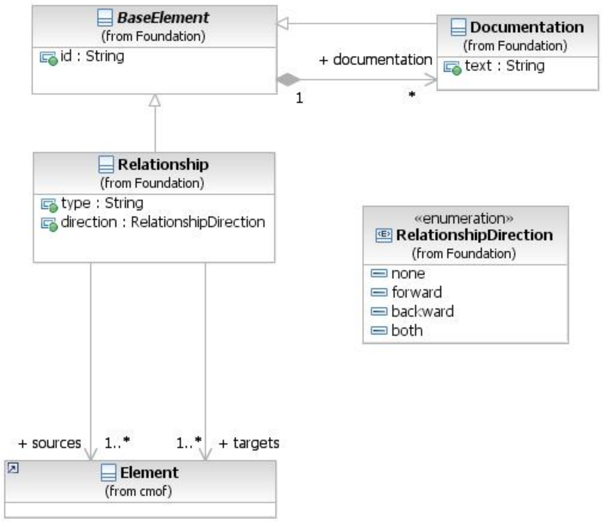


Basic Structure of BPMN 2.0 Models, cont.



Extensibility & Relationships





- The BPMN metamodel is aimed to be extensible. This allows BPMN adopters to extend the specified metamodel in a way that allows them to be still BPMN-compliant
- The ‘identity/relationship’ model enable BPMN and non-BPMN Artifacts to be related in non intrusive manner

Agenda

- BPMN 2.0 Background
    - Facts
    - Scope
    - New Features
- BPMN 2.0 Structure
- BPMN Modeling
    - Processes
    - Interactions
- Diagram Interchange
- Future

Process Modeling

- Process Basics
- Process metamodel
- Task Markers
- Non-Interrupting Events
- Event Sub-Processes
- Data Modeling
- Service Modeling
- Human Interactions

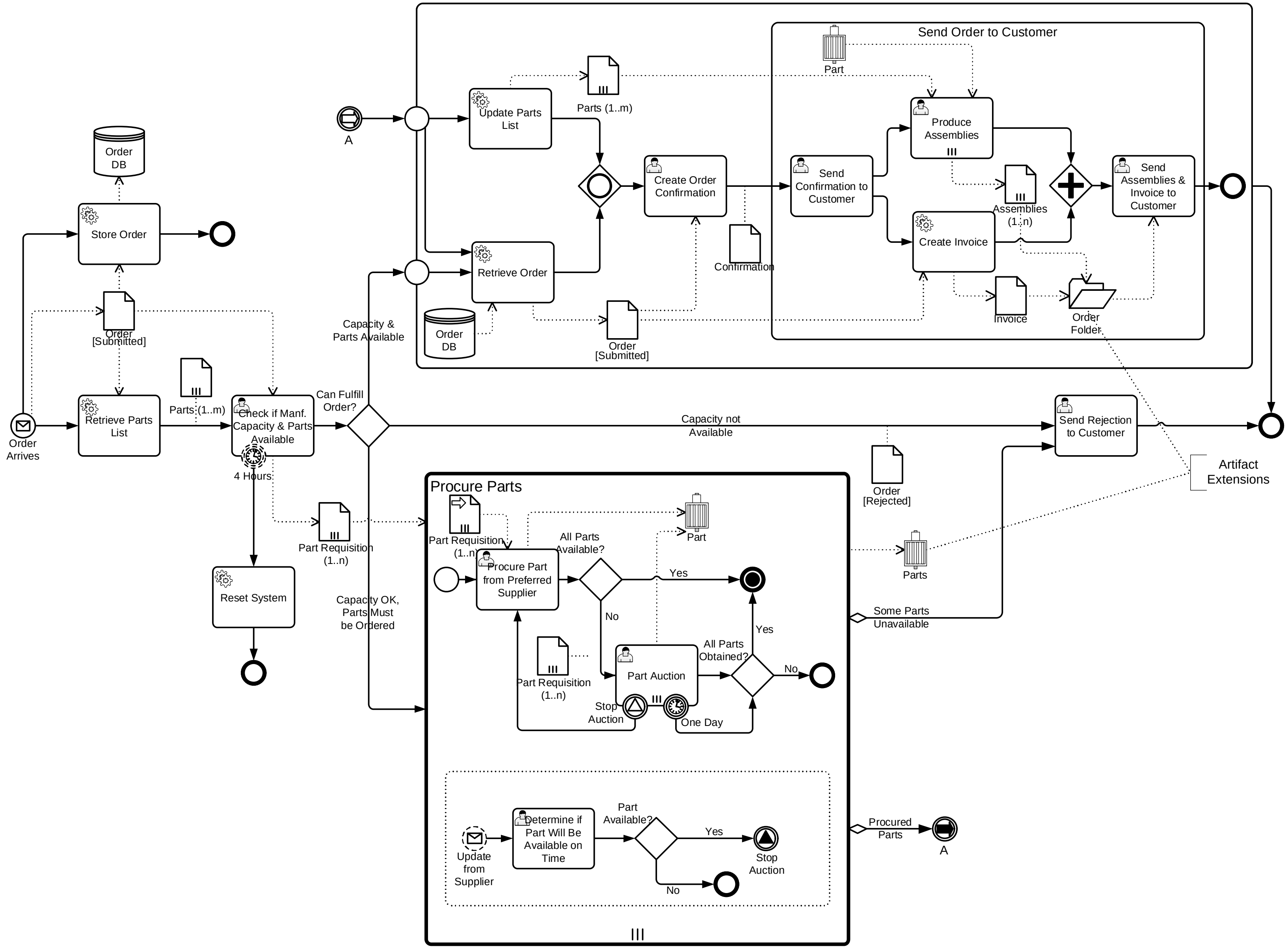
Process Basics

- A Process describes a sequence or flow of Activities in an organization with the objective of carrying out work
- A Process is depicted as a graph of Flow Elements, which are a set of Activities, Events, Gateways, and Sequence Flow
- Process definition allows different Process types:
  - Private non-executable (internal) business processes
  - Private executable (internal) business processes
  - Public processes
- The execution semantics for process elements formalized
  - Operational elements clearly and precisely defined
  - Non-operational elements may be extend; implementations may extend the semantics to make them executable

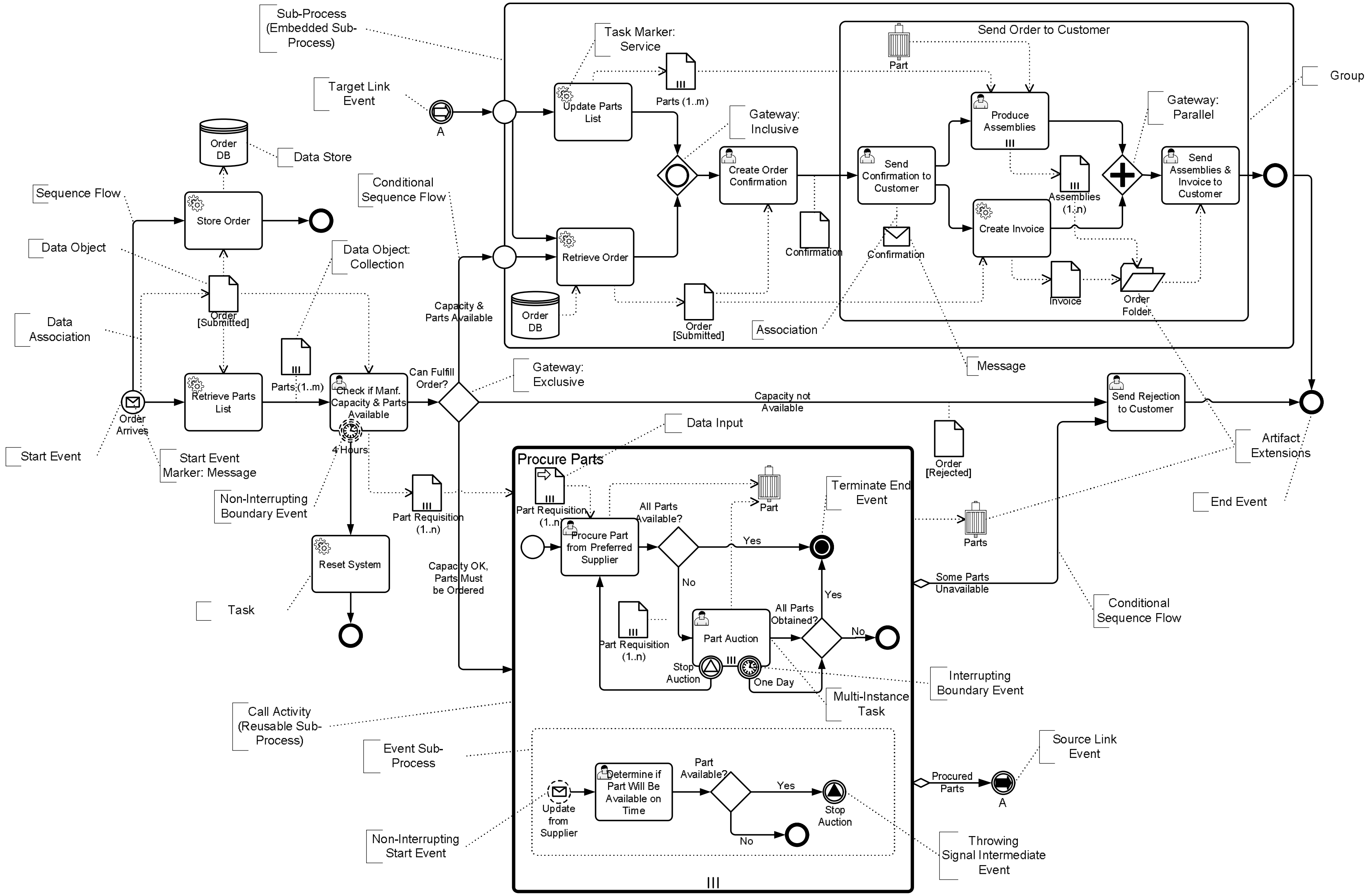
Process Basics

- Out of scope:
  - Definition of organizational models
  - Modeling of functional breakdowns
  - Data and information models
  - Modeling of strategy
  - Modeling of business rules

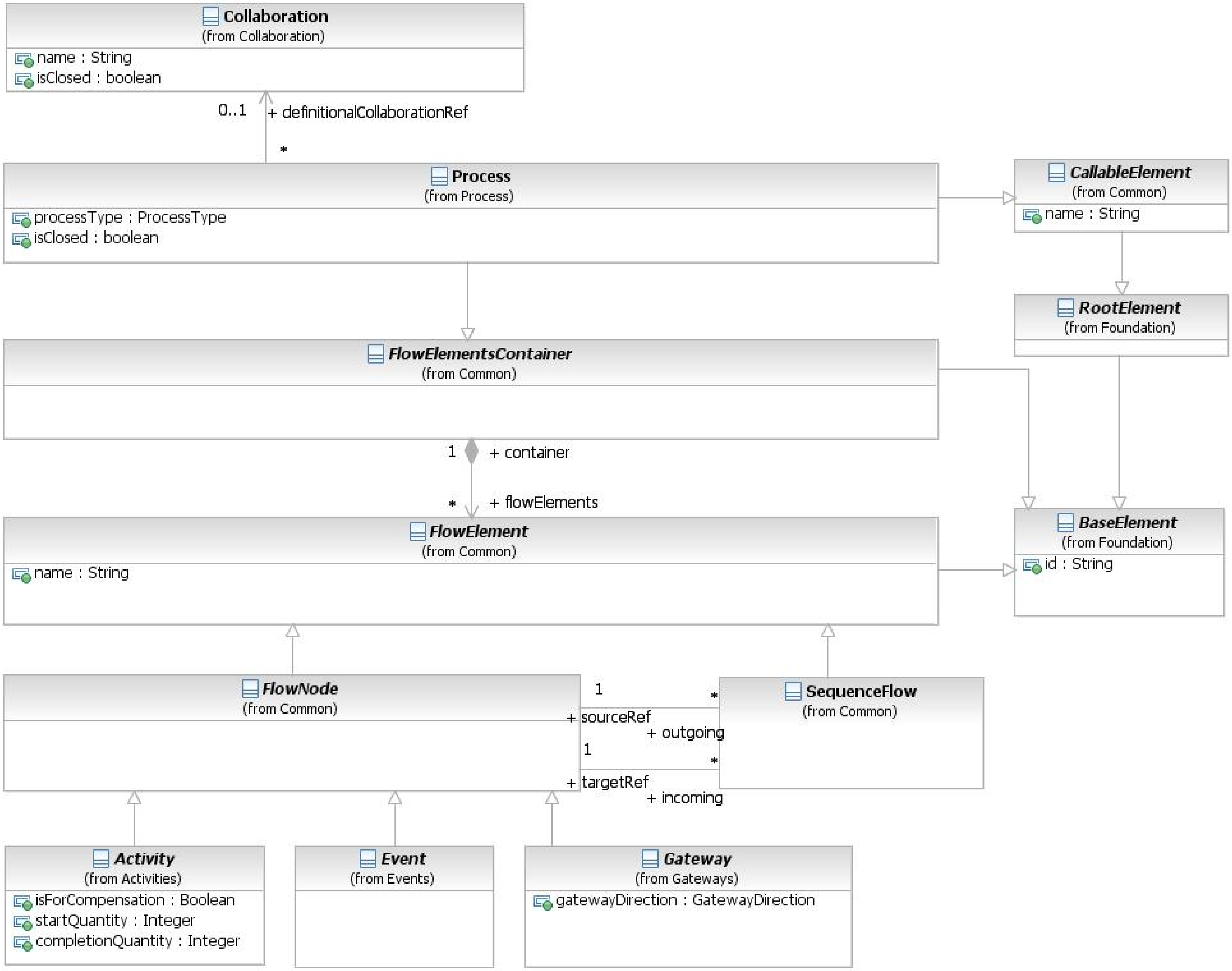
Process Diagram



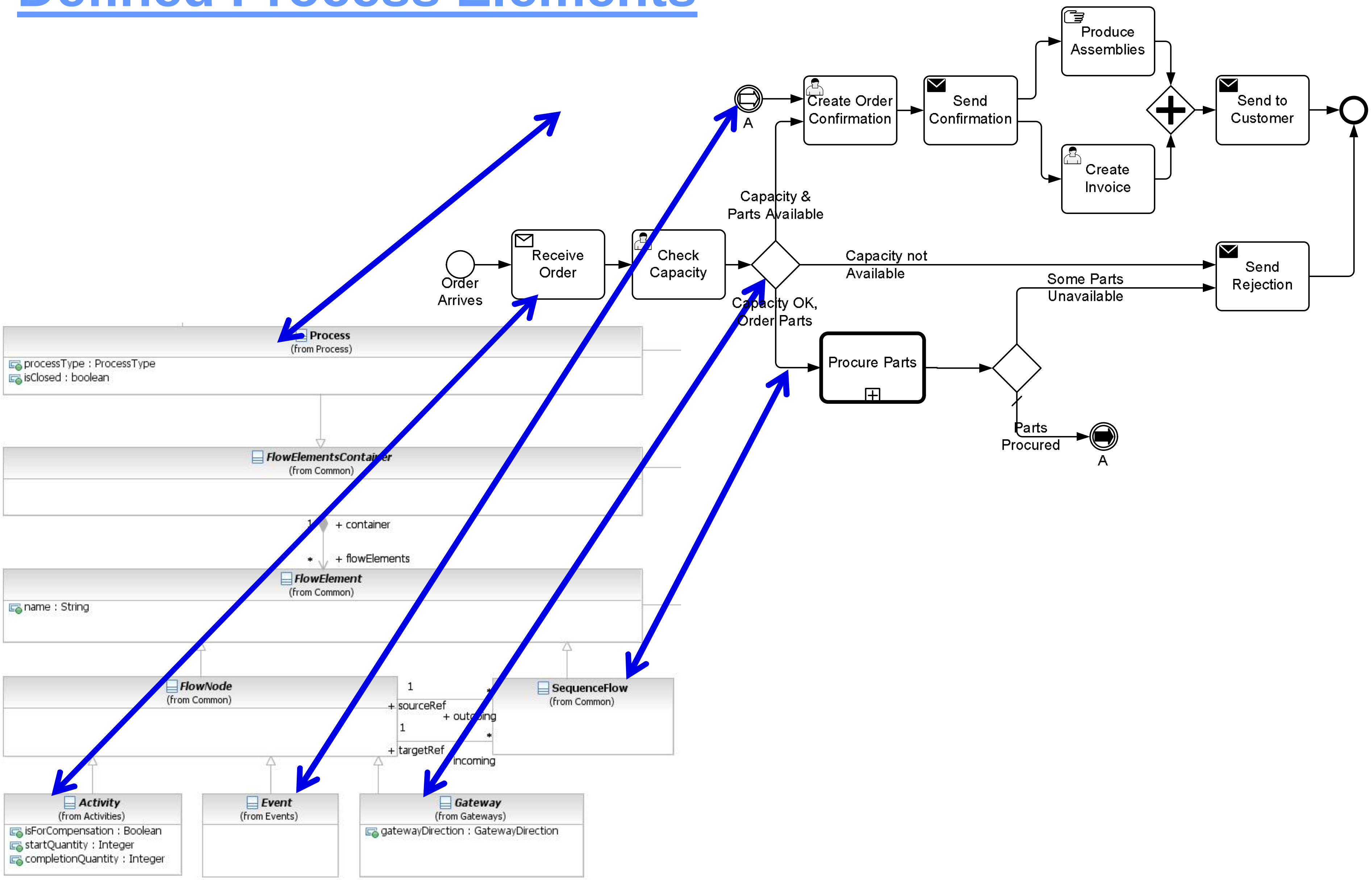
Process Diagram



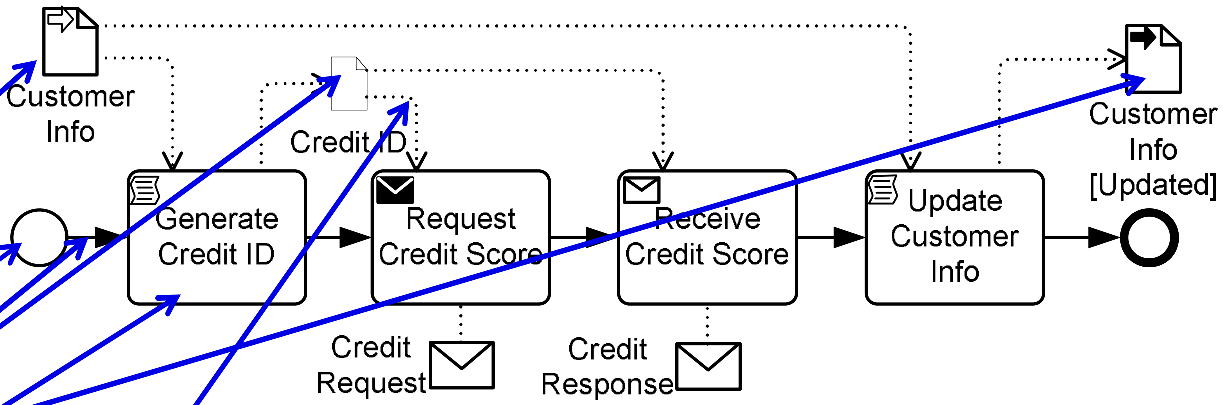
Process Metamodel



Defined Process Elements



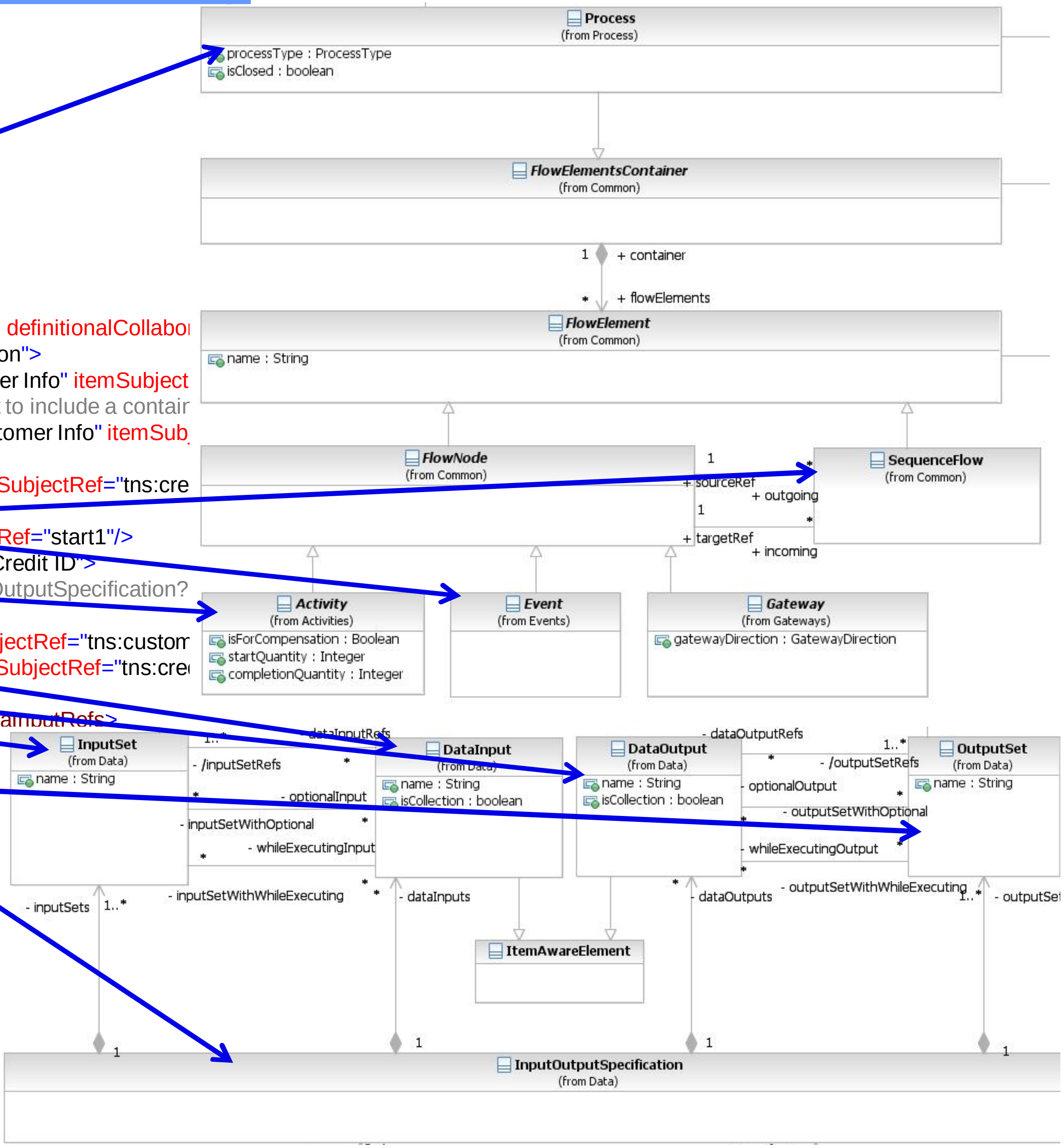
XSD of a Process



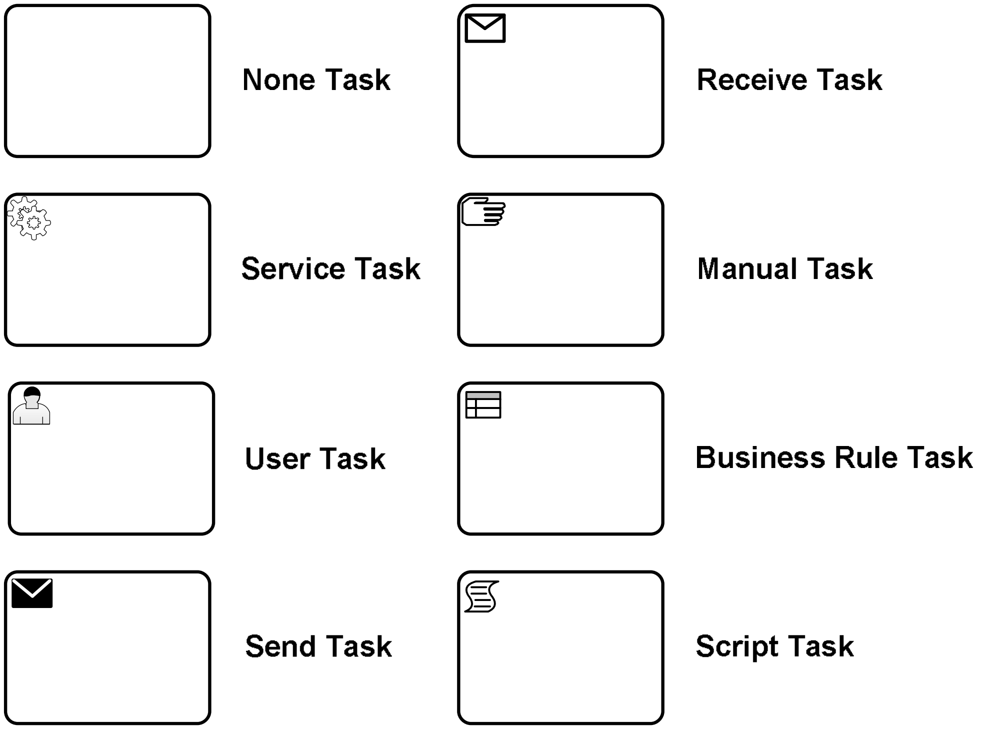
```
<process id="creditCheckProcess" name="Credit Check" definitionalCollaborationRef="creditCheckCollaboration">
  <ioSpecification id="creditCheckProcessIOSpecification">
    <dataInput id="customerInfoInput" name="Customer Info" itemSubjectRef="tns:customerInfoStructure"/>
    <!--Update Schema to DataInput and DataOutput to include a containment of DataState (now is only a reference) So it is removed for now -->
    <dataOutput id="customerInfoOutput" name="Customer Info" itemSubjectRef="tns:customerInfoStructure"/>
  </ioSpecification>
  <dataObject id="creditIDData" name="Credit ID" itemSubjectRef="tns:creditIDStructure"/>
  <startEvent/>
  <sequenceFlow targetRef="generateCreditID" sourceRef="start1"/>
  <scriptTask id="generateCreditID" name="Generate Credit ID">
    <!--Issue? Shouldn't this element be named inputOutputSpecification?-->
    <ioSpecification>
      <dataInput id="generateCreditIDInput" itemSubjectRef="tns:customerInfoStructure"/>
      <dataOutput id="generateCreditIDInputOutput" itemSubjectRef="tns:creditIDStructure"/>
      <inputSet>
        <dataInputRefs>generateCreditIDInput</dataInputRefs>
      </inputSet>
      <outputSet>
        <dataOutputRefs>generateCreditIDInputOutput</dataOutputRefs>
      </outputSet>
    </ioSpecification>
    <dataInputAssociation>
      <sourceRef>customerInfoInput</sourceRef>
      <targetRef>generateCreditIDInput</targetRef>
    </dataInputAssociation>
    <dataOutputAssociation>
      <sourceRef>creditIDData</sourceRef>
      <targetRef>generateCreditIDInputOutput</targetRef>
    </dataOutputAssociation>
  </scriptTask>
```

Metamodel to XSD Mapping

```
<process id="creditCheckProcess" name="Credit Check" definitionalCollabo
   <ioSpecification id="creditCheckProcessIOSpecification">
      <dataInput id="customerInfoInput" name="Customer Info" itemSubject
      <!--Update Schema - to DataInput and DataOutput to include a contair
      <dataOutput id="customerInfoOutput" name="Customer Info" itemSub
   </ioSpecification>
   <dataObject id="creditIDData" name="Credit ID" itemSubjectRef="tns:cre
   <startEvent/>
   <sequenceFlow targetRef="generateCreditID" sourceRef="start1"/>
   <scriptTask id="generateCreditID" name="Generate Credit ID">
      <!--Issue? Shouldn't this element be named inputOutputSpecification?
      <ioSpecification>
         <dataInput id="generateCreditIDInput" itemSubjectRef="tns:custom
         <dataOutput id="generateCreditIDOutput" itemSubjectRef="tns:cre
         <inputSet>
            <dataInputRefs>generateCreditIDInput</dataInputRefs>
         </inputSet>
         <outputSet>
            <dataOutputRefs>generateCreditIDOutput<
         </outputSet>
      </ioSpecification>
      <dataInputAssociation>
         <sourceRef>customerInfoInput</sourceRef>
         <targetRef>generateCreditIDInput</targetRef>
      </dataInputAssociation>
      <dataOutputAssociation>
         <sourceRef>creditIDData</sourceRef>
         <targetRef>generateCreditIDInputOutput</targ
      </dataOutputAssociation>
   </scriptTask>
```

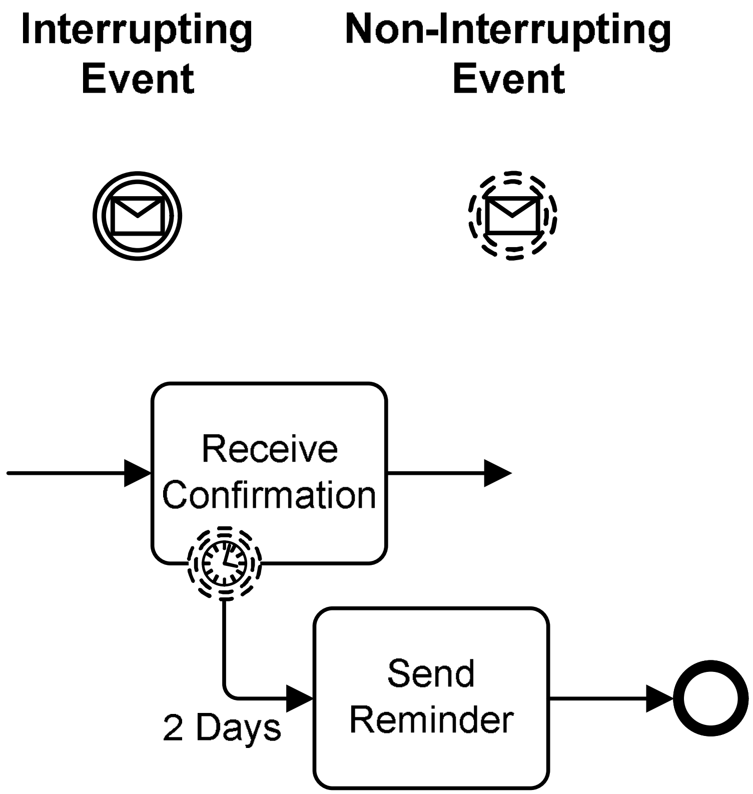


Task Markers



- A **Task** is an atomic activity that is included within a Process. A Task is used when the work in the Process is not broken down to a finer level of Process Model detail
- Specialized Tasks now have standard markers to indicate their type

Non-Interrupting Events



Interrupting Event

Non-Interrupting Event

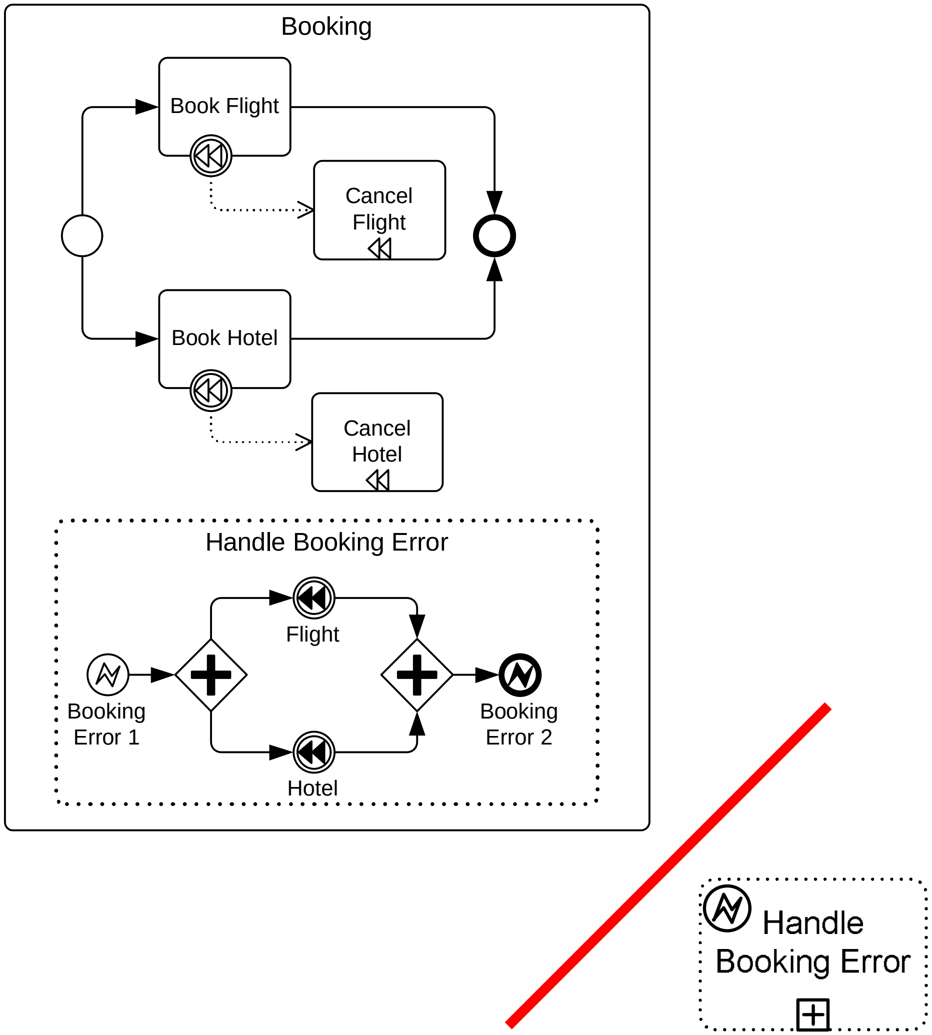
Receive Confirmation

2 Days

Send Reminder

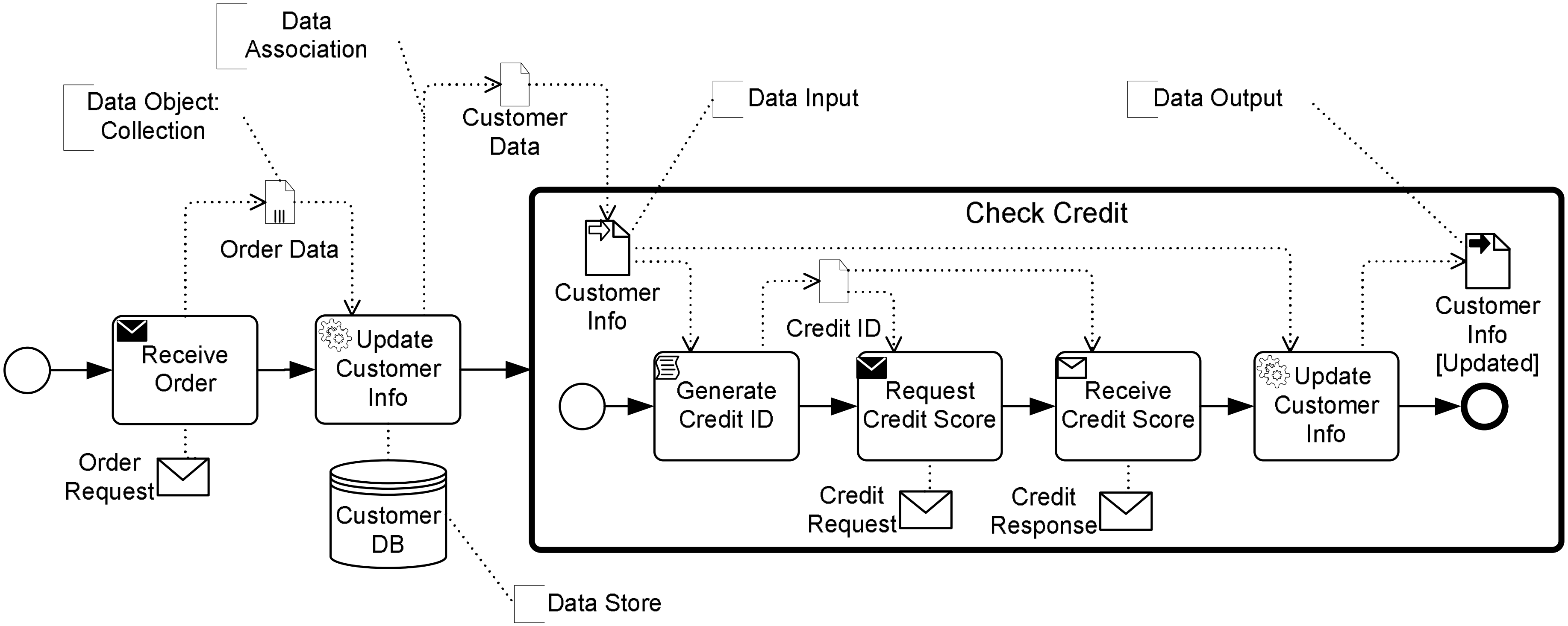
- An **Event** is something that “happens” during the course of a business process. These Events affect the flow of the Process and usually have a trigger or a result. They can start, interrupt, or end the flow
- Events attached to Activity boundaries can now be set to interrupt the Activity or not
  - For non-interrupting Events the Activity will continue and the flow will continue out the Event
- Non-interrupting Events have a dashed boundary

Event Sub-Processes



- A **Sub-Process** is a compound activity that is included within a Process. It is compound in that it can be broken down into a finer level of detail (a Process) through a set of sub-activities
- **Event Sub-Process** are included in a Process and only happen if its Start Event is triggered
- They can interrupt the Process or not
  - Non-interrupting can happen multiple times (the Start Events will have a dashed boundary)
- Collapsed Event Sub-Processes show the Start Event as a marker

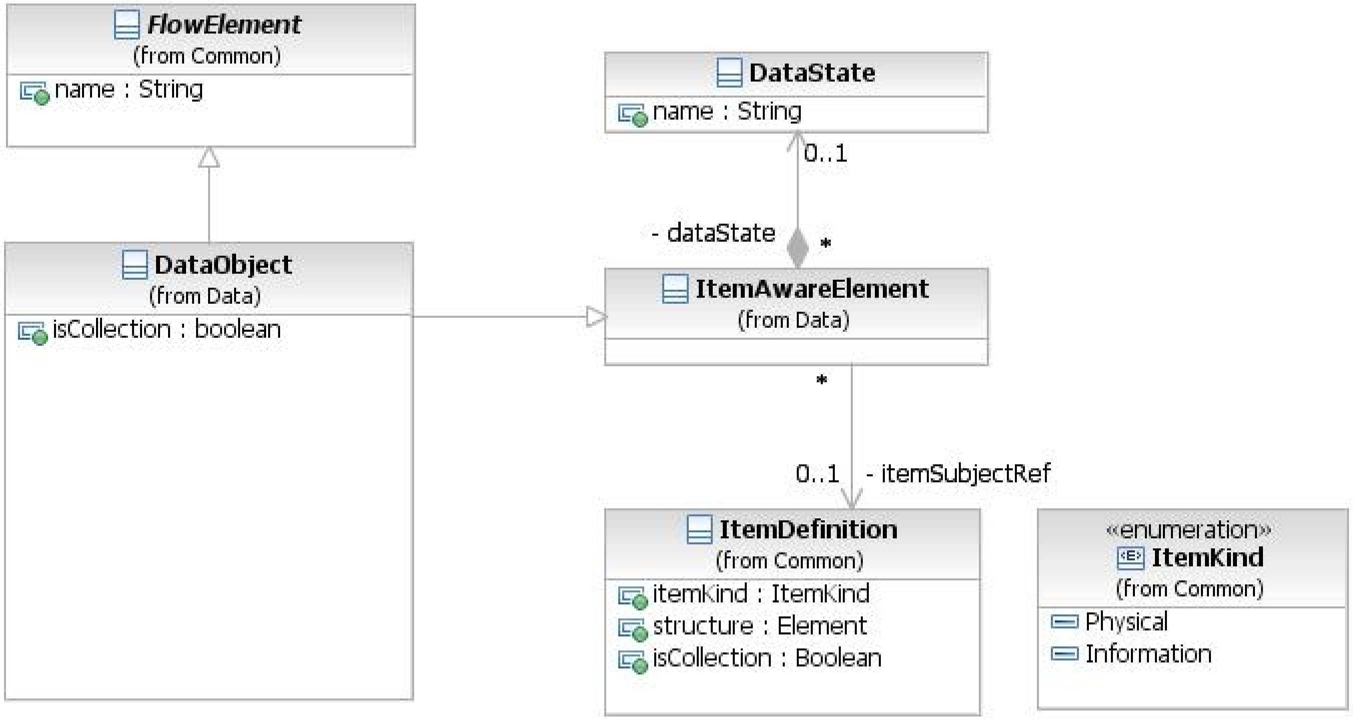
New Data Elements



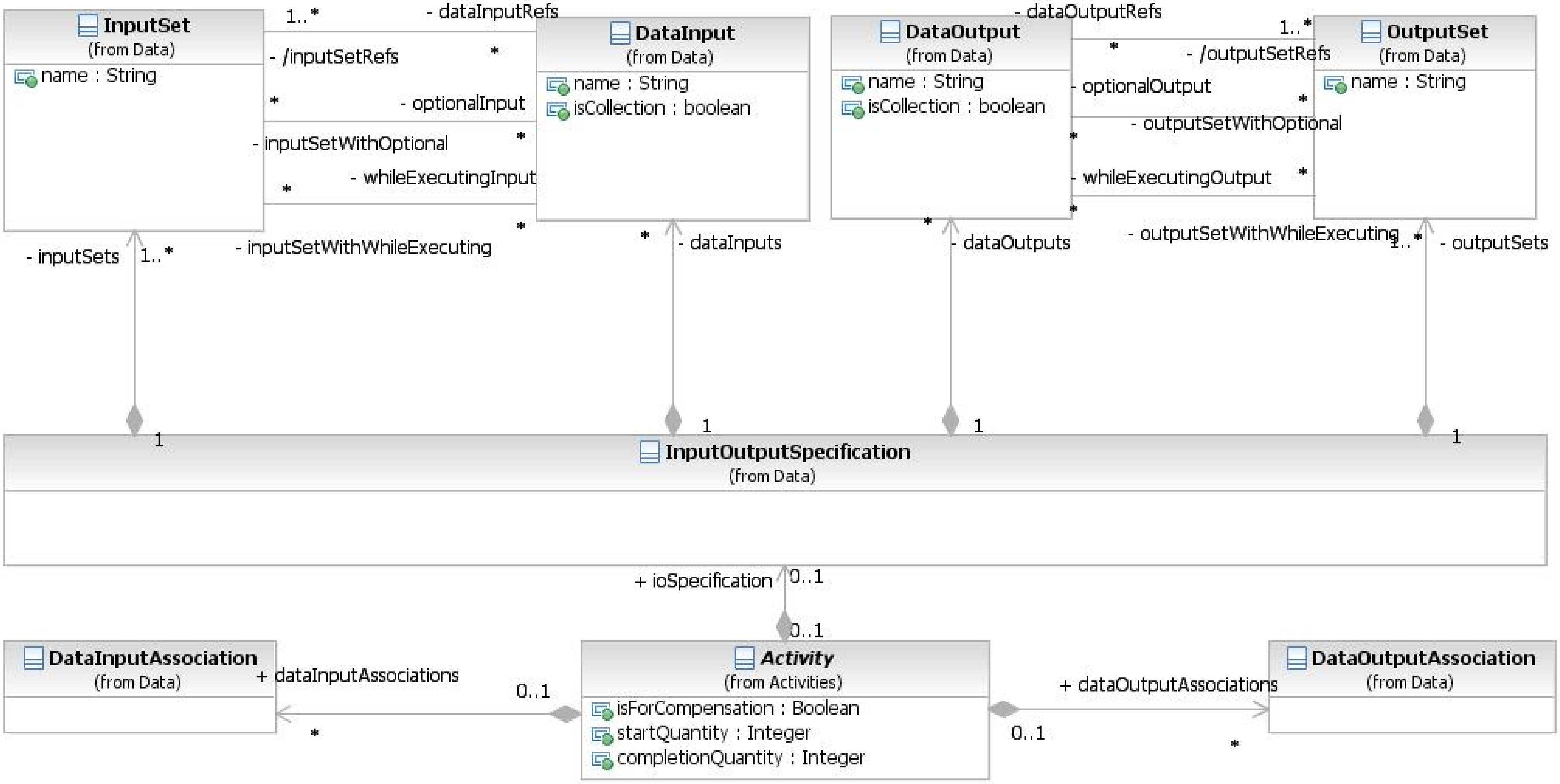
- The use of Data was upgraded in BPMN 2.0
  - Data now first class elements
  - Inputs & Outputs to a Process shown
  - Data Associations defined
  - Data Objects can be collections
  - Data Objects can be physical or information
  - Data Store added

Data Flow Modeling

Data Object Model

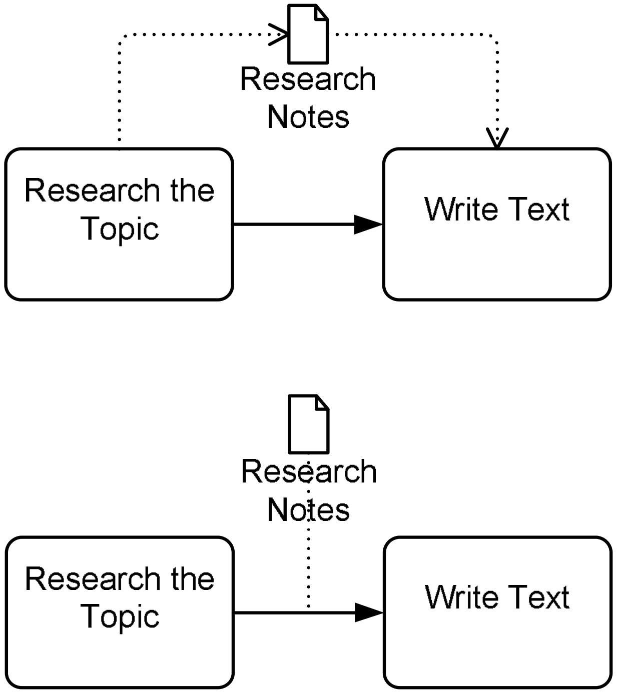


Defining Activity Inputs and Outputs



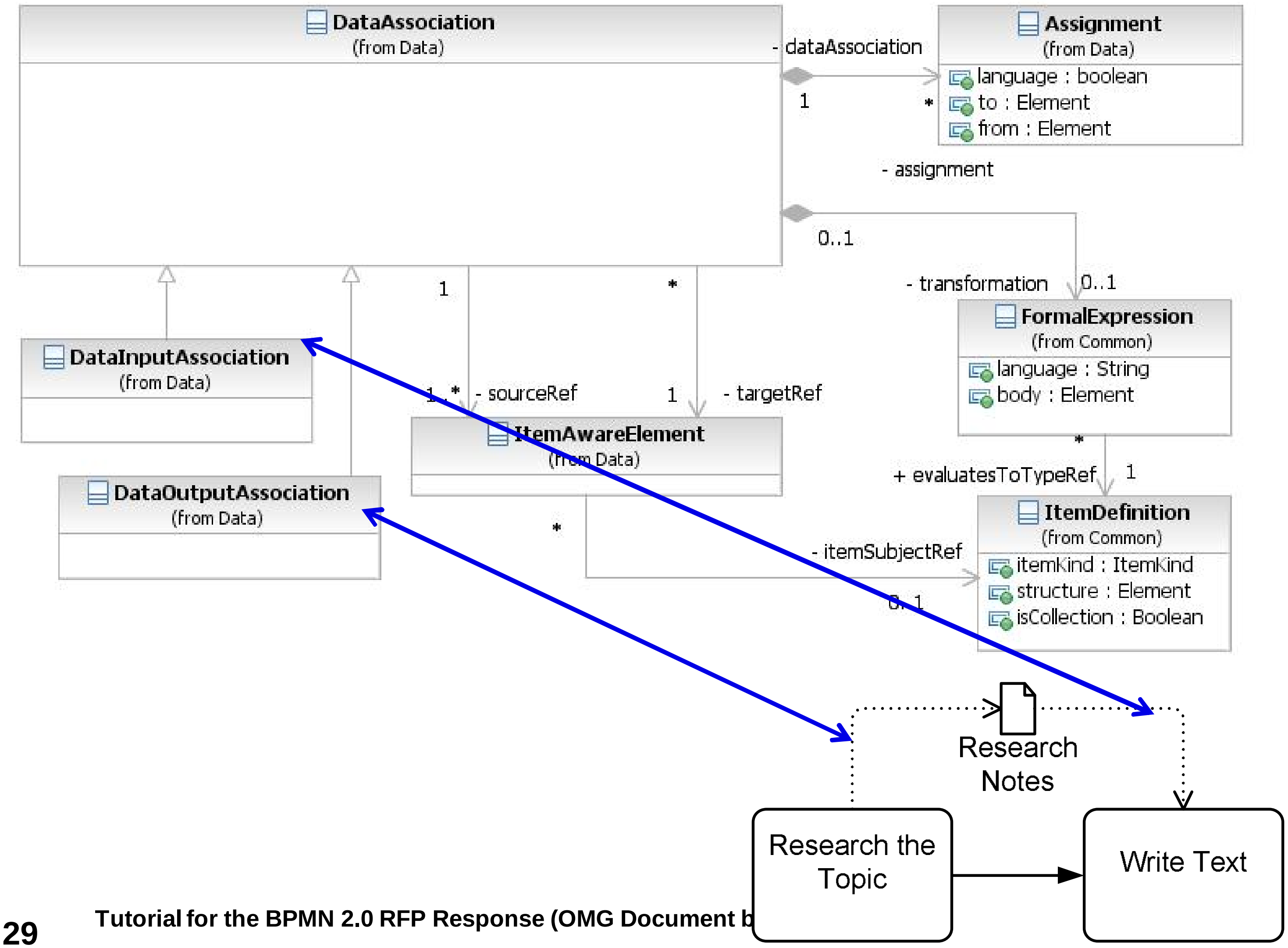
Data Flow Modeling (cont.)

Example: A `Data Association` used for an output and an input into Activities



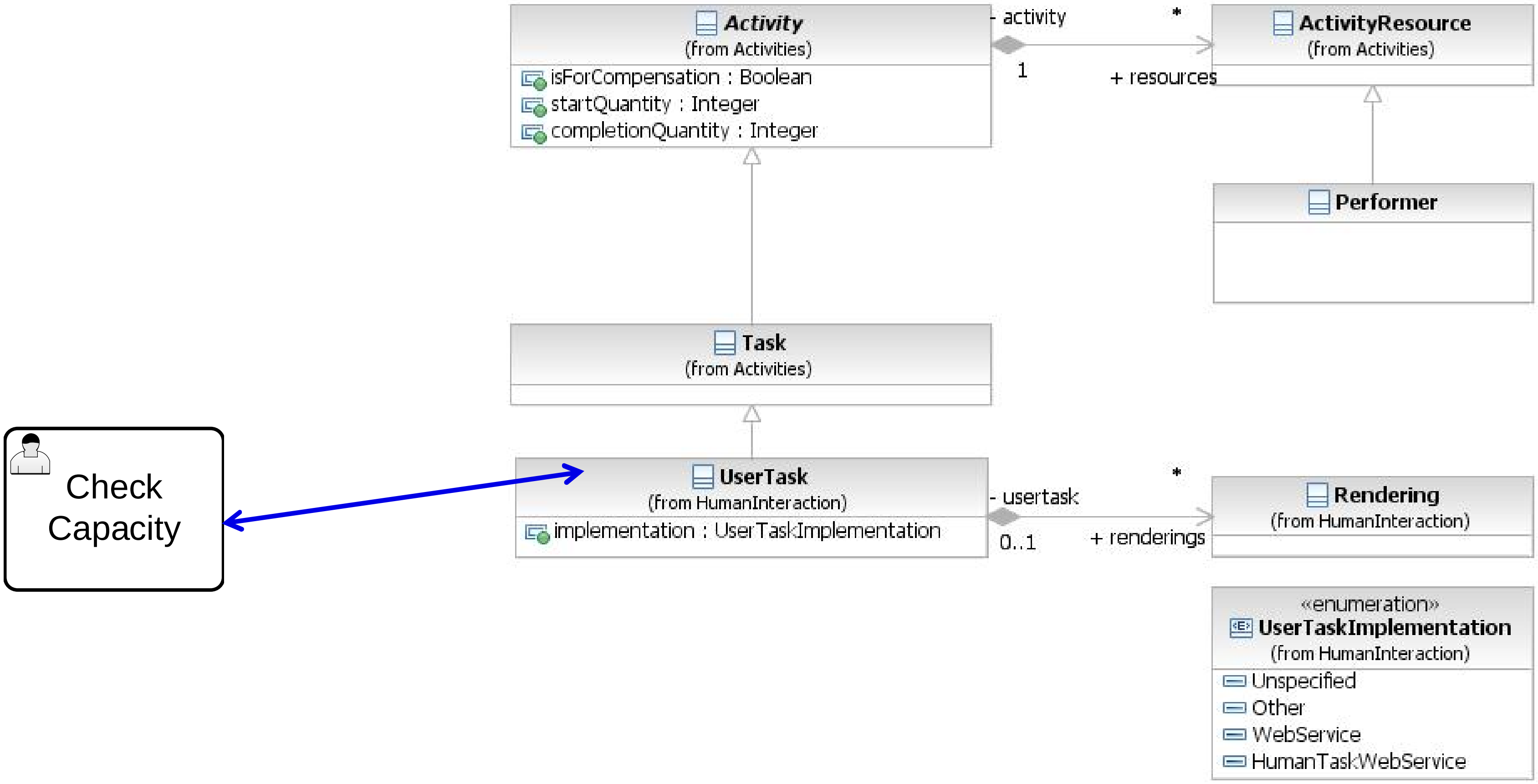
Data Flow Modeling (cont.)

Data Association Model



Human Interactions

- User Task: A typical “workflow” Task where a human performer performs the Task with the assistance of a software application



Check Capacity

Activity
(from Activities)
isForCompensation : Boolean
startQuantity : Integer
completionQuantity : Integer

- activity 1 * + resources

ActivityResource
(from Activities)

Performer

Task
(from Activities)

UserTask
(from HumanInteraction)
implementation : UserTaskImplementation

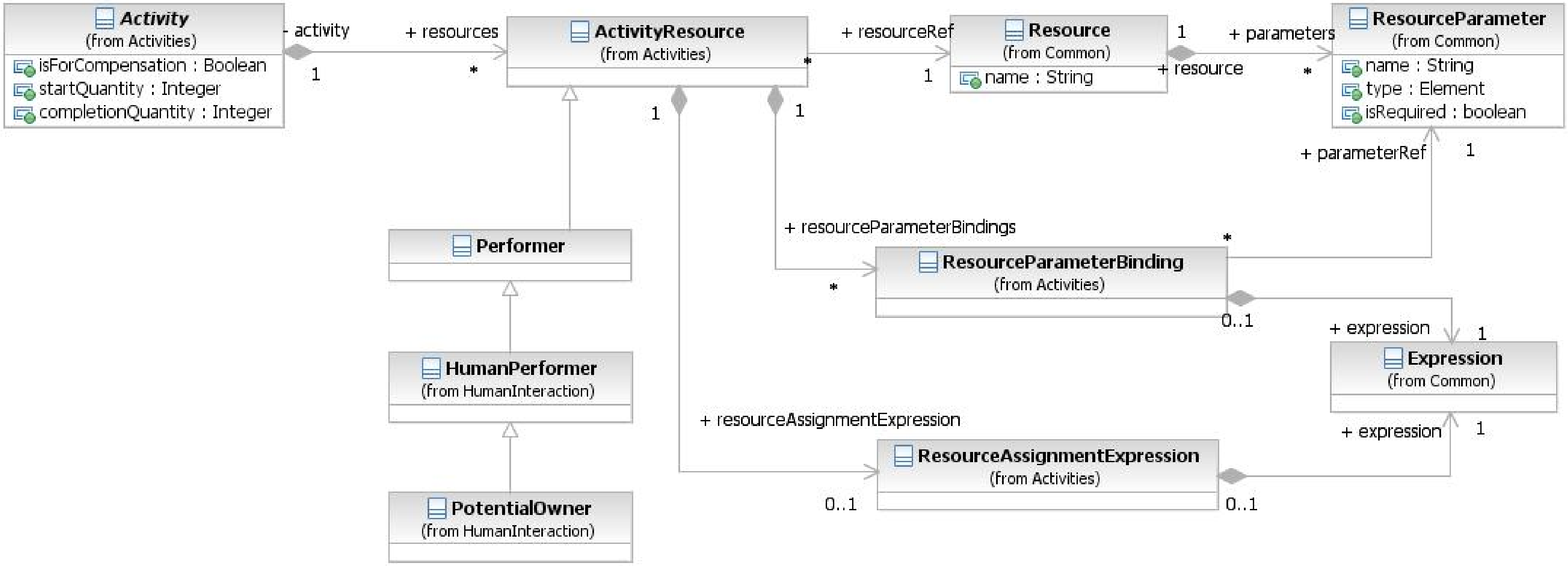
- usertask 0..1 * + renderings

Rendering
(from HumanInteraction)

«enumeration»
UserTaskImplementation
(from HumanInteraction)
Unspecified
Other
WebService
HumanTaskWebService

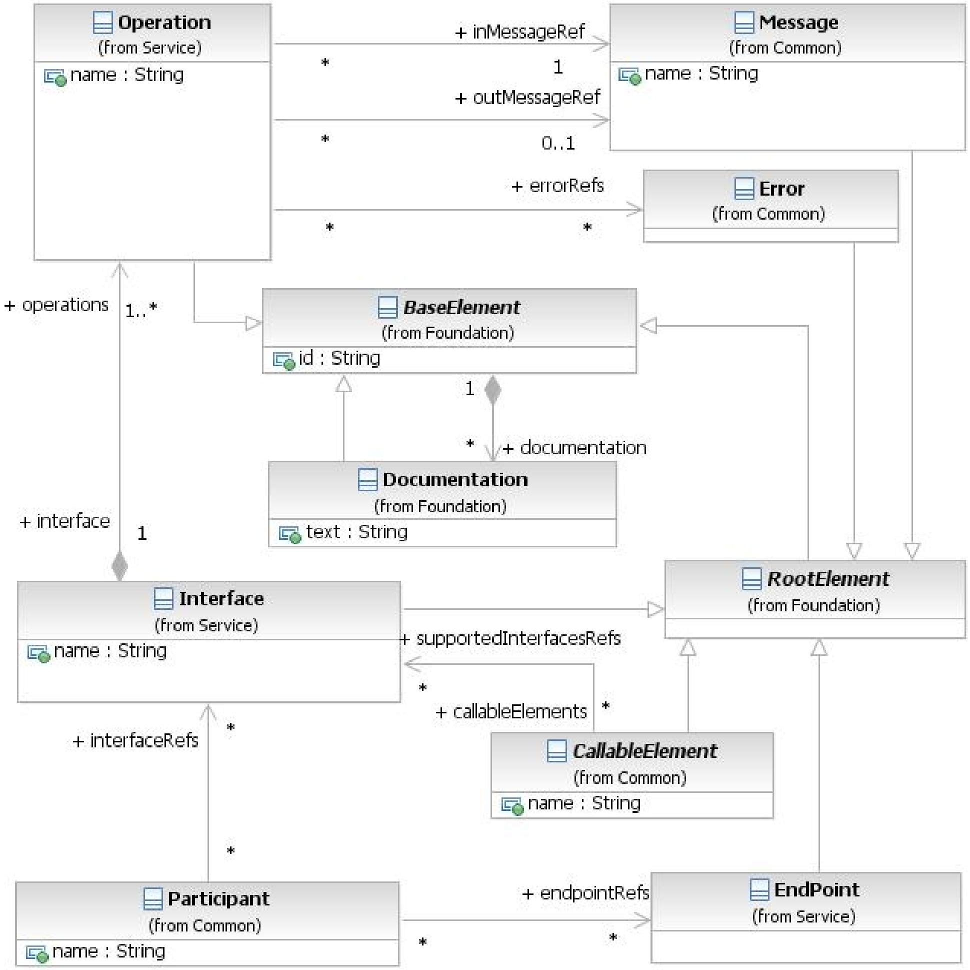
Human Interactions

- People Assignment

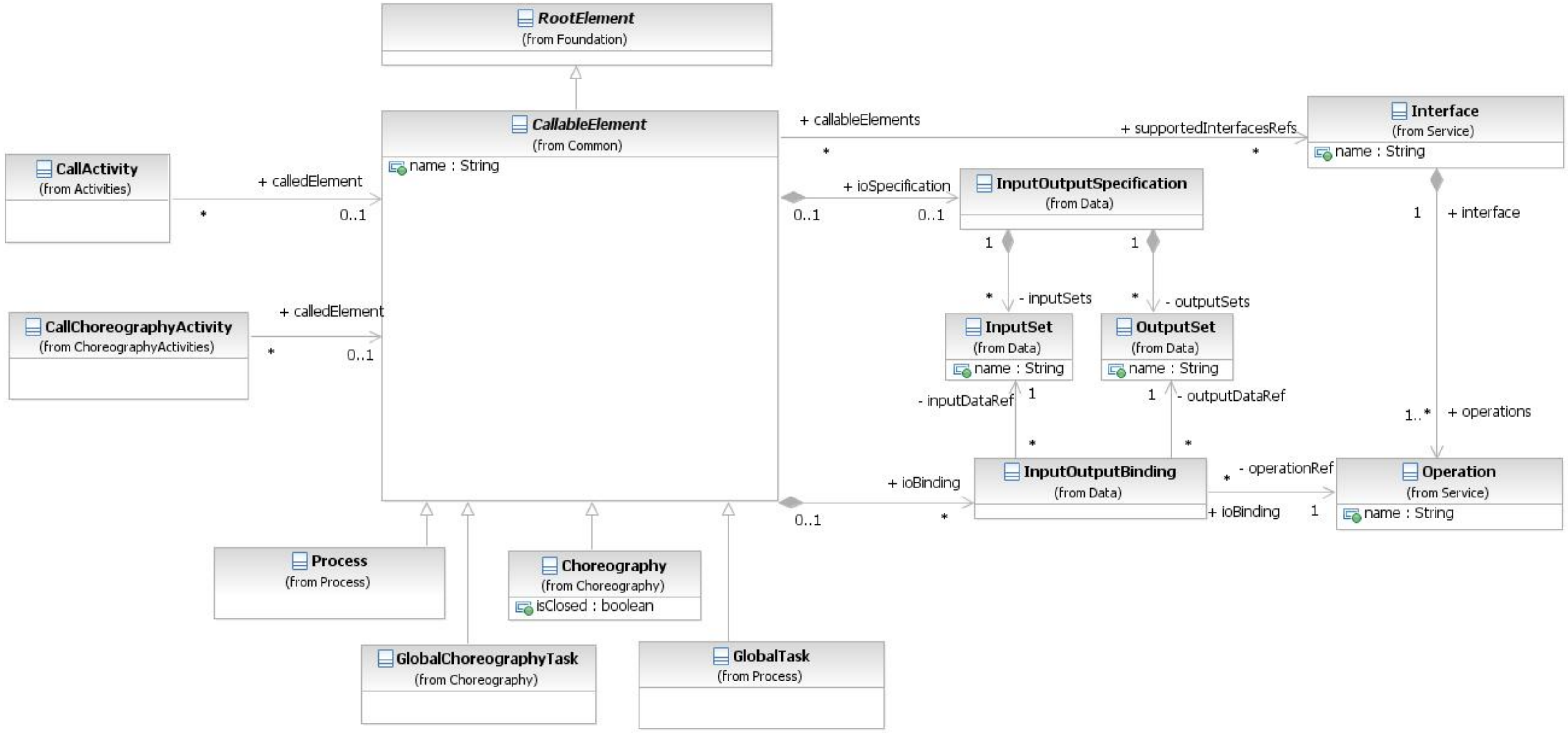


Service Model

- Introduces constructs necessary for modeling core SOA concepts



Service-enabled BPMN Elements



Agenda

- BPMN 2.0 Background
  - Facts
  - Scope
  - New Features
- BPMN 2.0 Structure
- BPMN Modeling
  - Processes
  - Interactions
- Diagram Interchange
- Future

Interaction Modeling

- Interaction Basics
  - Motivation
  - Participants, Messages, and Message Flow
- Message Flow
  - Sequencing and Grouping
- Multiple Participants
- Public and Private Processes
- Correlation between Processes and Message Flow
- Nesting Interactive Processes

Motivation: Business Services From Products

- Producers becoming service providers
  - Produce a cell phone ⇒ Provide mobile services
  - Fill an order ⇒ Provide maintenance services
- Benefits:
  - Increased market:
    - Provide multiple services on a single product
  - Differentiation
    - Add services competitors don’t have
  - Core business focus:
    - Partner for non-core services
- Business services require ongoing interaction with customers and partners.
  - Customers and partners agree on the expected interactions
  - What information or goods are needed when?
  - How are complaints and unusual situations handled?
  - Are followups scheduled?

Modeling Interactions

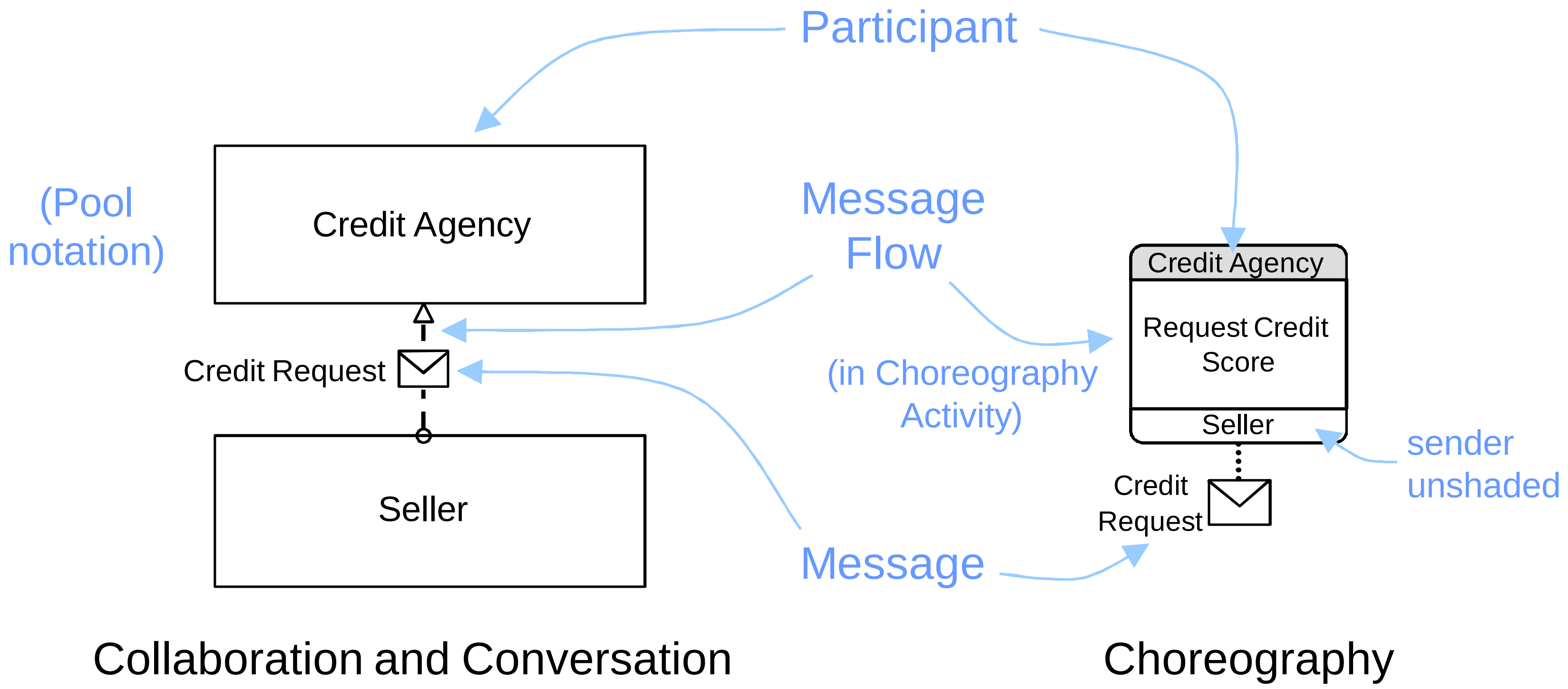
- Model interactions to:
  - Gather and share expertise
  - Coordinate and optimize interactions
- Benefits:
  - Customer retention, lower costs:
    - Reduce unnecessary or unsatisfactory interactions
  - Develop new business services
    - Combine and adapt existing services
- Three interaction diagrams:
  - Collaboration, Conversation, and Choreography
  - Each shows some aspects of interaction

Interaction Basics

- All interaction diagrams have Participants that send Messages to each other
- Participants
  - Can be “roles” in the interaction, such as Seller and Buyer
  - … or individuals, such as WalMart or US Government
  - Any scale: departments, businesses, or industries
- Messages
  - … are things sent between participants
  - Can be physical or informational
- Messages Flow between Participants
  - … at certain points during an interaction
  - The same message can be carried by more than one message flow

Interaction Basics

- Participants, messages, and message flows in the three interaction diagrams
- Collaboration and Conversation show Message Flow graphically, and Participants more prominently

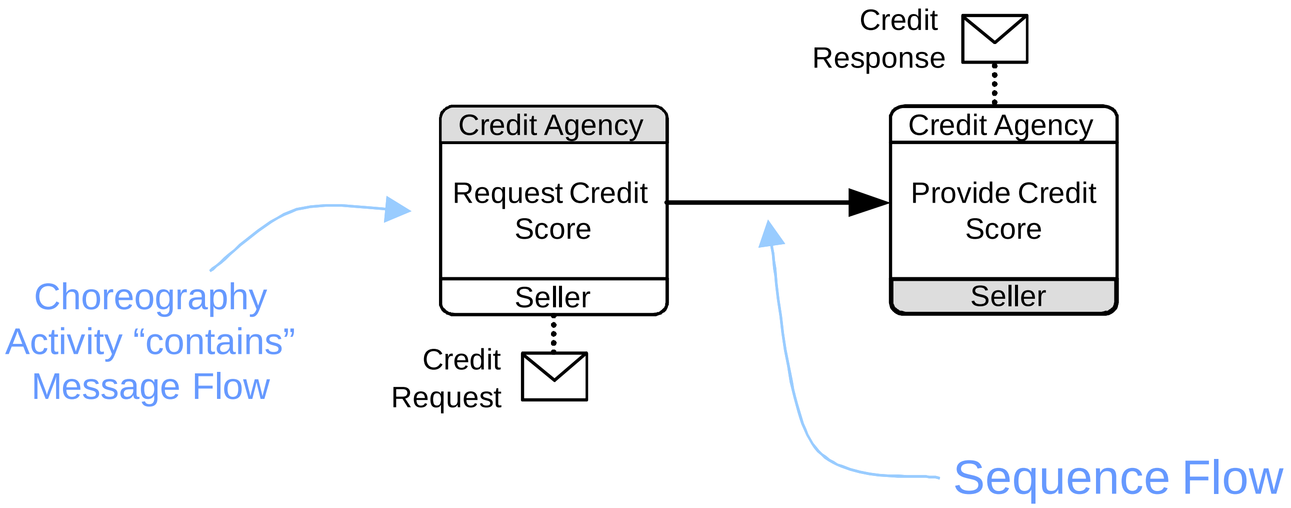


Interaction Basics

- Sequencing Message Flow
  - One message sent before another
  - In Choreography diagram
- Grouping Message Flow
  - On Choreography and Collaboration
- Processes in Participants
  - In Collaboration
- Correlation
  - Routing Messages to executing Processes
  - In Conversation

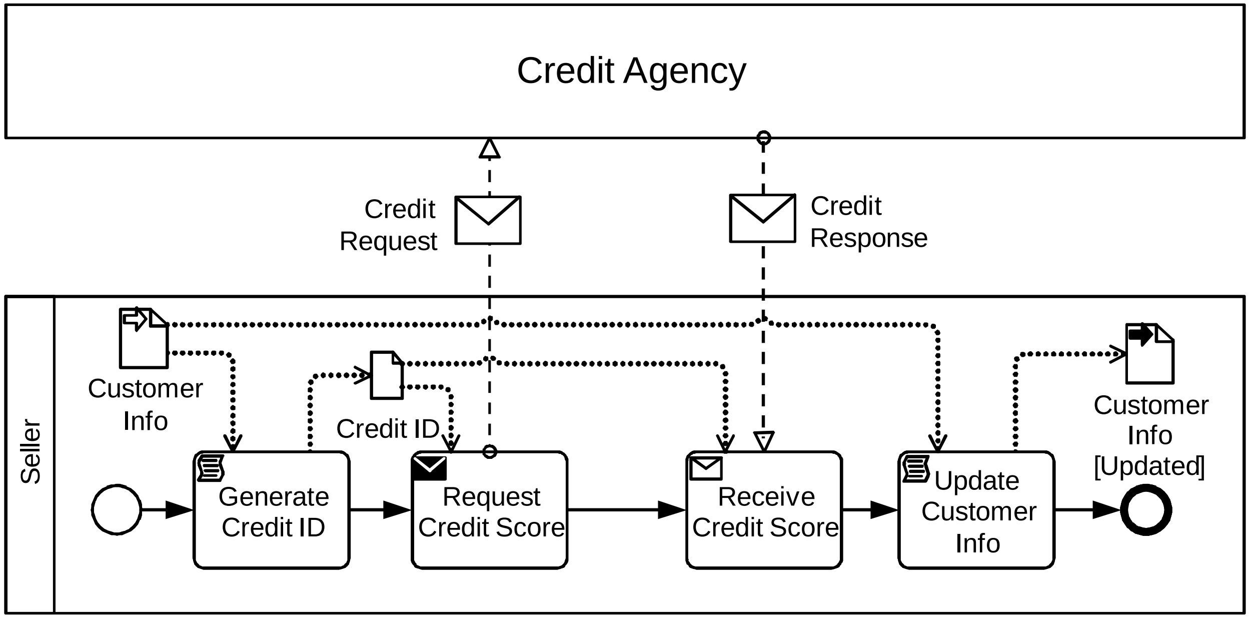
Message Flow Sequence

- Some messages flow before others during the interaction
- Choreography diagrams show this directly as sequence flow between Choreography Activities



Participants with Processes

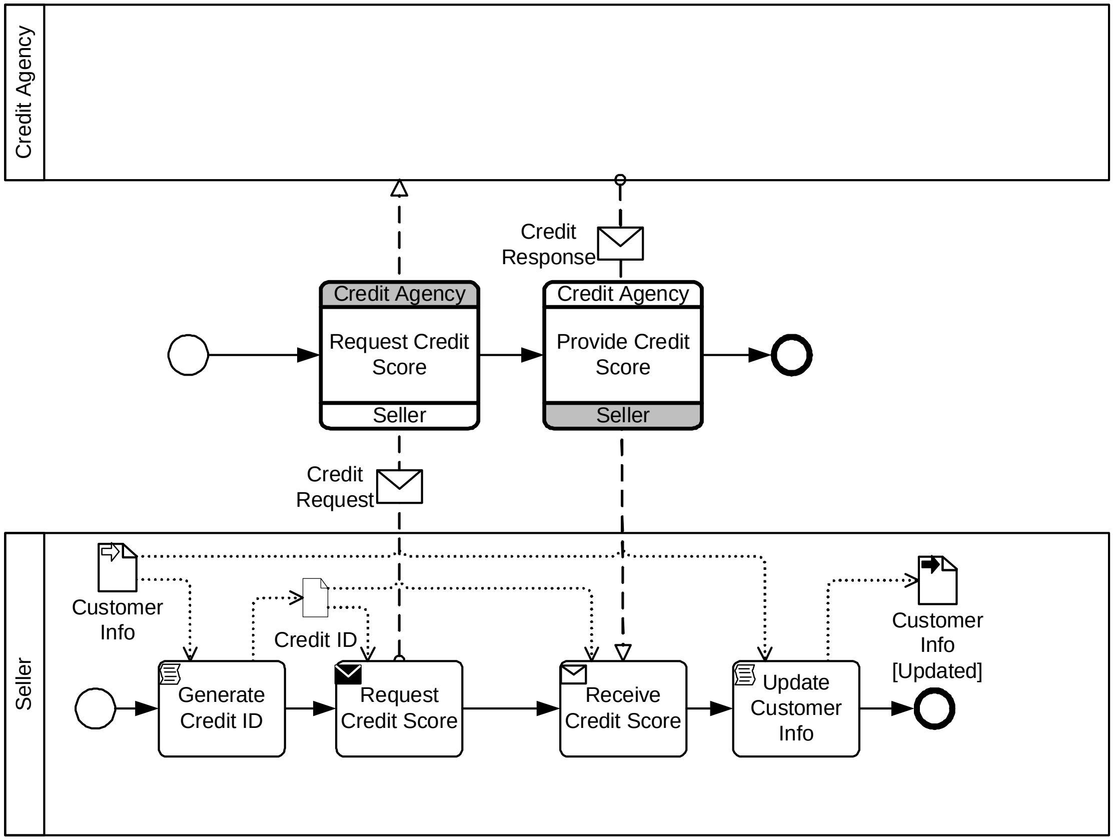
- Processes can be “inside” participants on Collaboration diagrams
  - (Not in Conversations)
- Message flow can connect to activities
  - indirectly specifying message flow sequence



(“Black-Box Pool”)

Choreography on Collaboration

- Choreography diagrams can be overlaid on Collaborations
  - (Not in Conversations)
  - Collaboration Message Flow can be “wired-up” to the appropriate Choreography Activities to show relationship between Processes and pre-defined Choreographies



Message Flow Sequencing Rules

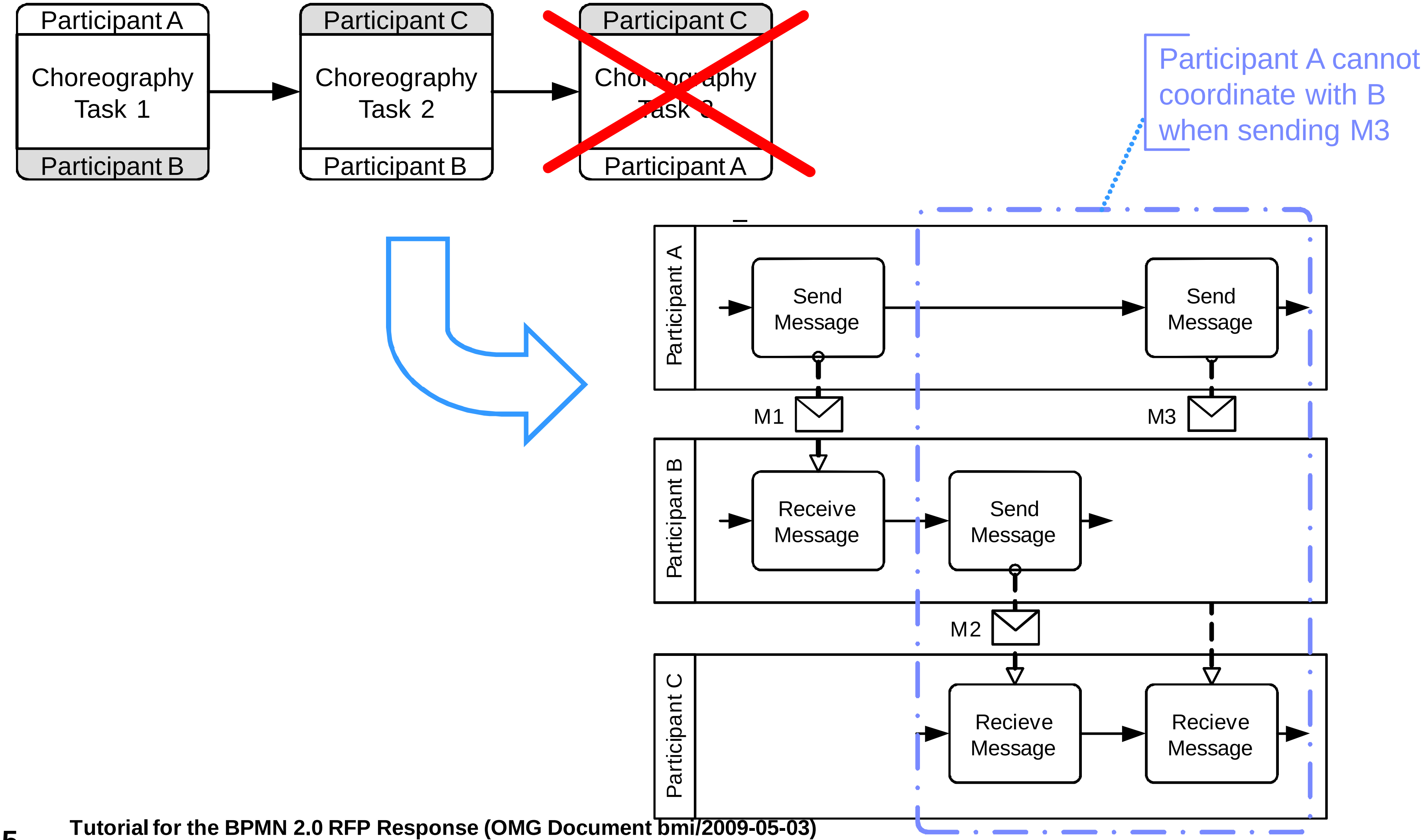
- No central controller for interactions as there is for process

- Participants cannot “see” into each other
  - Each participant only knows the progress of the Interaction by the Messages it sends and receives itself

- Data does not persist in the choreography
  - Data shared between Participants must have been sent or received in a Message

Message Flow Sequencing Rules

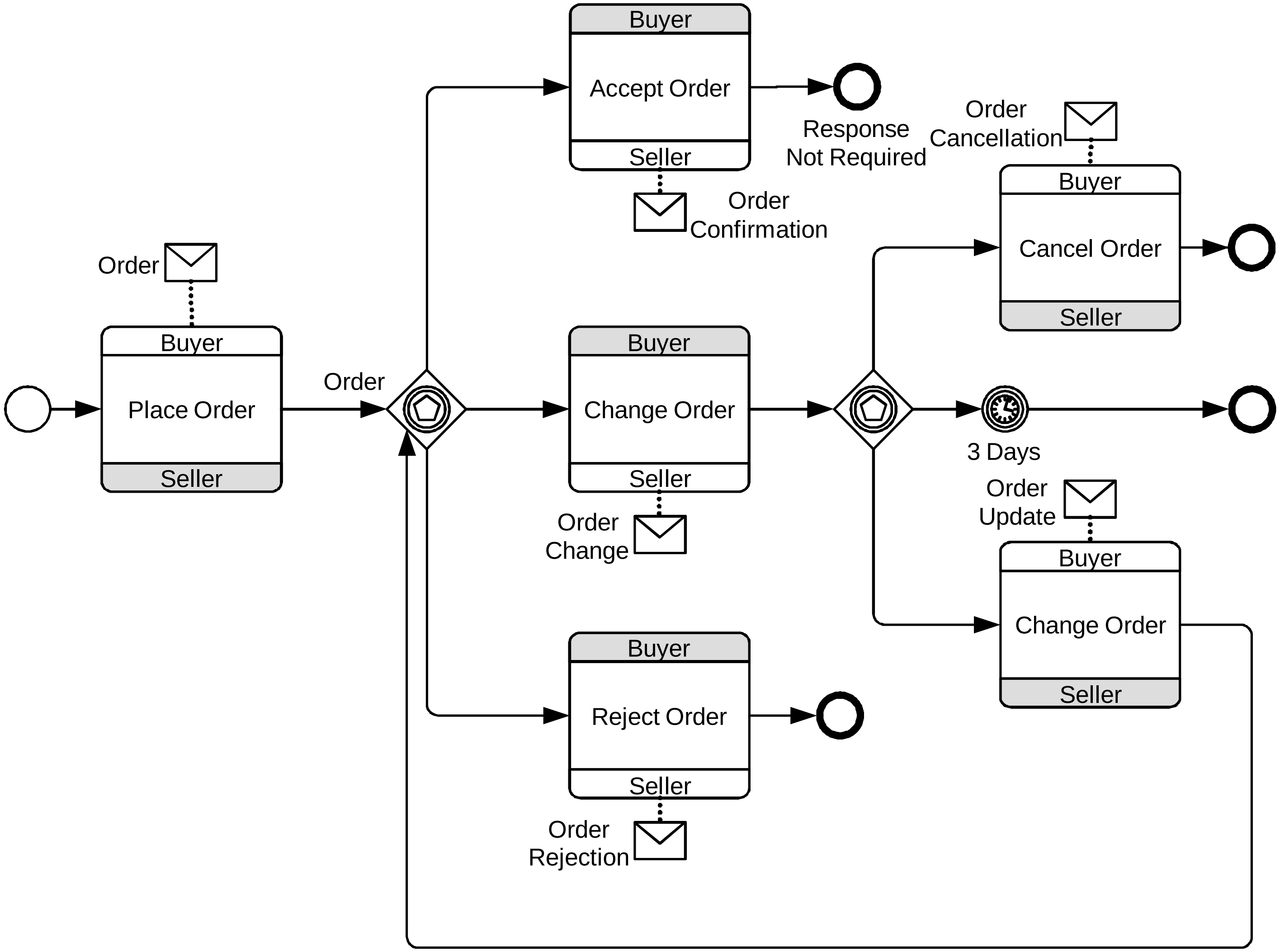
- Example of invalid sequencing



Participant A cannot coordinate with B when sending M3

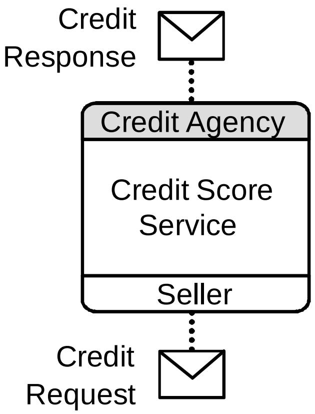
Message Flow Gateways

- Choreography diagrams support gateways and events

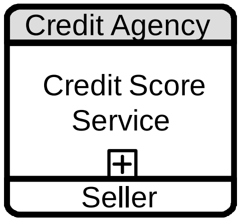


Grouping Message Flow (Choreography)

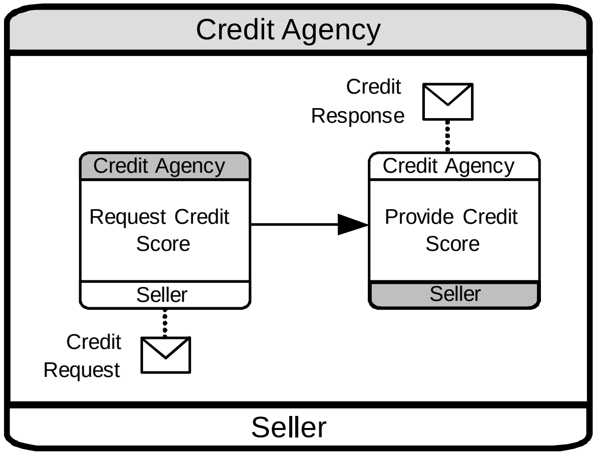
- Nested interactions
  - Similar to subprocesses and calling processes
- Choreography and Conversation diagrams show this



Choreography Task
with Multiple Message Flow
(no message sequence
beyond the initiator)



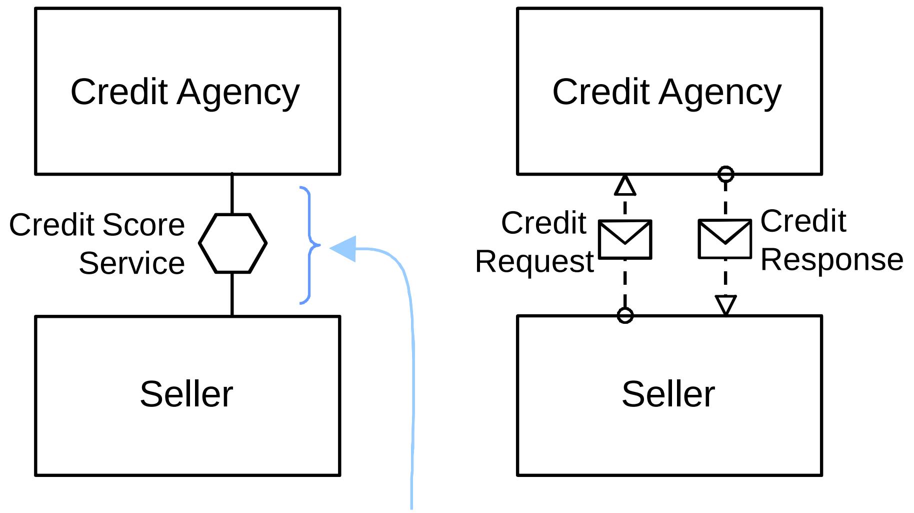
(Choreography Subprocess has thin border. Calling Global Choreography Tasks omits "plus" icon)



Choreography Subprocess
or Call Choreography Activity
(nested choreography has
message sequence)

Grouping Message Flow (Conversation)

- Communications and nested Conversations



Credit Agency

Credit Score Service

Seller

Credit Agency

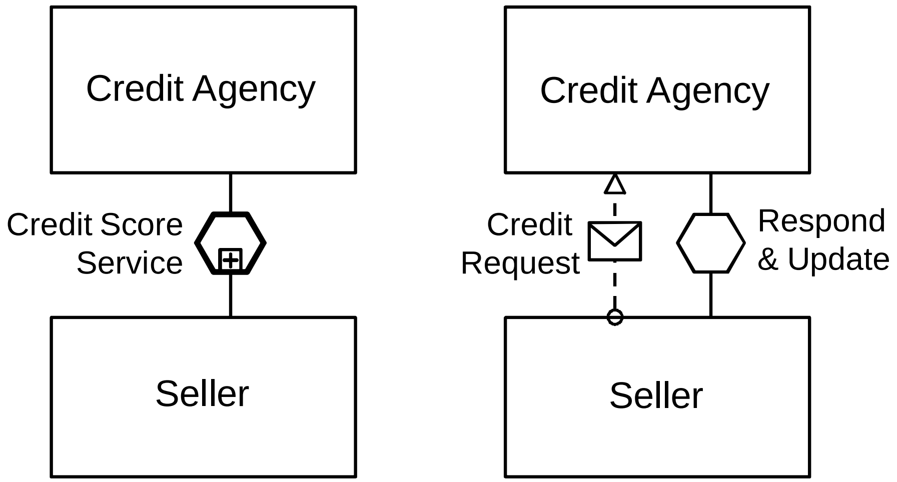
Credit Request

Credit Response

Seller

Communication

Communication with Message Flow
(no nested communications or conversations)



Credit Agency

Credit Score Service

Seller

Credit Agency

Credit Request

Respond & Update

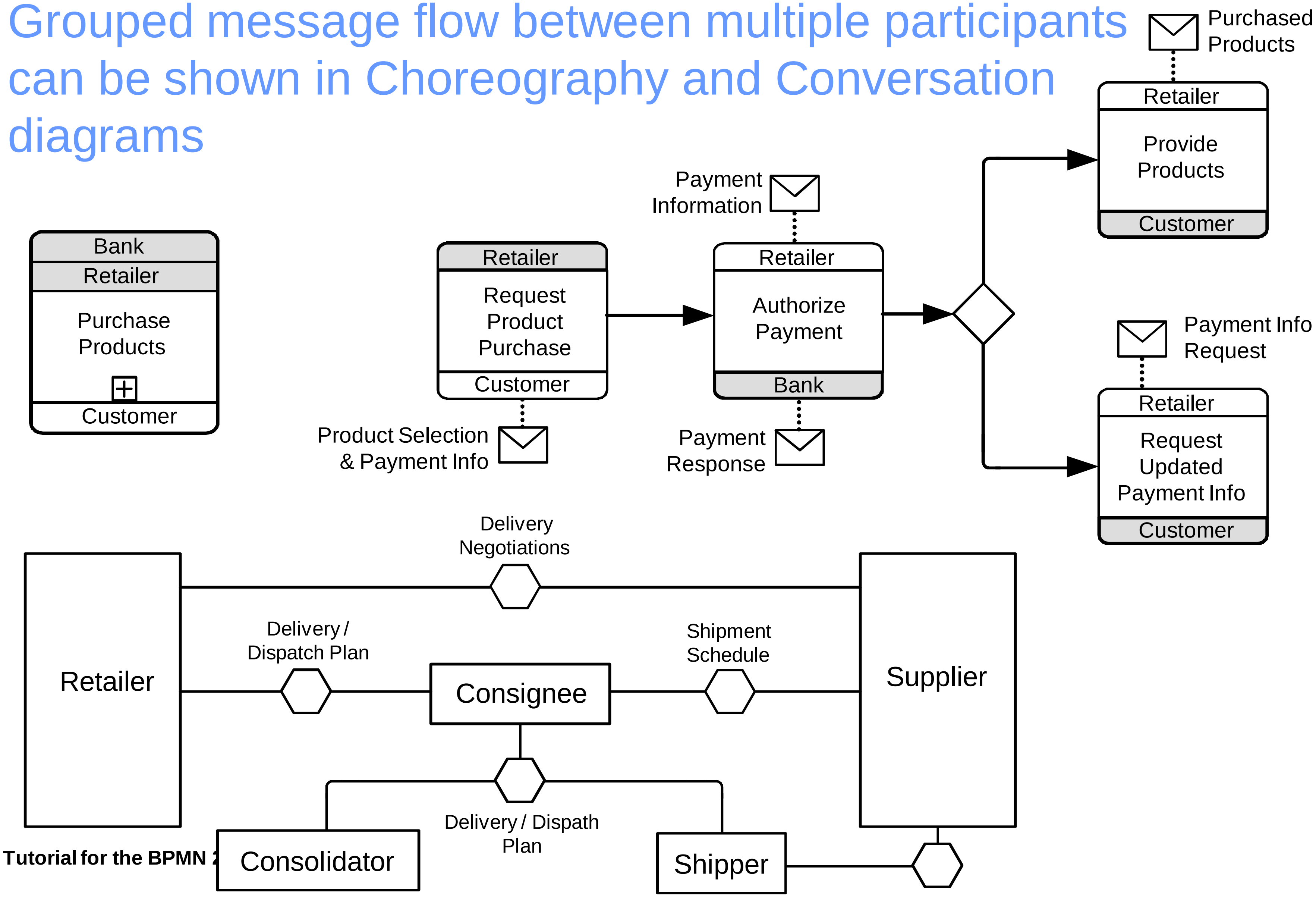
Seller

(Sub-Conversation has thin border.
Calling Global Communications omits
the ”plus” icon)

CallConversation or Subconversation
(nested communications
or conversations)

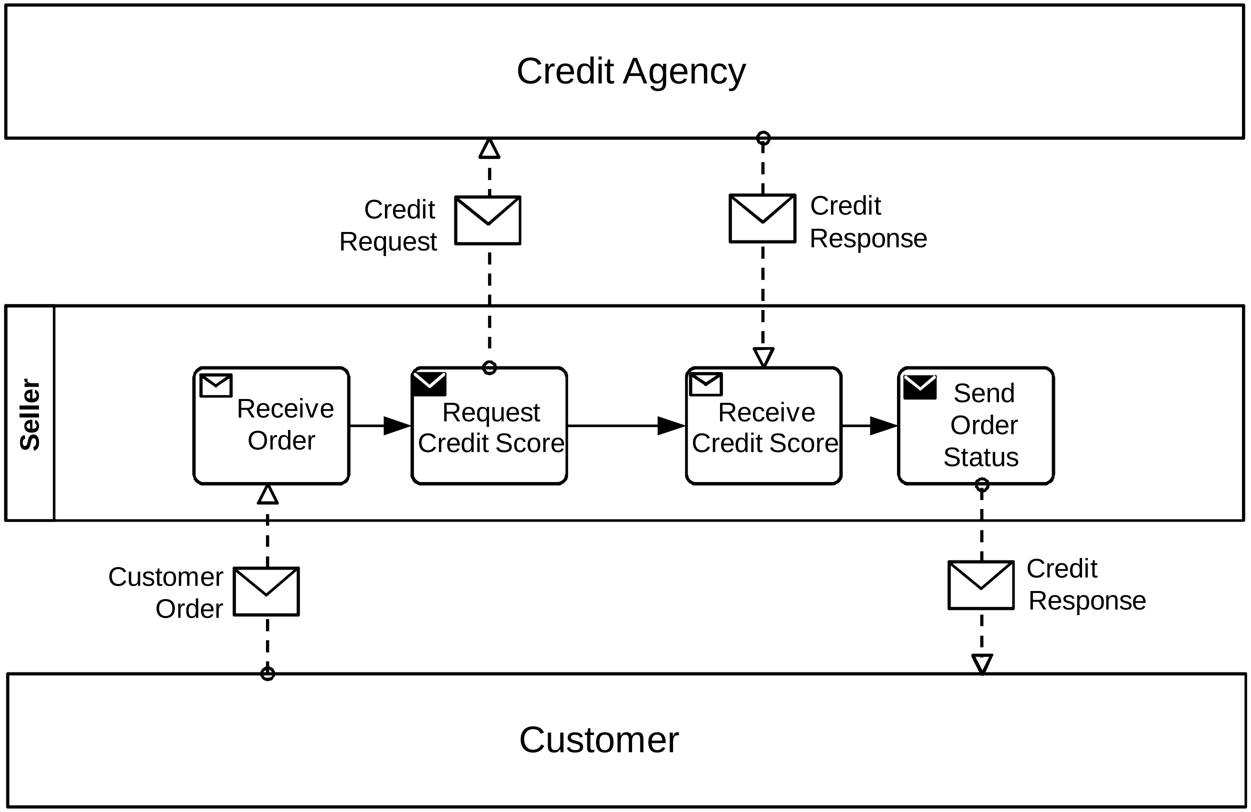
More Than Two Participants

- Grouped message flow between multiple participants can be shown in Choreography and Conversation diagrams



Multiple Participants (Process)

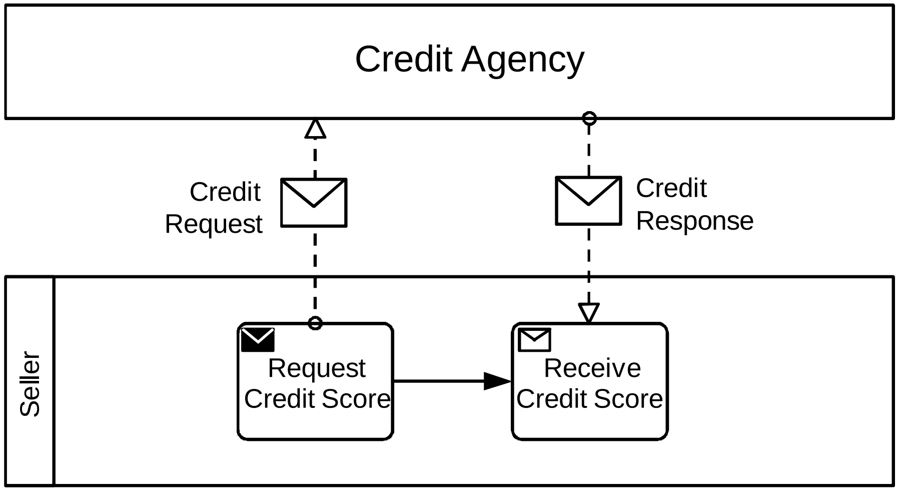
- Interactive processes might send and receive from multiple participants
  - Which participants, and when?
  - Include Collaboration in definition of a Process



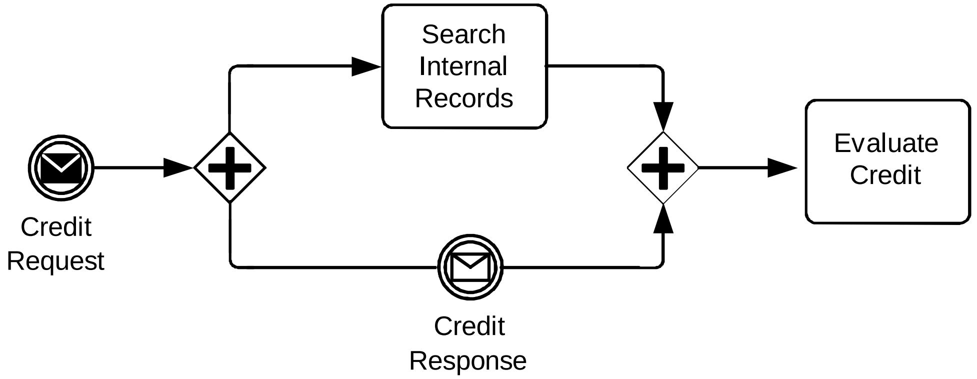
Collaboration is part of Process definition

Public and Private Processes

- Participants might only show public view of a process



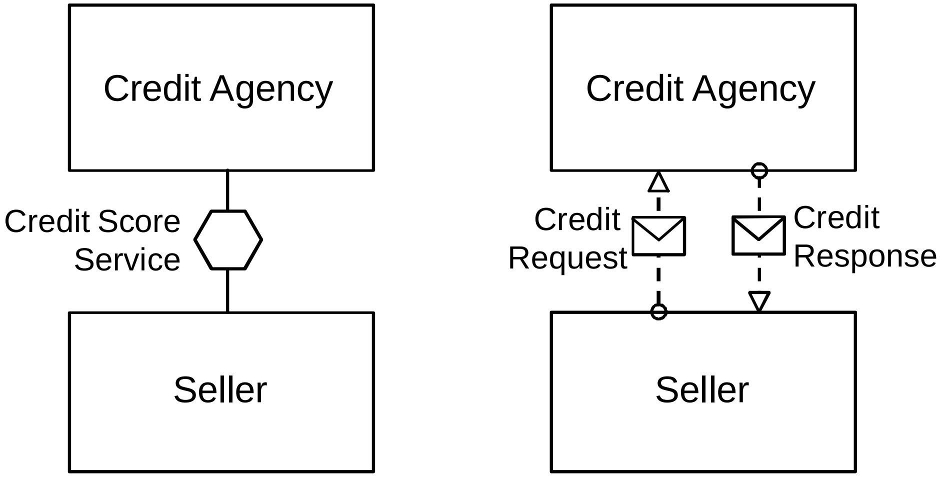
- Private process can model the same interaction differently



Processes and Message Flow

- Messages coming into a participant might be routed to existing processes already underway
  - Routing based on data in the message and/or process
  - Conversations capture part of this (nongraphically)

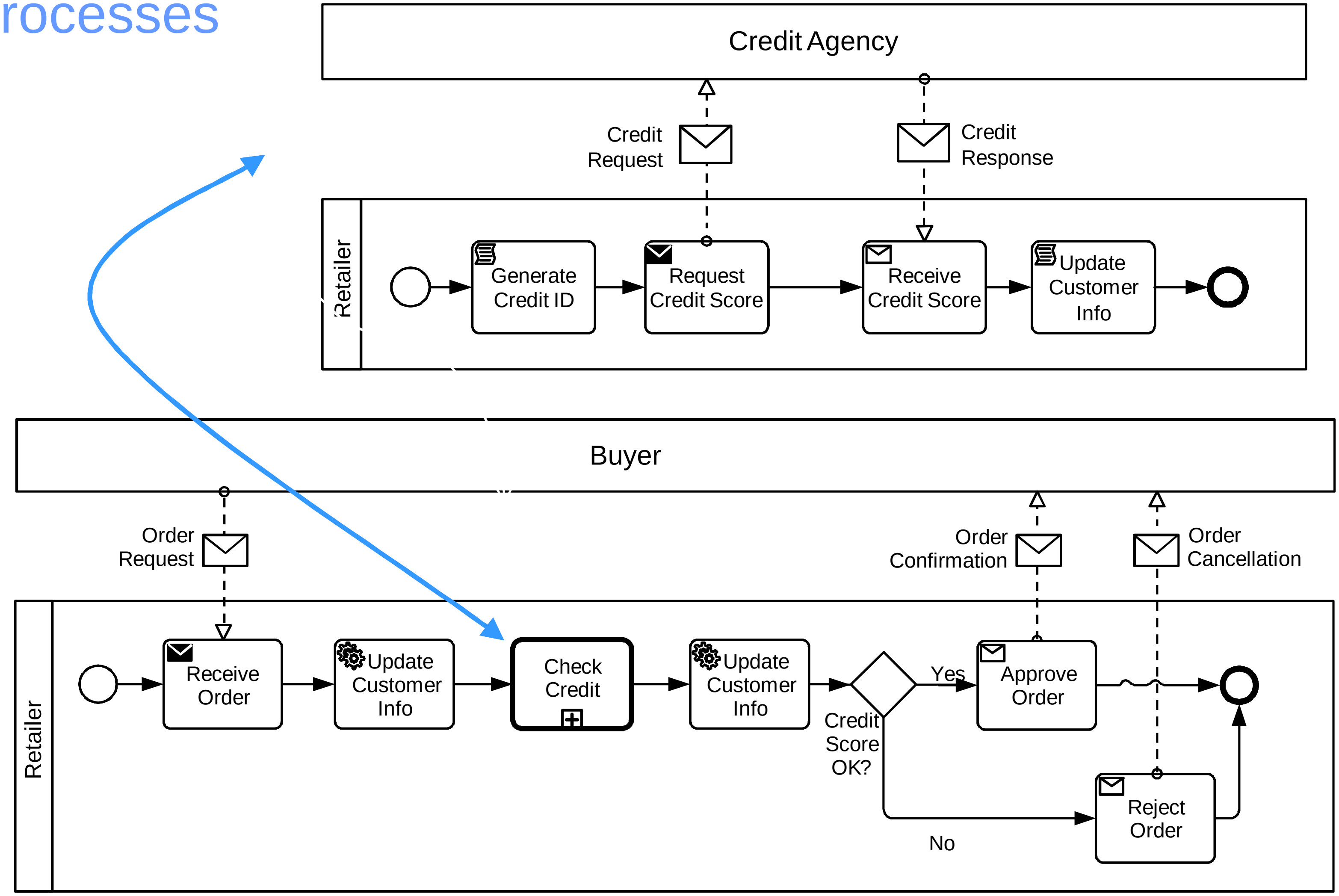
Communication specifies that personal identification number is used for correlation



Message sent during the same communication at runtime contain the same personal identification number

Nesting Interactive Processes

- Interactive Processes can be nested in other interactive processes

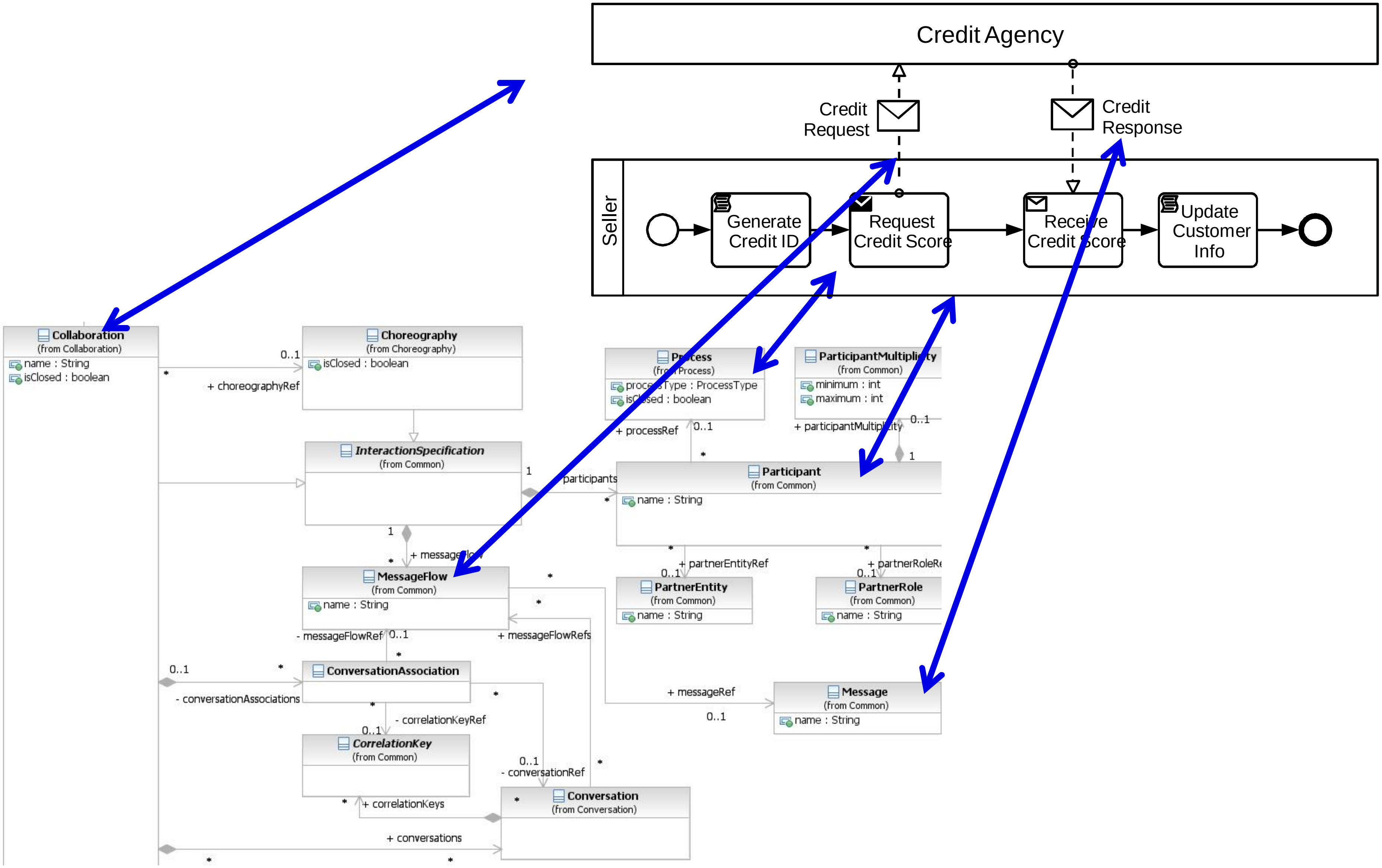


Interaction Diagram Capabilities

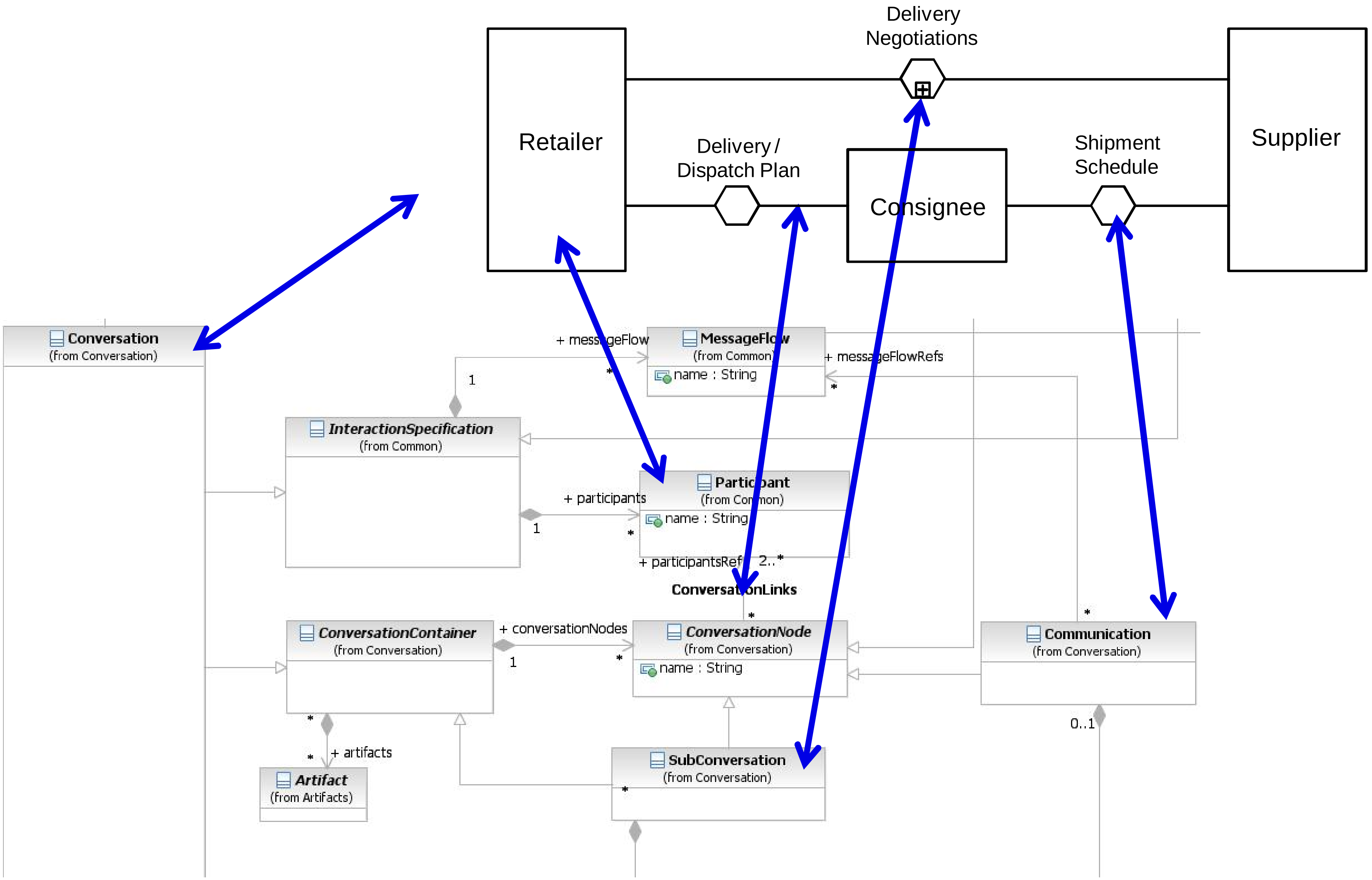
- Three diagrams for interactions
  - Collaboration, Conversation, and Choreography
  - All capture Participants, Messages, Message Flow
  - Choreography shows sequencing of message flows directly
  - Collaboration shows processes in participants
  - Choreography and Conversation show grouped message flow
  - Conversations capture Correlation

	Choreography	Collaboration	Conversation
Participants, Messages, Message Flow	X (Message Flow not shown graphically)	X	X
Sequencing Message Flow	X	(through processes in participants)	
Grouping Message Flow	X		X
Processes in Participants		X	
Correlation	(through attached conversation)	(through attached conversation)	X

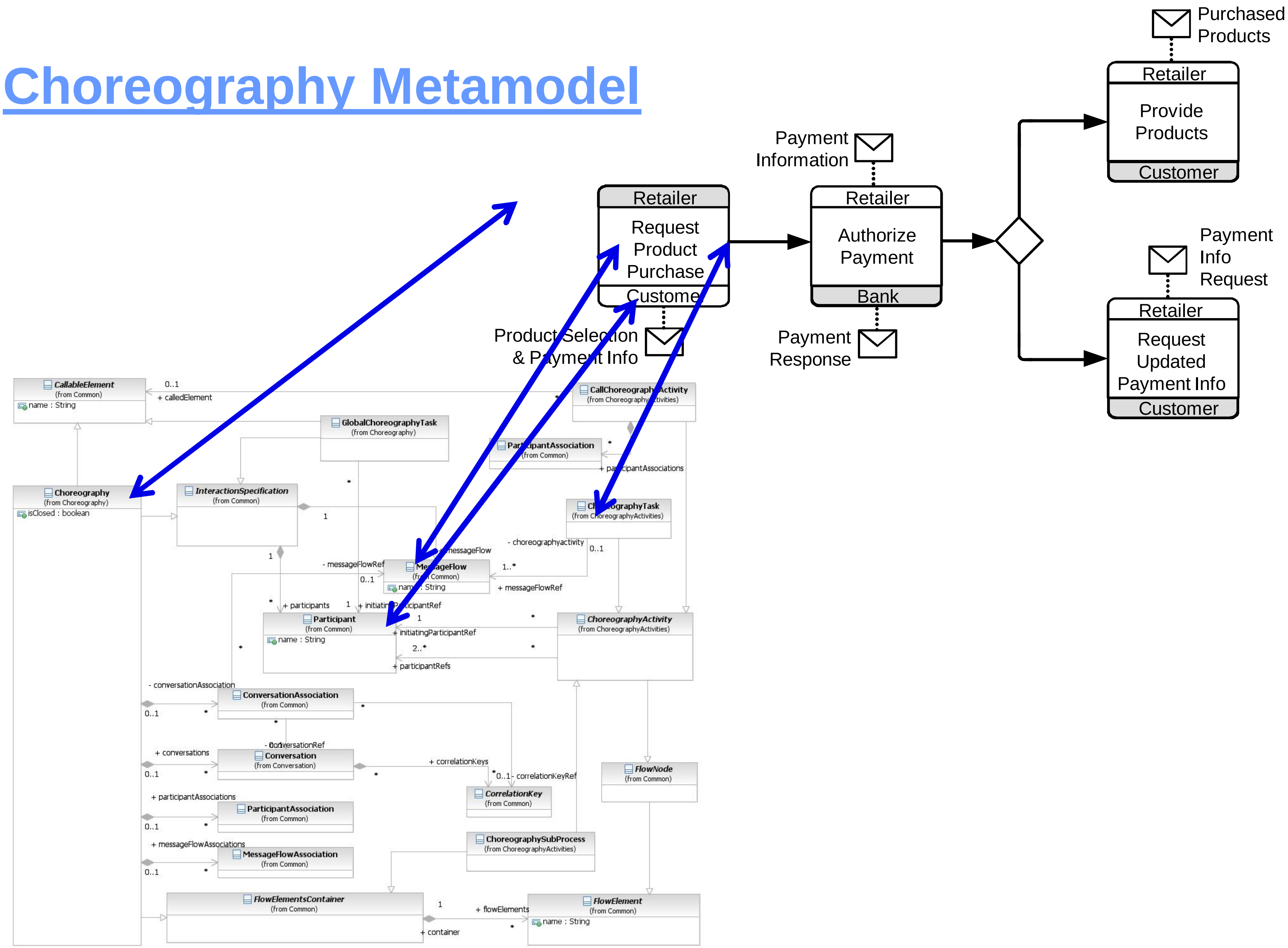
Collaboration Metamodel



Conversation Metamodel


Delivery Negotiations
Retailer
Delivery / Dispatch Plan
Consignee
Shipment Schedule
Supplier
Conversation (from Conversation)
+ messageFlow
1
*
MessageFlow (from Common)
name : String
+ messageFlowRefs
*
InteractionSpecification (from Common)
+ participants
1
*
Participant (from Common)
name : String
+ participantsRef
2..*
ConversationLinks
*
ConversationContainer (from Conversation)
+ conversationNodes
1
*
ConversationNode (from Conversation)
name : String
*
Communication (from Conversation)
0..1
*
+ artifacts
*
Artifact (from Artifacts)
SubConversation (from Conversation)
*

Choreography Metamodel



Agenda

- BPMN 2.0 Background
  - Facts
  - Scope
  - New Features
- BPMN 2.0 Structure
- BPMN Modeling
  - Process
  - Interactions
- Diagram Interchange
- Questions

Diagram Interchange Format

- An interchange format for BPMN diagrams provided
  - Allows separation of semantic data and diagram data
  - Adds flexibility, e.g. a BPMN element can have multiple diagram elements
- Architectural approach:
  - Diagram Interchange Metamodel
    - A core that captures the main design pattern
    - An existing work covered by a separate OMG Diagram Definition RPF reused
    - Still work in progress
  - BPMN Diagram Definition Library
    - An instance the core model covering the BPMN semantic model
    - Examples: BPMN Diagram Definitions, BPMN Lane Definition, BPMN Pool Definition, etc.

Diagram Interchange Format

- The Diagram Interchange Metamodel: The Core

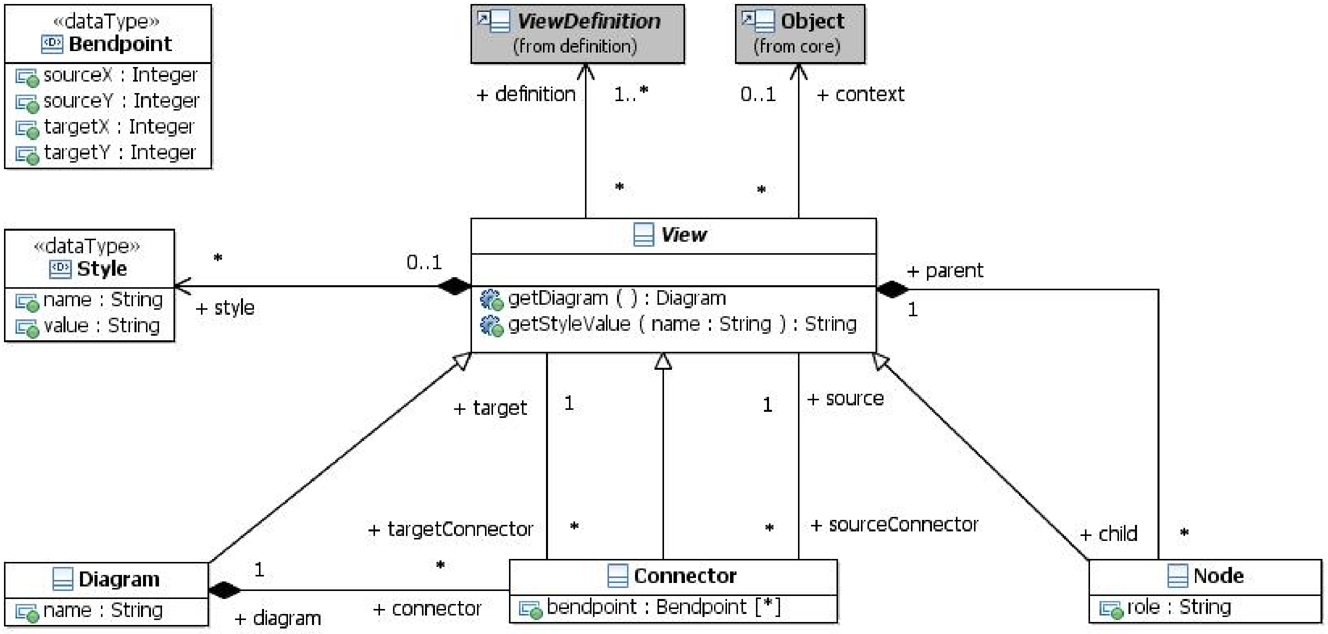


Diagram Interchange Format

- BPMN Diagram Definitions

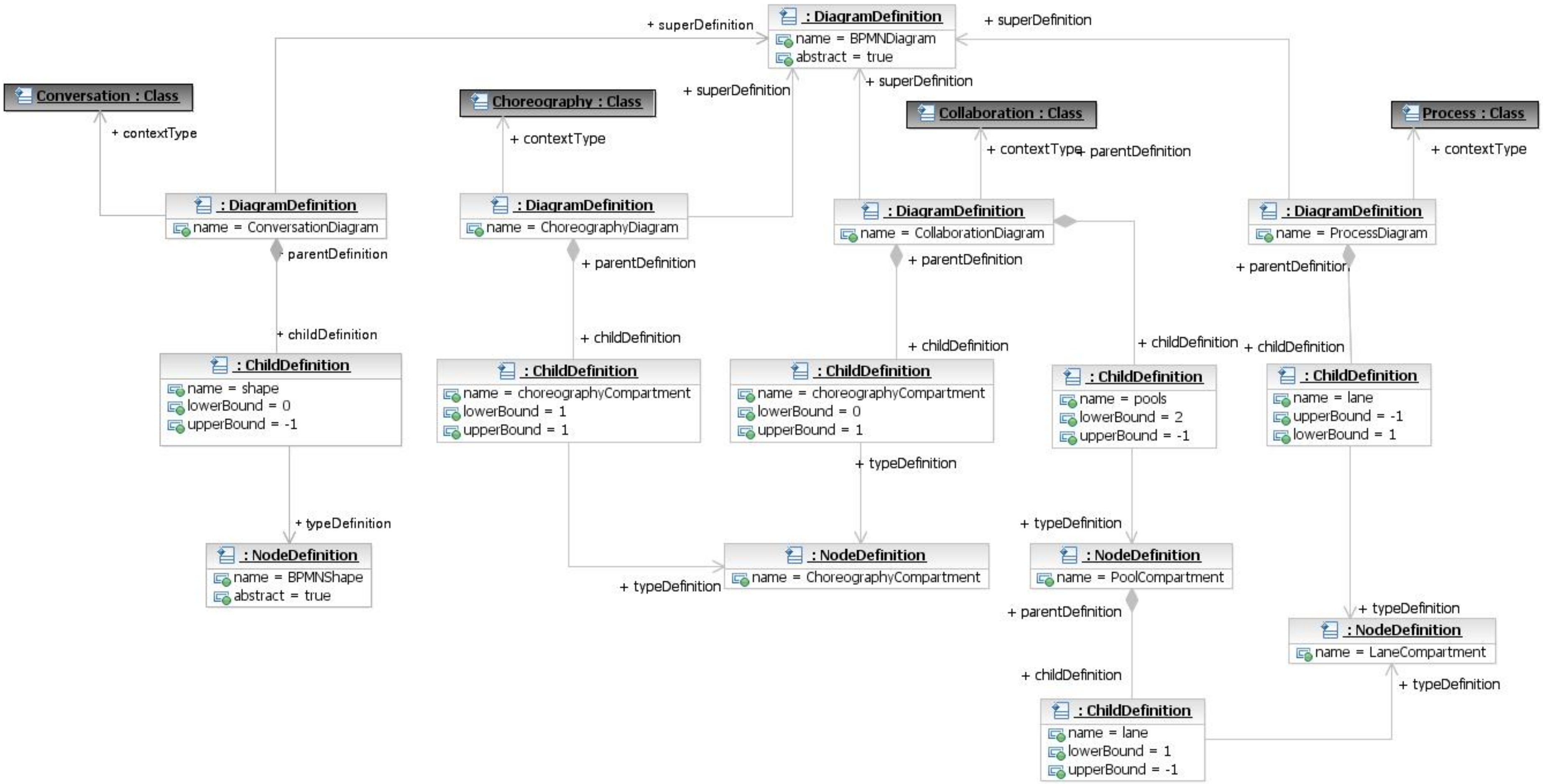
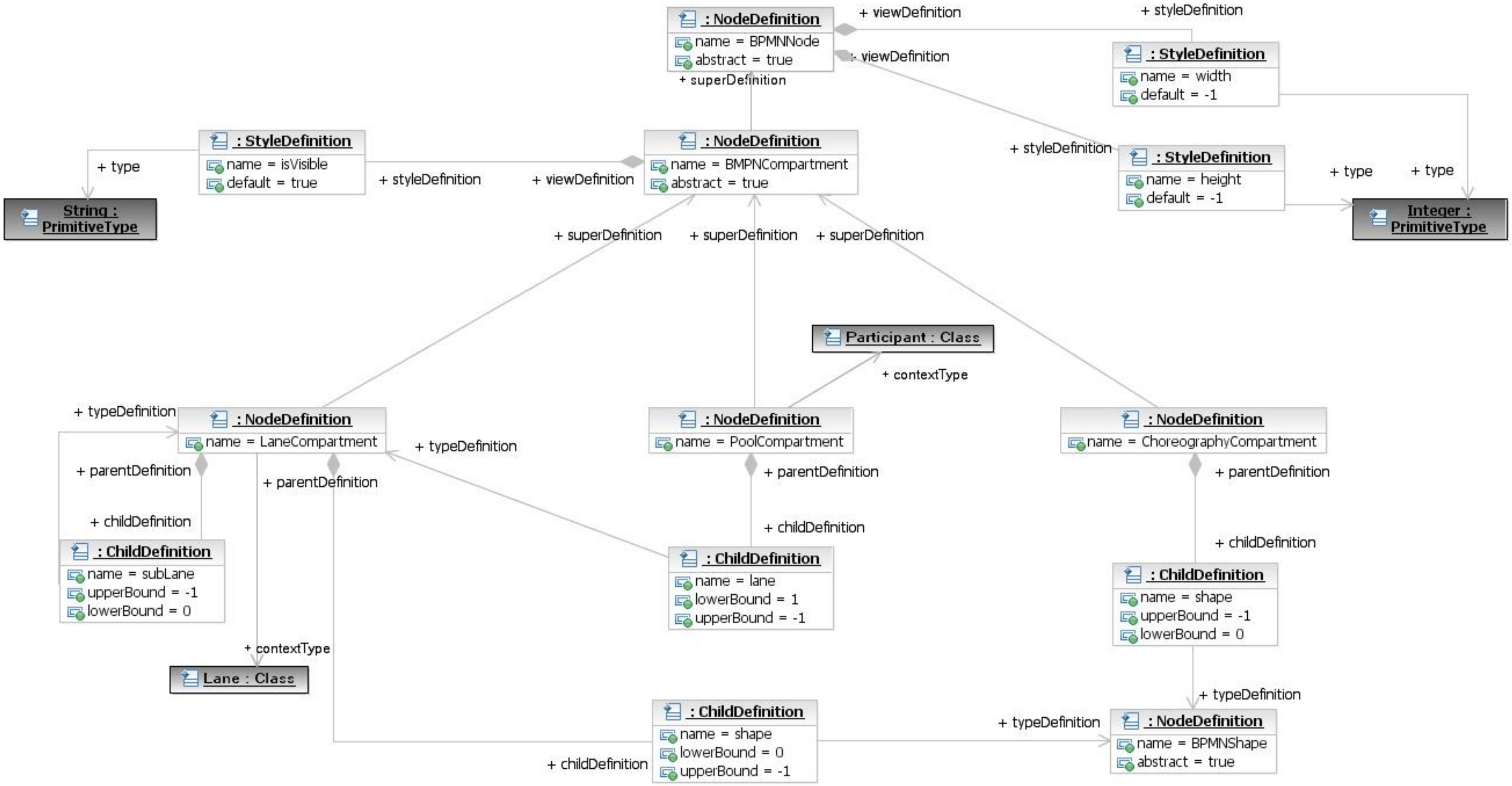


Diagram Interchange Format

- BPMN Compartment Definitions



Thank You

Questions?