

































## BPMN and eXtended BPMN Shapes

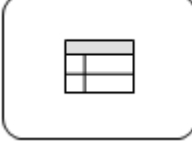


In the following we will illustrate the BPMN and eXtended BPMN Shapes. The element and notational deficiencies of BPMN, limited the utility of practical process mapping and modelling for much of how business process notations should or could be used, e.g. business process modelling, process transformation, software blueprint, etc. As a matter of fact, we believe that the BPMN 2.0 standard actually makes siloed thinking worse in that it tends to cause process experts and analysts to only consider processes (and therefore the process models) within a very narrow perspective and to not consider other aspects of work and the business which relate to the process. To fill this gap, the Global University Alliance, together with the LEADing Practice community, developed the eXtended BPMN standard to enhance the expressiveness of the existing BPMN 2.0 specification. While the eXtended BPMN icons build upon the existing BPMN 2.0 specification, they include linkage to aspects such as:

1. Business Competencies (added in January 2011).
2. Service Mapping (added in January 2011).
3. Interaction models (added in March 2011).
4. RACI Mapping (added in May 2011).
5. Value Mapping and thereby links to strategy, critical success factors, and objectives (added in November 2011).
6. Advanced Rules modelling (added in December 2011).
7. Decision making aspects (added in December 2011).
8. SAP Netweaver® connection (added in January 2012)
9. ERP connection, in terms of which objects can be reused for ERP implementation, e.g. SAP Solution Manager® (added in February 2012)
10. Measurements (KPIs, PPIs & SPIs) and link to reporting (added in May 2012)

Below are the shape types of BPMN and eXtended BPMN

BPMN 2.0	eXtended BPMN	Description (and description source)
		<p><b>Icon</b>      <b>Company (and product) or Organization</b></p> <p> LEADing Practice, eXtended BPMN Specification</p> <p> Object Management Group, BPMN 2.0 Specification</p> <p> SAP Netweaver Rules Composer</p> <p> Oracle Business Rules</p> <p> IBM® WebSphere Decision Center V8.0</p> <p> Microsoft® .NET Framework</p>
 None	 None	 No special Task Type is indicated.
 User Task	 User Task	 A User Task is a typical “workflow” Task where a human performer performs the Task with the assistance of a software application and is scheduled through a task list manager of some sort.
 Manual Task	 Manual Task	 A Manual Task is a Task that is expected to be performed without the aid of any business process execution engine or application.
	 Automated Task	 An automated task is an activity that is automated either through a business process execution engine or any application, e.g. using application features and functions to perform the application task.

BPMN 2.0	eXtended BPMN	Description (and description source)
 Service Task		 A Service Task is a Task that uses some sort of service, which could be a Web service or an automated application.
	 Manual Service	 A manual service is when the service is performed by a human.
	 Automated Service	 An automated service could be either an application service, data service, platform service, infrastructure service, or a Web Service.
 Receive Task	 Receive Task	 A Receive Task is a simple Task that is designed to wait for a Message to arrive from an external Participant (relative to the Process).
 Send	 Send	 A Send Task is a simple Task that is designed to send a Message to an external Participant (relative to the Process).
 Script	 Script	 A Script Task is executed by a business process engine. The modeller or implementer defines a script in a language that the engine can interpret. When the Task is ready to start, the engine will execute the script. When the script is completed, the Task will also be completed.

BPMN 2.0	eXtended BPMN	Description (and description source)
 <p data-bbox="165 524 335 555">Business Rule</p>	 <p data-bbox="405 524 574 555">Business Rule</p>	 A Business Rule Task provides a mechanism for the Process to provide input to a Business Rules Engine and to get the output of calculations that the Business Rules Engine might provide. The input/output specification of the Task will allow the Process to send data to and receive data from the Business Rules Engine.

The following table shows the additional shapes incorporated into eXtended BPMN, at the same time showing how the concepts which these shapes represent are included in the logic, language, and practice of modern business, while not being addressed or supported within BPMN 2.0.



**NOTE:** As specified by the OMG BPMN 2.02 standard is there a flexibility in the size, color, line style, and text positions of the defined graphical elements. The following extensions to a **BPMN** Diagram are permitted<sup>1</sup>:

- New markers or indicators **MAY** be added to the specified graphical elements. These markers or indicators could be used to highlight a specific attribute of a **BPMN** element or to represent a new subtype of the corresponding concept.
- A new shape representing a kind of **Artifact** **MAY** be added to a Diagram, but the new **Artifact** shape **SHALL NOT** conflict with the shape specified for any other **BPMN** element or marker.
- Graphical elements **MAY** be colored, and the coloring **MAY** have specified semantics that extend the information conveyed by the element as specified in this International Standard.
- The line style of a graphical element **MAY** be changed, but that change **SHALL NOT** conflict with any other line style **REQUIRED** by this International Standard.
- An extension **SHALL NOT** change the specified shape of a defined graphical element or marker (e.g., changing a square into a triangle, or changing rounded corners into squared corners, etc.).









For each shape the use of the object within the eXtended BPMN is identified and the current alternative interpretation of what the symbol means by major vendors. The fact that major vendors see each of these









<sup>1</sup> <http://www.omg.org/spec/BPMN/2.0.2/PDF/changebarred/>




objects to be part of what is needed to express some aspect of a process shows in effect that our research is correct and reinforces the necessity that the objects be recognized as part of a complete process standard.

eXtended BPMN shapes <sup>2</sup>	The description of the shapes and who uses them (and how):
 <p data-bbox="165 658 256 689">Ruleset</p>	<p data-bbox="421 461 1406 607">((☺)) A Ruleset is a collection and therefore a grouping of rules. These groupings may be based on common related rules, decisions tables, or the need to govern a specific set or behavior of tasks. Grouping rulesets allows for sharing of rules and execution by for example a rule engine.</p> <p data-bbox="421 633 1401 734"><b>SAP</b> A <u>Ruleset</u> is a logical collection of rules. A ruleset helps you group business rules that govern a specific function. A ruleset consists of: If-Then Rules, Decision Tables.</p> <p data-bbox="421 768 1414 913"><b>OD</b> A Ruleset is a container that includes definitions for a group of related rules and decision tables. A ruleset provides a unit of execution for rules and for Decision Tables. In addition, rulesets provide a unit of sharing for rules; rules belong to a ruleset.</p> <p data-bbox="421 943 1398 976"><b>IBM</b> A ruleset is a set of rules and rule artifacts that can be executed by the engine.</p> <p data-bbox="421 1010 1366 1122"><b>MS</b> Contains a collection of Rule classes along with the semantics for forward-chaining execution of those rules. A RuleSet can be executed directly in code or using the PolicyActivity activity.</p>
 <p data-bbox="165 1352 347 1384">Decision Table</p>	<p data-bbox="421 1155 1406 1267">((☺)) Decision tables identify a set of decisions based a series of given conditions and possible actions. Decision tables allow to work from the same information and are therefore a precise way to model complicated logic and associate.</p> <p data-bbox="421 1294 1406 1429"><b>SAP</b> A decision table can either be a tabular representation of related rules or a precise yet compact way to map complicated logic. Decision tables, like flowcharts and if-then-else and switch-case statements, associate conditions with actions to perform, but in many cases do so in a more elegant way.</p> <p data-bbox="421 1462 1414 1641"><b>OD</b> <u>An example</u> of using a Decision table is an activity to evaluate based on the region and the total amount of the order, whether or not the approval of a supervisor is required. Would be good to have this as a reusable Task instead of embedded in the Gateway. The Gateway can then just do the branching. Could include a link to a spreadsheet for example.</p> <p data-bbox="421 1675 1406 1776"><b>IBM</b> Decision tables provide an alternative way of viewing and managing large sets of symmetric business rules. A decision table contains condition columns and actions columns Reference to Business Rules <a href="#">here</a>.</p>












<sup>2</sup> these are not available in BPMN 2.0











<p>eXtended BPMN shapes<sup>2</sup></p>	<p>The description of the shapes and who uses them (and how):</p>
	<p> <i>Decision tables are a simple and clean method of representing large sets of business rules</i></p>
<p> Rule Flow</p>	<p> Rule flows are comprised of linked tasks that contain the instructions as to which rule of a set of rules is to be executed and in what order. The rules are organized into tasks. The ruleflow specifies how tasks are chained together: how, when, and under what conditions they are executed.</p> <p>When you are dealing with many large rule sets, managing the order in which rules are evaluated can become complex. The use of a Rule Flow allows you to specify the order in which rule sets are to be evaluated. It does so by providing you with a flow chart. You will be able to use this chart to define which rule sets should be evaluated in sequence and which in parallel, and to specify conditions under which rule sets should be evaluated. The reuse of rules components and their natural relations, like activity association, are an important aspect of rule modelling. This approach allows for consistency when expressing rules through modelling between the different rule artifacts from the rule flows, rule scripts, rule sets, flow rule sets, rules, and decision tables. To do advanced rule modelling it might be necessary to categorize some of the rules into main and sub rules.</p> <p>A business may contain many rule flows, one of which must be identified as the main rule flow of the project or area. Additional rule flows could then be added and modelled from the main rule flow. Other rule flows are then included through sub flow tasks.</p> <p> A <b>Rule Flow</b> is a sequence of activities for evaluating business rules. The order of the execution of the rules is diagrammatically represented in the form of a flow chart. It is a reusable entity within a flow ruleset, and is based on activities associated with artifacts such as rule scripts, rule flows, rulesets, flow rulesets, rules, and decision tables.</p> <p> Multiple rulesets can be executed in order. This is called rule flow. The ruleset stack determines the order. The order can be manipulated by rule actions that push and pop rulesets on the stack. In rulesets, the priority of rules applies to specify the order in which the rules in the ruleset are applied or “fired”. Rulesets also provide an effective data specification that identifies that the ruleset is always active, or that the ruleset is restricted based on a time and date range, or a starting or ending time and date.</p> <p> A ruleflow defines the flow of execution of the rules. The rules are organized into tasks and the ruleflow specifies how tasks are chained together: how, when, and under what conditions they are executed.</p>
<p></p>	<p> A Rule Script allows you to generate a rule in a scripting environment. This is done through the connection of other activities and the rules that govern these relations and the actions between them. Rule Scripts can furthermore be used with script types and automated test rules, e.g. configurable, programmed, query and event-based network alerts. The link between rule scripts and the mentioned script</p>

eXtended BPMN shapes <sup>2</sup>	The description of the shapes and who uses them (and how):
Rule Script	<p>type could be used as template-based query that can be run against backend tables or database views.</p> <p> A <b>Rule Script</b> is a reusable artifact within a flow ruleset and is a sequence of actions. It is associated with other activities in a flow ruleset, and is triggered when the conditions listed in the preceding activities are satisfied. A rule script can contain any of the following action types: Assign Action, Assert Action, Evaluate Decision, Table Action, Execute Action, Execute Ruleset Action, Execute Rule Action, For Each Action, If Else, If Action, While Action, Retract Action, Break and Continue Actions.</p> <p> The RuleScript allows to generate a rule from a scripting environment.</p>
 Rule	<p> Rules are statements describing a business policy or decision procedure. Some programming languages run business rules together into very complex algorithms. In business process analysis, each rule is usually stated independently, in the general format: If A and B, Then C. Workflow tools and detailed process diagrams both depend on business rules to specify how decisions are made. We generally associate business rules with activities. A decision table is adequate to show what happens if “A” or “B” happens, but dozens or even hundreds of business rules may need to be defined to clarify if “A” or “B” should occur. Training programs, job aids, software systems and knowledge management systems aim to document business rules either to automate the decision process or to and make the rules available to other decision makers.</p> <p> A <b>Rule</b> is a set of conditions and associated actions that are performed when the conditions are satisfied. Rules can be written in two forms: If and then statements, and Decision Tables.</p> <p> Business rules are statements that describe business policies or describe key business decisions.</p> <p> An action rule is made of a condition part and of an action part. The first part of the rule defines the condition in which the rule applies. The second part of the rule defines the action to take if the condition of the rule is true</p> <p> Defines a condition with an associated set of actions to perform. Rules can be related to:</p> <ul style="list-style-type: none"> <li>• Active: Gets or sets a value that indicates whether the Rule should be evaluated.</li> <li>• Condition: Gets or sets a RuleCondition for the Rule to evaluate.</li> <li>• Description: Gets or sets a description of the Rule.</li> <li>• ElseActions: Gets a collection of RuleAction classes to perform in the ELSE case.</li> <li>• Name: Gets or sets the name of the Rule.</li> <li>• Priority: Gets or sets a value that indicates the order in which a Rule should be run.</li> </ul>

eXtended BPMN shapes <sup>2</sup>	The description of the shapes and who uses them (and how):
	<ul style="list-style-type: none"> <li>• Re-evaluation: BehaviorGets or sets a value indicating whether a Rule can be re-evaluated.</li> <li>• ThenActions: Gets a collection of RuleAction classes to perform in the THEN case.</li> </ul>
 <p data-bbox="164 696 323 730">Flow Ruleset</p>	<p data-bbox="419 499 1409 712">((( ))) A flow ruleset is a collection or grouping of rules that apply to a common flow. These flow rulesets are either grouped based on common related rules in the flow, decisions tables or the need to govern a specific set or behavior of tasks in the flow. Grouping such flow rulesets allows for sharing of rules and execution, by a rule engine throughout the entire flow. As well as the relation to the reuse of such rulesets in other flows, starting from the main flow where the ruleset are applied.</p> <p data-bbox="419 741 1409 909"><b>SAP</b> <u>Flow Ruleset</u> is a ruleset that allows you to group business rules that need to be executed as in a flow chart. It always has one flow, designated as main flow, besides possibly many other rule flows. The execution of the Flow Ruleset starts with the main flow. A Flow Ruleset consists of: Aliases, Definition, Rule Flow, Rule Scripts, If-Then Rules, Decision Tables</p>
 <p data-bbox="164 1111 308 1144">Notification</p>	<p data-bbox="419 943 1409 1088">((( ))) An instance of a message delivered to one or more recipients. A notification activity sends a message to a user, both the recipients and the content need to be specified. Notification Activities furthermore allow flow rulesets to be applied in order to notify users of events that occur during the workflow.</p> <p data-bbox="419 1117 1409 1330"><b>SAP</b> A notification provides often through technology a means of delivering a message to a set of recipients. To deploy a notification it is necessary to specify the recipients and then parameterize the subject and message that are to be sent. For example: A notification is to be used to send e-mail notifications from the process. To do this you would specify the recipients, the subject, and message texts that are to be sent through the use of parameters.</p> <p data-bbox="419 1359 1409 1435"><b>SD</b> The <u>notification task</u> allows you to send different types of notifications to the users of the application.</p>
 <p data-bbox="164 1626 276 1659">Mapping</p>	<p data-bbox="419 1458 1409 1648">((( ))) Mapping activities includes identifying, defining, and plotting the business, information, and data objects involved in an activity. This allows the identification of possible grouping of common business, information, and data objects and their associated rules, e.g. ruleset. Enabling the change of complex objects and views into more simple views.</p> <p data-bbox="419 1677 1409 1845"><b>SAP</b> A mapping activity can be used to transform complex objects into more simple views, e.g. map data from data objects in the specific process context making it more simple data. Rules set can be used as a transform in a mapping activity. To make the mapping activity work, the mappings between the business, information, and data objects in the process context need to be defined.</p>



eXtended BPMN shapes <sup>2</sup>	The description of the shapes and who uses them (and how):
 <p>Reporting</p>	<p> Reporting is the exposure, description and portrayal of specific tasks, services and their associated information/ data. The information and data objects source can bind with the reporting activity the process components to the specific cockpits, dashboards, or scorecards.</p> <p>To illustrate and express what is happening or what has happened and could include timely collection, analysis, and then reporting of the information within the process. Depending on whether the reporting is real time reporting or after the fact reporting, the input mapping of the information to the reports needs to be specified.</p> <p> A reporting activity is used to collect data and information so as to perform analytics on this data. The reporting activity references a reporting data source and indicates where in the process data is gathered for reporting. An input mapping needs to be defined for the reporting activity to specify which data from the process context is collected by the reporting activity.</p>
<p>Business Objects</p>  <p>Collection of Business Objects</p>  <p>Business Input</p>  <p>Business Output</p>  <p>Business Store</p> 	<p> A Business Object is a real-world thing having business significance that is used to maintain a persistent source of information or the state of a business process examples include “ people”, “employee”, “products”, or instances of these objects such as a sales order, or a customer and (amount of ) revenue (in a context).</p> <p>Business Objects may be used for example in business, process, or service mapping and/ or in application implementations, where they are used as a semantic layer that shields users from the complexities of information table names and data relationships.</p> <p>Below are the graphical depictions of Business Objects used in business modelling, process mapping, or service mapping as well as in application and architectural mapping, e.g. business architecture, application architecture, and solution architecture:</p> <ul style="list-style-type: none"> <li>• <b>Business Object</b> - Represents real-world objects like people, employee, products or a sales order, customer, and (amount of) revenue (in a context).</li> <li>• <b>Collection of Business Objects</b> - Represents a collection of real-world objects, e.g. an employee roles.</li> <li>• <b>Business Input</b> - Is an external input for the real-world objects. A kind of input parameter.</li> <li>• <b>Business Output</b> - Is the real-world result of the entire process and/ or service flow. A kind of output parameter.</li> <li>• <b>Business Store</b> - The place the business objects are stored</li> </ul>
<p>Information Object</p> 	<p> An Information Object is used to specify information about real-world objects (people, employee, products, or a sales order, etc.) and therefore it is the digital representation of an existing entity in an Information System, e.g. Oracle ERP or SAP.</p> <p>It encompasses both the business information (in the form of functions and methods) and the application information (in the form of attributes) of this entity. Information Objects can be found and therefore modelled in business functions, business service,</p>

eXtended BPMN shapes <sup>2</sup>	The description of the shapes and who uses them (and how):
<p>Collection of Information Objects</p>  <p>Information Input</p>  <p>Information Output</p>  <p>Information Store</p> 	<p>or in processes.</p> <p>Below are the graphical depictions of Information Objects used in business modelling, process mapping, service mapping as well as in application, and architectural mapping, e.g. business architecture, information architecture, application architecture, and solution architecture:</p> <ul style="list-style-type: none"> <li>• <b>Information Object</b> - Represents a container of information within the flow of the process and/ or service, such as business documents, e-mails, or letters.</li> <li>• <b>Collection of Information Objects</b> - Represents a collection of information, e.g. a list of order items.</li> <li>• <b>Information Input</b> - Is an external information input for the entire process. A kind of input parameter.</li> <li>• <b>Information Output</b> - Is the information output/result of the entire process. A kind of output parameter.</li> <li>• <b>Information Store</b> - Is a place where the information can be read or written, e.g. knowledge management or a filing cabinet. It persists beyond the lifetime of the process instance.</li> </ul>
<p>Data Object</p>  <p>Collection of Data Objects</p>  <p>Data Input</p>  <p>Data Output</p>  <p>Data Store</p> 	<p> A Data Object is a logical cluster of all tables in the data set that have one or more columns containing data related to the same business entity. Data objects are not maps, but instead represent an object view of related Information Objects that represent Business Objects.</p> <p>Below are the graphical depictions of Data Objects used in process mapping, service mapping as well as in application and architectural mapping, e.g. information architecture, application architecture, solution architecture, and data architecture:</p> <ul style="list-style-type: none"> <li>• <b>Data Object</b> - Represents a container/logical cluster of all tables in the data set that have one or more columns containing data related to the same data entity, such as business documents, e-mails, or letters.</li> <li>• <b>Collection of Data Objects</b> - Represents a collection of data tables/columns containing data related to the same data entity, e.g. a list of order items.</li> <li>• <b>Data Input</b> - Is an external data input for the entire process. A kind of input parameter.</li> <li>• <b>Data Output</b> - Is the data result of the entire process. A kind of output parameter.</li> <li>• <b>Data Store</b> - Is a place where the process can read or write data, e.g. a database or a filing cabinet. It persists beyond the lifetime of the process instance.</li> </ul>

