BPM+ Knowledge Package Model and Notation (BKPMN)

V1.0 - beta 1

OMG Document Number: dtc/22-01-01
Normative:
Standard Document URL: https://www.omg.org/spec/BKPMN
Machine Consumable file(s):
https://www.omg.org/spec/BKPMN/20211101/BKPMN.xmi
https://www.omg.org/spec/BKPMN/20211101/BKPMN.xsd
https://www.omg.org/spec/BKPMN/20211101/BKPMNDI.xmi
https://www.omg.org/spec/BKPMN/20211101/BKPMNDI.xsd
https://www.omg.org/spec/BKPMN/20211101/BKPMN-Semantic.xsd
https://www.omg.org/spec/BKPMN/20211101/BKPMN-Library.xml

This OMG document replaces the submission document (bmi/21-12-15). It is an OMG Adopted Beta Specification and is currently in the finalization phase. Comments on the content of this document are welcome and should be directed to issues@omg.org by June 24, 2022.

You may view the pending issues for this specification from the OMG revision issues web page https://issues.omg.org/issues/lists.

The FTF Recommendation and Report for this specification will be published in December 2022. If you are reading this after that date, please download the available specification from the OMG Specifications Catalog.
USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

The material in this document details an Object Management Group specification in accordance with the terms, conditions and notices set forth below. This document does not represent a commitment to implement any portion of this specification in any company's products. The information contained in this document is subject to change without notice.

LICENSES

The companies listed above have granted to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version. Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification.

Subject to all of the terms and conditions below, the owners of the copyright in this specification hereby grant you a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide license (without the right to sublicense), to use this specification to create and distribute software and special purpose specifications that are based upon this specification, and to use, copy, and distribute this specification as provided under the Copyright Act; provided that: (1) both the copyright notice identified above and this permission notice appear on any copies of this specification; (2) the use of the specifications is for informational purposes and will not be copied or posted on any network computer or broadcast in any media and will not be otherwise resold or transferred for commercial purposes; and (3) no modifications are made to this specification. This limited permission automatically terminates without notice if you breach any of these terms or conditions. Upon termination, you will destroy immediately any copies of the specifications in your possession or control.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OMG specifications may require use of an invention covered by patent rights. OMG shall not be responsible for identifying patents for which a license may be required by any OMG specification, or for conducting legal inquiries into the legal validity or
General Use Restrictions

Any unauthorized use of this specification may violate copyright laws, trademark laws, and communications regulations and statutes. This document contains information which is protected by copyright. All Rights Reserved. No part of this work covered by copyright herein may be reproduced or used in any form or by any means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems--without permission of the copyright owner.

Disclaimer of Warranty

While this publication is believed to be accurate, it is provided "as is" and may contain errors or misprints. The Object Management Group and the companies listed above make no warranty of any kind, express or implied, with regard to this publication, including but not limited to any warranty of title or ownership, implied warranty of merchantability or warranty of fitness for a particular purpose or use. In no event shall the Object Management Group or any of the companies listed above be liable for errors contained herein or for direct, indirect, incidental, special, consequential, reliance or cover damages, including loss of profits, revenue, data or use, incurred by any user or any third party in connection with the furnishing, performance, or use of this material, even if advised of the possibility of such damages.

The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

Restricted Rights Legend

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227-7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 9C Medway Road, PMB 274, Milford, MA 01757, U.S.A.

Trademarks


For a complete list of trademarks, see: http://www.omg.org/legal/tm_list.htm. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

Compliance

The copyright holders listed above acknowledge that the Object Management Group (acting itself or through its
designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks or other special designations to indicate compliance with these materials.

Software developed under the terms of this license may claim compliance or conformance with this specification if and only if the software compliance is of a nature fully matching the applicable compliance points as stated in the specification. Software developed only partially matching the applicable compliance points may claim only that the software was based on this specification, but may not claim compliance or conformance with this specification. In the event that testing suites are implemented or approved by Object Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.

OMG’s Issue Reporting Procedure

All OMG specifications are subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Form listed on the main web page https://www.omg.org, under Documents, Report a Bug/Issue.
# Table of Contents

1. **Scope** ...................................................................................................................................................... 15
2. **Conformance** .......................................................................................................................................... 15
   2.1 BPM+ Knowledge Package Modeling Conformance ......................................................................... 15
   2.2 Visual Conformance ........................................................................................................................... 15
3. **References** .............................................................................................................................................. 16
   3.1 Normative References ........................................................................................................................ 16
   3.2 Non-normative References ................................................................................................................. 17
4. **Terms and Definitions** ............................................................................................................................ 17
5. **Symbols** .................................................................................................................................................. 19
6. **Additional Information** ........................................................................................................................... 19
   6.1 Conventions ........................................................................................................................................ 19
   6.2 Typographical and Linguistic Conventions and Style ........................................................................ 19
   6.3 Display of Metamodel Diagrams ........................................................................................................ 20
   6.4 Use of Text, Color, Size, and Lines in a Diagram .............................................................................. 21
   6.5 Abbreviations ...................................................................................................................................... 21
   6.6 Structure of this Document ................................................................................................................. 22
   6.7 Acknowledgements ............................................................................................................................. 22
7. **Overview** ................................................................................................................................................ 23
   7.1 Motivation .......................................................................................................................................... 23
   7.2 What is a BPM+ Knowledge Package? .............................................................................................. 23
   7.3 BPM+ Knowledge Package Model Diagram ...................................................................................... 24
8. **Specification Core Elements** .................................................................................................................. 25
9. **BKPMN Metamodel** .............................................................................................................................. 26
   9.1 Packaging........................................................................................................................................... 29
   9.1.1 BKPMNModelPackage ................................................................................................................... 29
   9.1.2 BKPMNModel ................................................................................................................................ 32
   9.1.3 BKPMNDefinitions ......................................................................................................................... 34
   9.1.4 BKPMNInstances ............................................................................................................................ 35
   9.2 BKPMN Vocabularies .......................................................................................................................... 36
   9.2.1 BKPMNVocabulary ....................................................................................................................... 36
10. **Knowledge Elements** ........................................................................................................................... 37
    10.1 KnowledgeElement ............................................................................................................................ 37
    10.2 Effect ................................................................................................................................................ 38
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.3</td>
<td>ImpactKind</td>
<td>40</td>
</tr>
<tr>
<td>10.4</td>
<td>LifecycleEvent</td>
<td>40</td>
</tr>
<tr>
<td>10.5</td>
<td>Measure</td>
<td>42</td>
</tr>
<tr>
<td>10.6</td>
<td>MeasureKind</td>
<td>43</td>
</tr>
<tr>
<td>10.7</td>
<td>RiskFactor</td>
<td>43</td>
</tr>
<tr>
<td>10.8</td>
<td>RiskKind</td>
<td>45</td>
</tr>
<tr>
<td>10.9</td>
<td>Determining Applicability</td>
<td>45</td>
</tr>
<tr>
<td>10.9.1</td>
<td>Applicability</td>
<td>47</td>
</tr>
<tr>
<td>10.10</td>
<td>Inputs and Outputs</td>
<td>48</td>
</tr>
<tr>
<td>10.10.1</td>
<td>BKPMNPackageInput</td>
<td>49</td>
</tr>
<tr>
<td>10.10.2</td>
<td>BKPMNPackageOutput</td>
<td>50</td>
</tr>
<tr>
<td>10.11</td>
<td>Recommendations</td>
<td>50</td>
</tr>
<tr>
<td>10.11.1</td>
<td>Recommendation</td>
<td>50</td>
</tr>
<tr>
<td>10.11.2</td>
<td>RecommendationStatus</td>
<td>52</td>
</tr>
<tr>
<td>10.11.3</td>
<td>RecommendationStrength</td>
<td>53</td>
</tr>
<tr>
<td>11</td>
<td>Package Elements</td>
<td>53</td>
</tr>
<tr>
<td>11.1</td>
<td>PackageElement</td>
<td>53</td>
</tr>
<tr>
<td>11.2</td>
<td>Contributor</td>
<td>54</td>
</tr>
<tr>
<td>11.3</td>
<td>ContributorKind</td>
<td>56</td>
</tr>
<tr>
<td>11.4</td>
<td>Publisher</td>
<td>56</td>
</tr>
<tr>
<td>11.5</td>
<td>ParticipantRef</td>
<td>57</td>
</tr>
<tr>
<td>11.6</td>
<td>PolicyRef</td>
<td>58</td>
</tr>
<tr>
<td>11.7</td>
<td>Purpose</td>
<td>59</td>
</tr>
<tr>
<td>11.8</td>
<td>UsageTerms</td>
<td>60</td>
</tr>
<tr>
<td>12</td>
<td>Manifest Documents</td>
<td>61</td>
</tr>
<tr>
<td>12.1</td>
<td>Manifest</td>
<td>61</td>
</tr>
<tr>
<td>12.2</td>
<td>Operational Documents</td>
<td>62</td>
</tr>
<tr>
<td>12.2.1</td>
<td>OperationalDocumentRef</td>
<td>63</td>
</tr>
<tr>
<td>12.2.2</td>
<td>LanguageProfile</td>
<td>64</td>
</tr>
<tr>
<td>12.2.3</td>
<td>References to BKPMN Elements</td>
<td>65</td>
</tr>
<tr>
<td>12.2.3.1</td>
<td>BKPMNDocumentRef</td>
<td>65</td>
</tr>
<tr>
<td>12.2.3.2</td>
<td>BKPMNElementRef</td>
<td>66</td>
</tr>
<tr>
<td>12.2.4</td>
<td>References to BPMN Elements</td>
<td>68</td>
</tr>
<tr>
<td>12.2.4.1</td>
<td>BPMNDocumentRef</td>
<td>68</td>
</tr>
<tr>
<td>12.2.4.2</td>
<td>BPMNElementRef</td>
<td>69</td>
</tr>
<tr>
<td>12.2.4.3</td>
<td>EventRef</td>
<td>71</td>
</tr>
<tr>
<td>12.2.4.4</td>
<td>ResourceRoleRef</td>
<td>72</td>
</tr>
</tbody>
</table>
12.2.5 References to CMMN Elements
12.2.5.1 CMMNDocumentRef
12.2.5.2 CMMNElementRef
12.2.5.3 MilestoneRef

12.2.6 References to DMN Elements
12.2.6.1 DMNDocumentRef
12.2.6.2 DMNElementRef
12.2.6.3 DecisionRef
12.2.6.4 ExpressionRef

12.2.7 References to MDMI Elements
12.2.7.1 MDMIDocumentRef
12.2.7.2 MDMIElementRef

12.2.8 References to PPMN Elements
12.2.8.1 PPMNDocumentRef
12.2.8.2 PPMNElementRef
12.2.8.3 PMOrganizationRef
12.2.8.4 PMPersonRef

12.2.9 References to SDMN Elements
12.2.9.1 SDMNDocumentRef
12.2.9.2 SDMNElementRef
12.2.9.3 DataItemRef

12.3 Resource Documents
12.3.1 ResourceDocumentRef
12.3.2 DocumentKind
12.3.3 BKPMNPackageSummary
12.3.4 InitializationDataRef
12.3.5 NarrativeRef
12.3.6 NarrativeKind
12.3.7 PolicyDocumentRef
12.3.8 TestCaseRef

13 BPM+ Knowledge Model
13.1 BPMPlusKnowledgeModel
13.1.1 Graphical Elements
13.2 KnowledgeModel
13.3 Model Structural Elements
13.3.1 BKPMNModelElement
Table of Figures

Figure 1: Overview of a BPM+ Knowledge Package .................................................................24
Figure 2: Example of a BPKMNDiagram ..............................................................................25
Figure 3: The Specification Core Elements (SCE) High-Level Metamodel ................................................. 26
Figure 4: BKPMN Main Packages .............................................................................................................. 27
Figure 5: The BKPMN Package High Level Metamodel ........................................................................... 28
Figure 6: The BKPMN Core Element Metamodel .................................................................................... 29
Figure 7: The BKPMNModelPackage Metamodel .................................................................................. 30
Figure 8: The BKPMNModel Metamodel ............................................................................................... 33
Figure 9: The BKPMN Metamodel ........................................................................................................... 34
Figure 10: The BKPMNInstances Metamodel ......................................................................................... 35
Figure 11: The BKPMNVocabulary Metamodel ...................................................................................... 36
Figure 12: The KnowledgeElement Metamodel .................................................................................... 38
Figure 13: The Effects Metamodel ........................................................................................................... 39
Figure 14: The LifecycleEvents Metamodel ......................................................................................... 41
Figure 15: The Measure Metamodel ....................................................................................................... 42
Figure 16: The RiskFactor Metamodel ..................................................................................................... 44
Figure 17: The Applicability Metamodel ............................................................................................... 46
Figure 18: Process for Determining the Applicability of a BKPMN Package ............................................. 46
Figure 19: Decision Model for Determining the Applicability of a BKPMN Package ............................... 47
Figure 20: The metamodel for BKPMNPackageInputs and BKPMNPackageOutputs ............................ 49
Figure 21: The Recommendations Metamodel ...................................................................................... 51
Figure 22: The PackageElement Metamodel .......................................................................................... 54
Figure 23: The Contributor Metamodel .................................................................................................. 55
Figure 24: Publisher MM ......................................................................................................................... 57
Figure 25: The Participants Metamodel ................................................................................................. 58
Figure 26: The PolicyRef Metamodel ..................................................................................................... 59
Figure 27: The Purpose Metamodel ........................................................................................................ 60
Figure 28: The Manifest Metamodel ....................................................................................................... 62
Figure 29: The OperationalDocumentRef Metamodel .......................................................................... 63
Figure 30: The Profile metamodel ........................................................................................................... 64
Figure 31: The BKPMNDocument Metamodel ......................................................................................... 65
Figure 32: The BKPMNElementRef Metamodel .................................................................................... 67
Figure 33: The BPMNDocument Metamodel .......................................................................................... 68
Figure 34: The BPMNElementRef Metamodel ....................................................................................... 70
Figure 35: The EventRef Metamodel ...................................................................................................... 71
Figure 36: The CMMNDocument Metamodel ........................................................................................ 73
Figure 37: The CMMNElementRef Metamodel ...................................................................................... 74
Figure 38: The DMNDocument Metamodel ........................................................................................... 76
Figure 39: The DMNElementRef Metamodel .......................................................................................... 77
Table of Tables

Table 1. Submission Requirements............................................................... Error! Bookmark not defined.
Table 2. Glossary .......................................................................................... 17
Table 3. Acronyms .......................................................................................... 21
Table 4. BKPMNModelPackage Attributes and/or Associations .................. 31
Table 5. BKPMNModel Attributes and/or Associations ................................... 33
Table 6. BKPMNDefinitions Attributes and/or Associations ......................... 35
Table 7. BKPMNInstances Attributes and/or Associations ............................. 36
Table 8. Effect Attributes and/or Associations .............................................. 40
Table 9. LifecycleEvent Attributes and/or Associations ................................. 41
Table 10. Measure Attributes and/or Associations .......................................... 43
Table 11. RiskFactor Attributes and/or Associations ....................................... 44
Table 12. Applicability Attributes and/or Associations .................................... 48
Table 13. BKPMNPackageInput Attributes and/or Associations .................... 50
Table 14. BKPMNPackageOutput Attributes and/or Associations .................. 50
Table 15. Recommendation Attributes and/or Associations .......................... 51
Table 16. Contributor Attributes and/or Associations ..................................... 56
Table 17. Publisher Attributes and/or Associations ........................................ 57
Table 18. PolicyRef Attributes and/or Associations ........................................ 59
Table 19. UsageTerms Attributes and/or Associations ..................................... 61
Table 20. Manifest Attributes and/or Associations .......................................... 62
Table 21. OperationalDocumentRef Attributes and/or Associations ................ 63
Table 22. LanguageProfile Attributes and/or Associations .............................. 65
Table 23. BKPMNDocumentRef Attributes and/or Associations ...................... 66
Table 24. BKPMNElementRef Attributes and/or Associations ........................ 67
Table 25. BPMNDocumentRef Attributes and/or Associations ......................... 69
Table 26. BPMNElementRef Attributes and/or Associations ............................ 71
Table 27. CMMNDocumentRef Attributes and/or Associations ....................... 73
Table 28. CMMNElementRef Attributes and/or Associations ............................ 75
Table 29. DMNDocumentRef Attributes and/or Associations ............................ 76
Table 30. DMNElementRef Attributes and/or Associations ............................. 78
Table 31. ExpressionRef Attributes and/or Associations ................................................................................ 79
Table 32. MDMIDocumentRef Attributes and/or Associations ........................................................................... 79
Table 33. MDMIElementRef Attributes and/or Associations ............................................................................... 81
Table 34. PPMNDocumentRef Attributes and/or Associations ............................................................................ 82
Table 35. PPMNElementRef Attributes and/or Associations .................................................................................. 83
Table 36. SDMNDocumentRef Attributes and/or Associations ............................................................................... 85
Table 37. SDMNElementRef Attributes and/or Associations .................................................................................. 87
Table 38. ResourceDocumentRef Attributes and/or Associations .......................................................................... 90
Table 39. BKPMNPackageSummary Attributes and/or Associations ....................................................................... 91
Table 40. NarrativeRef Attributes and/or Associations ...................................................................................... 92
Table 41. PolicyDocumentRef Attributes and/or Associations ............................................................................. 93
Table 42. BPMPlusKnowledgeModel Attributes and/or Associations ............................................................... 94
Table 43. BPMPlusModelElement Attributes and/or Associations ......................................................................... 99
Table 44. CaseRef Attributes and/or Associations ............................................................................................ 102
Table 45. OperationalModelRef Attributes and/or Associations ........................................................................ 108
Table 46. ProcessRef Attributes and/or Associations ......................................................................................... 110
Table 47. ModelElementConnector Attributes and/or Associations ...................................................................... 115
Table 48. Group Attributes and/or Associations ................................................................................................ 120
Table 49. BKPMNVocabulary Instances ......................................................................................................... 121
Table 50. ItemKind Literals .................................................................................................................................. 123
Table 51. ItemKind Literals .................................................................................................................................. 124
Table 52. BaseElement Attributes and/or Associations ...................................................................................... 125
Table 53. BusinessRuleTask Attributes and/or Associations ................................................................................ 126
Table 54. CaseTask Attributes and/or Associations ........................................................................................... 126
Table 55. Case Attributes and/or Associations ................................................................................................ 127
Table 56. CaseParameter Attributes and/or Associations .................................................................................. 127
Table 57. Decision Attributes and/or Associations ............................................................................................ 128
Table 58. DecisionParameter Attributes and/or Associations ............................................................................ 128
Table 59. CMMNElement Attributes and/or Associations .................................................................................. 129
Table 60. DMNElement Attributes and/or Associations ...................................................................................... 130
Table 61. List of BPM+ Models for the Hello Patient Use Case ............................................................................. 131
Table 62. List of Data Elements used by the BPM+ Models in the Hello Patient Use Case .................................. 133
Table 63. BKPMNDI Attributes and/or Associations .......................................................................................... 138
Table 64. BKPMNDiagramAttributes and/or Associations .................................................................................. 139
Table 65. BKPMNShape Attributes and/or Associations ...................................................................................... 142
Table 66. DI::Edge Attributes and/or Associations ............................................................................................. 143
Table 67. Depiction Resolution of DataItems ................................................................. 144
Table 68. Multiplicity Decorator Depiction ................................................................. 146
Table 69. Semantic Reference Decorator Depiction .................................................... 146
Table 70. Depiction Resolution of the Relationship Connector .................................. 146

**Preface**

**OMG**

Founded in 1989, the Object Management Group, Inc. (OMG) is an open membership, not-for-profit computer industry standards consortium that produces and maintains computer industry specifications for interoperable, portable, and reusable enterprise applications in distributed, heterogeneous environments. Membership includes Information Technology vendors, end users, government agencies, and academia.

OMG member companies write, adopt, and maintain its specifications following a mature, open process. OMG’s specifications implement the Model Driven Architecture® (MDA®), maximizing ROI through a full-lifecycle approach to enterprise integration that covers multiple operating systems, programming languages, middleware and networking infrastructures, and software development environments. OMG’s specifications include: UML® (Unified Modeling Language™); CORBA® (Common Object Request Broker Architecture); CWM™ (Common Warehouse Metamodel); and industry-specific standards for dozens of vertical markets.

More information on the OMG is available at https://www.omg.org/.

**OMG Specifications**

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. All OMG Specifications are available from the OMG website at:

https://www.omg.org/spec

All of OMG’s formal specifications may be downloaded without charge from our website. (Products implementing OMG specifications are available from individual suppliers.) Copies of specifications, available in PostScript and PDF format, may be obtained from the Specifications Catalog cited above or by contacting the Object Management Group, Inc. at:

OMG Headquarters
9C Medway Road
PMB 274
Milford, MA 01757
USA
Tel: +1-781-444-0404
Fax: +1-781-444-0320
Email: pubs@omg.org

Certain OMG specifications are also available as ISO standards. Please consult https://www.iso.org

**Issues**

All OMG specifications are subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Form listed on the main web page https://www.omg.org, under Documents, Report a Bug/Issue.
1 Scope

The focus of this specification is to specify the content and structure of a BPM-Plus (BPM+) Knowledge Package. A BPM+ Knowledge Package is considered a “box” or “wrapper” model that contains models developed through the other BPM+ standards (BPMN, CMMN, and DMN), and other artifacts.

Based on recent experience with the use of the current set of BPM+ standards the need of a packaging mechanism for groups of related BPM+ models was identified (see the use case described in Clause 16.1 for a use case example. For example, using BPM+ models to define a large topic, such as the behaviors of a healthcare clinical guideline (e.g., a hypertension) may result in dozens of individual Process, Case, and Decision models. But there is currently a lack of a mechanism to package the related models with relevant metadata to aid in understanding the content and to aid in the discovery of appropriate BPM+ Knowledge Packages.

A key expectation is that BPM+ Knowledge Packages will be readily distributed to interested parties (i.e., spreading the knowledge). However, unlike a physical package that contains unique physical elements and can be shipped to different locations, a BPM+ Knowledge Package references BPM+ models and other artifacts. These models and artifacts can be re-used by multiple BPM+ Knowledge Packages and, technically, don’t exist “inside” the package. Thus, the “shipping” of a BPM+ Knowledge Package consists of the distribution of a set of model files (either XMI or XML). The file for the BPM+ Knowledge Package will include a manifest that identifies the files that contain the BPM+ models or other artifacts. All the files identified in the manifest will be distributed along with the BPM+ Knowledge Package file. BKPMN is currently focused on the BMI behavioral standards, but supports the inclusion of additional models and documents.

A BPM+ Knowledge Package will also contain metadata that describes the purpose and contents of the Knowledge Package and to help organizations discover the Knowledge Package that is appropriate for their circumstances.

2 Conformance

Software can claim compliance or conformance with BKPMN 1.0 if and only if the software fully matches the applicable compliance points as stated in the specification. In addition, the structural elements provided by Specification Common Elements (SCE 1.0) are also required in a compliant or conformant software solution. Software developed only partially matching the applicable compliance points can claim only that the software was based on this specification but cannot claim compliance or conformance with this specification.

2.1 BPM+ Knowledge Package Modeling Conformance

The implementation claiming conformance to the BPM+ Knowledge Package Modeling Conformance SHALL comply with all of the requirements set forth in Clauses 9 through 13, and it should be conformant with the Visual Notation Conformance in Clause 16 Conformant implementations SHALL fully support and interpret the exchange format specified in Clause 15.

This compliance point is intended to be used by BKPMN modeling tools.

2.2 Visual Conformance

An implementation that creates and displays BKPMN models SHALL conform to the specifications and restrictions with respect to diagrammatic relationships between graphical elements, as described in Clause Error! Reference source not found.. A key element of BKPMN is the choice of shapes and icons used for the graphical elements identified in this specification. The intent is to create a standard visual language that all BPM+ Knowledge Package Data modelers will recognize and understand. An implementation that creates and displays BKPMN models SHALL use the graphical elements, shapes, markers and decorators illustrated in this specification.

There is flexibility in the size, color, line style, and text positions of the defined graphical elements, except where
otherwise specified. In particular:

- **BKPMN** elements MAY have labels (e.g., its name and/or other attributes) placed inside the shape, or above or below the shape, in any direction or location, depending on the preference of the modeler or modeling tool vendor.

- The fills that are used for the graphical elements MAY be white or clear. The notation MAY be extended to use other fill colors to suit the purpose of the modeler or tool (e.g., to highlight the value of an object attribute).

- Graphical elements, shapes, and decorators MAY be of any size that suits the purposes of the modeler or modeling tool with the condition that the additional graphical elements SHALL NOT conflict with any current BPM+ Standard defined graphical element.

- The lines that are used to draw the graphical elements MAY be black.
  - The notation MAY be extended to use other line colors to suit the purpose of the modeler or tool (e.g., to highlight the value of an object attribute).
  - The notation MAY be extended to use other line styles to suit the purpose of the modeler or tool (e.g., to highlight the value of an object attribute) with the condition that the line style SHALL NOT conflict with any current BPM+ Standard defined line style.

The following extensions to a **BKPMN** model are permitted:

- New decorators or indicators MAY be added to the specified graphical elements. These decorators or indicators could be used to highlight a specific attribute of a **BKPMN** element or to represent a new subtype of the corresponding concept with the condition that the additional graphical elements SHALL NOT conflict with any current BPM+ Standard defined decorator or indicator.

- A new shape representing a kind of knowledge element MAY be added to a model with the condition that the shape SHALL NOT conflict with the shape specified for any other BPM+ Standard element or decorator.

- Graphical elements MAY be colored, and the coloring MAY have specified semantics that extend the information conveyed by the element as specified in this 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 specification or the other BPM+ Standards.

- An extension SHALL NOT change the specified shape of a defined graphical element or decorator. (e.g., changing a square into a triangle, or changing rounded corners into squared corners, etc.).

This compliance point is intended to be used by entry-level **BKPMN** tools.

## 3 References

### 3.1 Normative References

The following normative documents contain provisions which, through reference in this text, constitute provisions of this specification. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply.

- Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, IETF RFC 2119, March 1997
  http://www.ietf.org/rfc/rfc2119.txt


- [DD] Diagram Definition (DD™)


- [MDMI] OMG Model Driven Message Interoperability (MDMI), Version 1.0:
  https://www.omg.org/spec/MDMI/
3.2 Non-normative References

The following normative documents contain provisions which, through reference in this text, constitute exemplars or influencers of this specification. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply.


4 Terms and Definitions

The table below presents a glossary for this specification:

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applicability</td>
<td>Applicability describes the conditions for which the Knowledge Package is intended. A DMN Decision Service can be defined to express the formal logic of the applicability conditions.</td>
</tr>
<tr>
<td>Association</td>
<td>A type of Artifact that is a connecting object that is used to indicate there is some relationship between two BKPMN graphical elements. There are no semantics defined by the connector.</td>
</tr>
<tr>
<td>BPM+ Knowledge Model</td>
<td>A diagram that provides a visual representation of the breadth and scope of a Knowledge Package in terms of the BPM+ models it contains.</td>
</tr>
<tr>
<td>BPM+ Knowledge Package</td>
<td>A Knowledge Package is mechanism for packaging and distributing a set of BPM+ models (i.e., the knowledge).</td>
</tr>
<tr>
<td>Case</td>
<td>A CMMN element that involves actions taken regarding a subject in a particular situation to achieve a desired outcome.</td>
</tr>
<tr>
<td>Choreography</td>
<td>A BPMN Element that is an ordered sequence of B2B message exchanges between two or more Participants. In a Choreography there is no central controller, responsible entity, or observer of the Process. The ChoreographyRef element provides a link to a Choreography in a BPMN document.</td>
</tr>
<tr>
<td>Collaboration</td>
<td>A BPMN Element that is a collection of Participants shown as Pools, their interactions as shown by Message Flows, and MAY include Processes within the Pools and/or Choreographies between the Pools. The CollaborationRef element provides a link to a Collaboration in a BPMN document.</td>
</tr>
<tr>
<td>Contributor</td>
<td>A Contributor is a person that assisted in the development of the Knowledge Package. There are different types of Contributors, such as editors, reviewers, etc.</td>
</tr>
<tr>
<td>DataItem</td>
<td>An SDMN element that represents a common definition and structure for the data handling elements of the other BPM+ models.</td>
</tr>
<tr>
<td>Decision</td>
<td>A DMN element that is the act of determining an output value (the chosen option), from a number of input values, using logic defining how the output is determined from the inputs.</td>
</tr>
<tr>
<td>Decision Service</td>
<td>A DMN element that defines reusable logic within a decision model. A decision service exposes one or more decisions from a decision model, a reusable element, a service, which might be consumed (for example).</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DecisionServiceRef</td>
<td>A BKPMN element that provides a link to a Decision Service in a DMN document. The DecisionServiceRef is a graphical object in a &quot;&quot;BPM+ Knowledge Model.</td>
</tr>
<tr>
<td>Development Method</td>
<td>A &quot;mini&quot; KnowledgePackage that is specifically designed for the people, activities, data, and decisions that are required as part of the development of a KnowledgePackage.</td>
</tr>
<tr>
<td>Effect</td>
<td>An expected response to the application of a KnowledgePackage element. In Medicine, for example, a reduction in blood pressure would be an expected response to the application of a medication.</td>
</tr>
<tr>
<td>Event History</td>
<td>The Event History lists the significant events that have occurred during the development of the Knowledge Package.</td>
</tr>
<tr>
<td>Expression</td>
<td>A BKPMN element that uses natural-language text to extract information from the different elements, normally data elements. These Expressions are not executable and are considered underspecified. Expressions are contained (aggregated) within the BKPMN elements that use them.</td>
</tr>
<tr>
<td>Input</td>
<td>An Input represents a key Data Item that is necessary to be present at the start of a performance of a Knowledge Package.</td>
</tr>
<tr>
<td>Manifest</td>
<td>The Manifest is a list of files that contain the components of the Knowledge Package. This will usually include multiple BPM+ documents (e.g., bpmn files). Other files, such as narratives, will also be in the Manifest.</td>
</tr>
<tr>
<td>Model Artifact</td>
<td>A graphical object that provides supporting information about the BPM+ Knowledge Model. However, it does not have any behavioral semantics.</td>
</tr>
<tr>
<td>Order Set</td>
<td>An order set is a group of related orders which a physician can place with a few keystrokes or mouse clicks. An order set allows users to issue prepack’ged &quot;roupes of orders that apply to a specified diagnosis or a particular period of time (Clinfowiki)</td>
</tr>
<tr>
<td>Order Set Handler</td>
<td>A &quot;mini&quot; KnowledgePackage that is specifically designed for the activities, data, and decisions that are required when a Order Set is used in another KnowledgePackage.</td>
</tr>
<tr>
<td>ParticipantRef</td>
<td>A Participant is distinguished from a Performer. Performers are those resources, often Human, that are responsible for performing activities within a Process or a Case.</td>
</tr>
<tr>
<td>Process</td>
<td>A BPMN element that d‘scri’es a sequence or flow of Activities in an organization with the objective of carrying out work. The ProcessRef element provides a link to a Process in a BPMN document.</td>
</tr>
<tr>
<td>Questionnaire Handler</td>
<td>A &quot;mini&quot; KnowledgePackage that is specifically designed for the activities, data, and decisions that are required when a Order Set is used in another KnowledgePackage.</td>
</tr>
<tr>
<td>QuestionnaireHandlerRef</td>
<td>A BKPMN element that provides a link to a QuestionnaireHandler in a BKPMN document, which is a specialized type of Knowledge Package. The QuestionnaireHandlerRef is a graphical object in a BPM+ Knowledge Model.</td>
</tr>
<tr>
<td>Risk Factor</td>
<td>Indicates the impact and probability of loss.</td>
</tr>
<tr>
<td>Semantic Reference</td>
<td>The SemanticReference element allows model elements to be linked to External Sources of Truth, such as ontologies or data models.</td>
</tr>
<tr>
<td>Side Effect</td>
<td>A potential Effect created by an element of the Knowledge Package. A Side Effect is an Effect that is not intended and is often adverse.</td>
</tr>
<tr>
<td>Text Annotation</td>
<td>A type of Model Artifact that are a mechanism for a modeler to provide additional information for the reader of a BKPMN model. The TextAnnotation object can be connected to a specific object on the Model with an Association.</td>
</tr>
</tbody>
</table>
Undefined Behavior

A BKPMN element that is used as a placeholder for one of the other behavioral elements. It is mainly used early in the development of a Knowledge Package. A specific behavior is known to be part of the package, but it is not clear at that point whether the behavior is a Process or a Case. This allows the inclusion of the behavior in the overall context before the modeller is ready to set its type.

5 Symbols

There are no symbols defined in this specification.

6 Additional Information

6.1 Conventions

The section introduces the conventions used in this document. This includes (text) notational conventions and notations for schema components. Also included are designated namespace definitions.

6.2 Typographical and Linguistic Conventions and Style

This document incorporates the following conventions:

- The keywords “MUST,” “MUST NOT,” “REQUIRED,” “SHALL,” “SHALL NOT,” “SHOULD,” “SHOULD NOT,” “RECOMMENDED,” “MAY,” and “OPTIONAL” in this document are to be interpreted as described in RFC-2119.
- A term is a word or phrase that has a special meaning. When a term is defined, the term name is highlighted in bold typeface.
- A reference to another definition, section, or specification is highlighted with underlined typeface and provides a link to the relevant location in this specification.
- A reference to a graphical element is highlighted with a bold, capitalized word (e.g., ProcessRef).
- A reference to a non-graphical element or BKPMN concept is highlighted by being italicized and will be presented with the Times New Roman font (e.g., Manifest).
- A reference to an attribute or model association will be presented with the Courier New font (e.g., Expression).
- Non-normative examples are set off in boxes and accompanied by a brief explanation.
- XML and pseudo code is highlighted with mono-spaced typeface. Different font colors MAY be used to highlight the different components of the XML code.
- The cardinality of any content part is specified using the following operators:
  - [ ] — exactly once
  - [0..1] — 0 or 1
  - [0..*] — 0 or more
  - [1..*] — 1 or more
- Attributes separated by | and grouped within { and } — alternative values
  - <value> — default value
  - <type> — the type of the attribute
6.3 Display of Metamodel Diagrams

The metamodel presented in these sections utilizes the patterns and mechanisms that are used for the current BPM+ specifications. OMG specifications rarely display the entire metamodel of a technical specification in a single diagram. The entire metamodel would be very large, complicated, and hard to follow. Typically, a specification will present sub-sets of the overall metamodel as they apply to specific topics. For example, in the Business Process Model and Notation 2.0 (BPMN) specification there are metamodel diagrams that show the elements relating to activities or data elements. This document will follow that pattern and present sub-sets of a larger metamodel.

The metamodel diagrams are Unified Modeling Language (UML) structure diagrams. In addition to the metamodel, OMG specifications provide XML schemas which map to the metamodels. In general, it is through XML documents that BPM+ models are stored and exchanged.

Further, some of the metamodel elements are references to elements from other specifications. To clarify the owner of the metamodel element, there is a parenthesized text that identifies the model owner of that element. In addition, colors are used to support the text identification of the owner-language of that element. The colors are used as an aid to distinguish the languages but does not represent a normative aspect of the metamodels nor do they add any semantic information about the metamodels.

The table below presents examples of elements used throughout the metamodel diagrams within this specification:

<table>
<thead>
<tr>
<th>Table 1: BKPMN Metamodel Color-Coding</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Element</strong></td>
</tr>
<tr>
<td><strong>SCE Structural Class</strong></td>
</tr>
<tr>
<td><strong>BKPMN General Class</strong></td>
</tr>
<tr>
<td><strong>BKPMN General Class (focus of diagram)</strong></td>
</tr>
<tr>
<td><strong>External Class</strong></td>
</tr>
<tr>
<td><strong>BKPMN Class Instance</strong></td>
</tr>
</tbody>
</table>
6.4 Use of Text, Color, Size, and Lines in a Diagram

Text Annotation objects can be used by the modeler to display additional information about a BKPMNDiagram or attributes of the objects within the diagram.

- Diagram elements MAY have labels (e.g., its name and/or other attributes) placed inside the shape, or above or below the shape, in any direction or location, depending on the preference of the modeler or modeling tool vendor.
- The fills that are used for the graphical elements MAY be white or clear.
  - The notation MAY be extended to use other fill colors to suit the purpose of the modeler or tool (e.g., to highlight the value of an object attribute).
- Diagram elements and markers MAY be of any size that suits the purposes of the modeler or modeling tool.
- The lines that are used to draw the graphical elements MAY be black.
  - The notation MAY be extended to use other line colors to suit the purpose of the modeler or tool (e.g., to highlight the value of an object attribute).
  - The notation MAY be extended to use other line styles to suit the purpose of the modeler or tool (e.g., to highlight the value of an object attribute) with the condition that the line style SHALL NOT conflict with any current defined line style of the diagram.

6.5 Abbreviations

The table below presents a list of acronyms, and their definition, that are used in this specification:

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BHMN</td>
<td>BPM+ Harmonization Model and Notation</td>
</tr>
<tr>
<td>BKPMN</td>
<td>BPM+ Knowledge Package Model and Notation</td>
</tr>
<tr>
<td>BKPMNDI</td>
<td>BPM+ Knowledge Package Model and Notation Diagram Interchange</td>
</tr>
<tr>
<td>BPMN</td>
<td>Business Process Model and Notation</td>
</tr>
<tr>
<td>BPM+</td>
<td>Business Process Management Plus</td>
</tr>
<tr>
<td>CMMN</td>
<td>Case Management Model and Notation</td>
</tr>
<tr>
<td>DC</td>
<td>Diagram Commons</td>
</tr>
<tr>
<td>DD</td>
<td>Diagram Definition</td>
</tr>
<tr>
<td>DI</td>
<td>Diagram Interchange</td>
</tr>
</tbody>
</table>
6.6 Structure of this Document

This document provides a brief introduction to BKPMN and its purpose (see the section entitled “Error! Reference source not found.”). The introduction is followed by normative clauses that define the elements of the specification and their properties and associations (see the sections entitled “BKPMN Metamodel” (Clause 9); “Knowledge Elements” (Clause 10); “Package Elements” (Clause 11); “Manifest Documents” (Clause 12); “BPM+ Knowledge Model” (Clause 13); “BKPMN Library” (Clause 14); “BPM+ Models Harmonization: (Clause 15); and “BKPMN Diagram Interchange” (Clause 18)).

6.7 Acknowledgements

Submitting Organizations (RFP Process)

- Auxilium Technology Group, LLC
- BPM Advantage Consulting, Inc.

Supporting Organizations (RFP Process)

The following organizations support this specification but are not formal submitters:

- Airbus Group
- BookZurman, Inc.
- Camunda Services GmbH
- Department of Veterans Affairs
- FICO
- KnowProcess Limited
- Mayo Clinic
- Red Hat
- Thematix Partners, LLC
- Trisotech
- Xzyos, LLC

Special Acknowledgements

The following persons were members of the core teams that contributed to the content of this specification: John...
7 Overview

The focus of this specification is to define the content and structure of a BPM-Plus (BPM+) Knowledge Package. A BPM+ Knowledge Package is considered a “box” or “wrapper” model that contains models developed through the other BPM+ standards (and other artifacts).

7.1 Motivation

The motivation for defining the current specification for a BPM+ Knowledge Package was derived from work being done for the development healthcare clinical guidelines using the current BPM+ standards. The current set of BPM+ standards are: 

- **BPMN**: Case Management Model and Notation (CMMN);
- **DMN**: Decision Model and Notation (DMN).

The BPM+ Health organization has published the “Field Guide to Shareable Clinical Pathways”. The Field Guide provides the methods and style of modeling with BPMN, CMMN, and DMN for a more formal definition of healthcare clinical guidelines. A Shareable Clinical Pathway is a healthcare domain-specific BPM+ Knowledge Package Model.

During the development of the Field Guide and supporting BPM+ Use Cases, it became apparent that there is no concrete model or structure that is available that would allow developers of Shareable Clinical Pathways to create, discover, and distribute a coherent package that is a Shareable Clinical Pathway.

7.2 What is a BPM+ Knowledge Package?

A BPM+ Knowledge Package has five major components: Processes, Cases, Decisions, Data – the four pillars – and a wrapper that combines them within a package. The wrapper is called a BPM+ Knowledge Package. Together, these five components will make up the next iteration of the set of BPM+ standards. This document specifies a fourth BPM+ standard named **BKPMN**. The fifth BPM+ standard for the data pillar is the proposed Shared Data Model and Notation (SDMN). An optional component for a BPM+ Knowledge Package is another specification that is currently in development: Pedigree and Provenance Model and Notation (PPMN). In addition to these three new BPM+ standards, there is a sixth standard, Specification Core Elements (SCE), that provides a set of common modeling language elements, such as root element and basic packing capabilities. Instead of defining these basic, non-language specific elements within each of the new languages, BKPMN, PPMN, and SDMN are built upon the structures provided by SCE.

The figure below shows the relationships between the six standards that are key to a BPM+ Knowledge Package.
Figure 1: Overview of a BPM+ Knowledge Package

7.3 BPM+ Knowledge Package Model Diagram

The BPM+ Health organization has been defining Shareable Clinical Pathways. In the generic sense, they have been defining a type of Knowledge Package. To support the production of consistent Knowledge Packages and their distribution, this specification defines a new modeling standard named BPM+ Knowledge Package Model and Notation (BKPMN). This standard would follow the structure of other OMG standards and would become a member of the BPM+ list of standards.

Further, a diagram that was similar to Example of a BPKMNDiagram, below, was included for each Shareable Clinical Pathway. At that time, the diagram was called an Architectural Scope diagram.
Figure 2: Example of a BKPMNDiagram

In BKPMN, this diagram is a representation of a KnowledgeModel. The intent of this diagram is to provide a visual depiction of the breadth and scope of a BPM+ Knowledge Package – in terms of the BPM+ models it contains. The diagram is not intended to represent the entire set of contents of the BPM+ Knowledge Package, but only those elements that represent the behavioral aspects of the BPM+ Knowledge Package, particularly the BPM+ models.

8 Specification Core Elements

The BKPMN metamodel elements are built upon (directly reference) the elements of the Specification Core Elements (SCE) metamodel. The SCE metamodel is defined in a separate specification [OMG doc number bmi-2021-12-09] and contains a set of basic metamodel classes that are common to BKPMN, PPMN, and SDMN – and potentially other OMG specifications.

As can be seen in the figure below, SCE defines elements that can be used by BPM+ modeling specifications – that is, the elements are not specific to any particular area of concern, such as data, process, decision, etc. For example, the SCE Documentation element can be used (and is used) in current BPM+ modeling specifications since it is important to allow modelers to provide documentation about a model element.

Because SCE defines these elements, BKPMN does not have to duplicate them in this specification. BKPMN can just create metamodel bindings to the elements in SCE. Thus, throughout this specification, SCE elements may be seen in metamodel diagrams and BKPMN elements will be shown as being specializations of those SCE elements. The SCE and BKPMN metamodel elements will be identified as described in Section 7.3.

The figure below presents a SCE high-level metamodel view of many of the SCE elements that are used in
BKPMN.

Figure 3: The Specification Core Elements (SCE) High-Level Metamodel

9 BKPMN Metamodel

The figure below displays the organization of the main packages of the BPM+ Knowledge Package Model and Notation (BKPMN) metamodel.
Figure 4: BKPMN Main Packages

The high-level view of the overall BKPMN Metamodel displays the key elements of a BKPMNDefinitions (for more details see BKPMNDefinitions, below).
The Core Element Model high-level metamodel, including the `BKPMNModelPackage`, which is like the “Definitions” element in the metamodels that are defined in by the current BPM+ specifications, defines the basic infrastructure elements of a `BKPMNModelPackage` document.

The figure below displays the core elements of BKPMN.
9.1 Packaging

A BKPMN provides mechanisms for packaging and distributing the content of a BPM+ Knowledge Package, which includes a set of BPM+ models (i.e., the knowledge). A BKPMNPackage references separate but connected BPM+ models (BPMN Processes, CMMN Cases, and DMN Decision Services). It should be noted that BKPMN is designed to support the BMI behavioral standards, but other kinds of models can be included in a BKPMNPackage. A BKPMNPackage also references a Data Item library for the data that will be used by the BPM+ models (the proposed standard SDMN defines the elements of the Data Item library). The Manifest contains the references to the documents that contain the various BPM+ Models that are required for the BKPMNPackage.

There are 3 elements that are used for packaging BKPMN content: BKPMNModelPackage, BKPMNModel, and BKPMNDefinitions (see the next three sections).

9.1.1 BKPMNModelPackage

The BKPMNModelPackage class is the outermost containing object for all BKPMN elements. It defines the scope of visibility and the namespace for all contained elements. The interchange of BKPMN files will always be through a BKPMNModelPackage. Specifically, an XML file for a BPM+ Knowledge Model usually would be appended with a “.bkpmn” label.

A BKPMNModelPackage contains two sub-packages: The BKPMNModel, which contains all the semantic content
of the Knowledge Package; and BKPMNDI, which contains all the diagram interchange information for any BKPMN models that are represented through diagrams.

A BKPMNModelPackage also contains metadata about the topic of the package to aid in understanding the content and to find appropriate BKPMNModelPackages.

The following figure shows the BKPMNModelPackage metamodel.

![BKPMNModelPackage Metamodel](image)

**Figure 7:** The BKPMNModelPackage Metamodel

**Generalizations**

The BKPMNModelPackage element inherits the attributes and/or associations of:

- SCEModelPackage (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

**Properties**

The following table presents the additional attributes and/or associations for BKPMNModelPackage:
<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
</table>
| **completed**: boolean [1]
  default: false | This property is set to `true` when a Knowledge Package contains a complete and coherent set of models. This means that all the models that are referenced by the *Manifest* are completed and the dependencies (relationships) between those models are well defined. Further, all the dependencies (relationships) between Knowledge Package elements and the referenced models are also well defined – these dependencies SHALL be validated before the property can set to `true`. When `completed` is still `false` (the default setting), a Knowledge Package can still be discovered and/or distributed, but there is no expectation that the Knowledge Package is valid, complete, or ready to use in practice. |
| **documentKind**: String [0..1] | The `documentKind` setting provides a classification mechanism for the *BKPMNDefinitions*. This classification could be used as part of a search for a particular *BKPMNModelPackage*, for example. |
| **effectivePeriodEnd**: date [0..1] | This attribute defines the end date for when *BKPMNModelPackage* is no longer in effect. |
| **effectivePeriodStart**: date [0..1] | This attribute defines the start date for when *BKPMNModelPackage* is in effect. |
| **jurisdiction**: String [0..*] | This attribute defines the countries or other areas (such as states) where the *BKPMNModelPackage* is in effect. |
| **model**: BKPMNModel [1] | This the *BKPMNModel* sub-package contained within a *BKPMNModelPackage*. This redefines the `model` association of *SCEModelPackage*. |
| **presentation**: BKPMNDI [0..1] | This attribute contains the Diagram Interchange information contained within this package. This redefines the `presentation` association of *SCEModelPackage*. See the section entitled “BKPMN Diagram Interchange,” below, for more information about *BKPMN* diagram interchange. |
| **published** : boolean [1] | This property is set to `true` when a Knowledge Package is “officially” published and available for review and consumption. This is analogous to the publishing of a book or the release of a version of software. The completed property SHALL be `true` before published property can set to `true`. When set to `true`, the content of the Knowledge Package, both internal and referenced, is considered immutable (locked). This includes the versionDate of the Knowledge Package. When the version of the Knowledge Package is updated, then the published property is reset to `false`. Beyond the validation that applies to the complete property, there is no specify mechanism to validate that a Knowledge Package is fully suitable for publishing. The Knowledge Package development team will determine when it is at a level of completeness that warrants setting published to `true`. When published is still `false` (the default setting), a Knowledge Package can still be discovered and/or distributed, but there is no expectation that the Knowledge Package is valid, complete, or ready to use in practice. |

### 9.1.2 BKPMNModel

The `BKPMNModel` element is an intermediate packaging element that is included to follow the packaging pattern set by SCE. In SCE, the `SCEModel` package contains content for `BKPMNDefinitions`, instances, and profiles. **BKPMN** only includes a package for `BKPMNDefinitions`, but if future versions of **BKPMN** contain instances or profiles, the `BKPMNModel` will be the package that will contain this new content. The `BKPMNModel` element is contained within a `BKPMNModelPackage` package.

The following figure shows the `BKPMNModel` metamodel (which includes the standard instance provided by the **BKPMN** Library).
Generalizations

The $BKPMNModel$ element inherits the attributes and/or associations of:

- $SCEModel$ (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The following table presents the additional attributes and/or associations for $BKPMNModel$:

Table 4. $BKPMNModel$ Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bkpmnVocabulary :</td>
<td>This is a list of the $BKPMNVocabularies$ that are included in the $BKPMNModel$.</td>
</tr>
</tbody>
</table>
9.1.3 BKPMNDefinitions

The `BKPMNDefinitions` element is one of the two main containers for the contents of a BPM+ Knowledge Package. It contains the `Manifest`, a set of `KnowledgeElements`, and a set of `BPMPlusKnowledgeModels` to organize the content and define the relationships. See the appropriate sections below for more details on each type of content.

The `BKPMNDefinitions` element is contained within a `BKPMNModel` package.

The figure below displays the `BKPMNDefinitions` metamodel.

![BKPMN Metamodel](image_url)

**Figure 9: The BKPMN Metamodel**

### Generalizations

The `BKPMNDefinitions` element inherits the attributes and/or associations of:

- `SCEDefinitions` (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).
Properties

The following table presents the additional attributes and/or associations for BKPMNDefinitions:

Table 5. BKPMNDefinitions Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>knowledgeElement : KnowledgeElement [0..*]</td>
<td>The knowledgeElement association ... See the section entitled “KnowledgeElement” for the list of elements that are considered KnowledgeElements.</td>
</tr>
<tr>
<td>knowledgeModel : BPMPlusKnowledgeModel [0..*]</td>
<td>This is a list of KnowledgeModels that included in the BKPMNDefinitions. See the section entitled “BPM+ Knowledge Model,” below, for more information.</td>
</tr>
</tbody>
</table>

9.1.4 BKPMNInstances

The BKPMNInstances element is one of the two main containers for the contents of a BPM+ Knowledge Package. It contains a set of PackageElements. See the appropriate sections below for more details on each type of content.

The BKPMNInstances element is contained within a BKPMNModel package.

The figure below displays the BKPMNInstances metamodel.

Figure 10: The BKPMNInstances Metamodel

Generalizations

The BKPMNInstances element inherits the attributes and/or associations of:

- SCEInstances (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties
The following table presents the additional attributes and/or associations for BKPMNDefinitions:

Table 6. BKPMNInstances Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>packageElement : PackageElement [0..*]</td>
<td>The packageElement association ... See the section entitled “PackageElement” for the list of elements that are considered PackageElements.</td>
</tr>
</tbody>
</table>

9.2 BKPMN Vocabularies

Vocabularies (lists of terms) can be added to a model package of a modeling language dependent on SCE. BKPMNVocabularies are sets of terms defined by an external ontology. The terms link to formal definitions for the model elements that are created by the modeling language. The SemanticReference element is used to provide a link to the terms that can be associated with the appropriate model elements. SCEVocabularies are contained within an SCEModel package.

The figure below displays the BKPMNVocabulary metamodel (which includes the standard instance provided by the BKPMN Library).

![ BKPMNVocabulary Metamodel ]

Figure 11: The BKPMNVocabulary Metamodel

9.2.1 BKPMNVocabulary

An BKPMNVocabulary is a list of terms, through the SemanticReference element, that can be used to relate to model elements to the external definition or meaning. The terms themselves do not represent the definitions or meanings
but provide links to an external source. Multiple **BKPMN Vocabularies** can be defined. They are contained in an **SDMN Model**.

Further, **BKPMN Vocabularies** can be used for creating a user-defined list of enumerated values for use within a **BKPMN** (as opposed to a fixed enumeration list). It is up to the **BKPMN** modeling tool to organize the **SDMN Vocabularies** into the appropriate enumerated lists. Since the **SemanticReference** element has a name and the links to **BKPMN Vocabularies** definitions are optional, the list (the “enumeration” **BKPMN Vocabularies**) can be created before the specific external definitions are established.

**BKPMN** has nine pre-defined **BKPMN Vocabularies** for the enumerated terms for the **ContributorKind** element (see the section entitled “**ContributorKind**” for more information), the **DocumentKind** element (see the section entitled “**DocumentKind**” for more information), the **ImpactKind** element (see the section entitled “**ImpactKind**” for more information), the **MeasureKind** element (see the section entitled “**MeasureKind**” for more information), the **NarrativeKind** element (see the section entitled “**NarrativeKind**” for more information), the **Purpose** element (see the section entitled “**Purpose**” for more information), the **RecommendationStatus** element (see the section entitled “**RecommendationStatus**” for more information), the **RecommendationStrength** element (see the section entitled “**RecommendationStrength**” for more information), and the **RiskKind** (see the section entitled “**RiskKind**” for more information).

**Generalizations**

The **BKPMN Vocabularies** element inherits the attributes and/or associations of:

- **SCE Vocabularies** (see the **SCE** Specification for more information [OMG doc number bmi-2021-12-09]).

**Properties**

The **BKPMN Vocabularies** element does not have any additional attributes and/or associations.

### 10 Knowledge Elements

This chapter of the specification will define the **KnowledgeElement** and its subclasses.

#### 10.1 KnowledgeElement

The **KnowledgeElement** abstract class is one of two intermediary classes to help with the organization of multiple elements that are contained within a **BKPMNDefinitions**. Instead of creating separate containment associations to **BKPMNDefinitions**, there is one containment association from **KnowledgeElement** to **BKPMNDefinitions** and then there are a set of concrete subclasses to **KnowledgeElement** (which can be seen in the figure below).

The class is named **KnowledgeElement** to indicate that its subclasses are elements that can be categorized as reflecting some aspect of the knowledge that is contained in the **BKPMNDefinitions**.

The figure below displays the **KnowledgeElement** metamodel.
Generalizations

The KnowledgeElement element inherits the attributes and/or associations of:

- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The KnowledgeElement element does not have any additional attributes and/or associations.

10.2 Effect

An Effect is some change to the environment or data that may occur when the behaviors of BPM+ Knowledge Package is performed. In the healthcare domain, for example, a reduction in blood pressure may be the effect of the application of a medication. Each BKPMNDefinitions or Recommendation can contain a list of Effects.

The figure below displays the Effects metamodel (which includes the standard instance provided by the BKPMN Library).
Figure 13: The Effects Metamodel

Generalizations
The `Effect` element inherits the attributes and/or associations of:
- `KnowledgeElement` (see the section entitled “KnowledgeElement” for more information).

Further, the `KnowledgeElement` element inherits the attributes and/or associations of:
- `ElementType` (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties
The following table presents the additional attributes and/or associations for `Effect`:
Table 7. Effect Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>effectText : String [0..1]</td>
<td>An optional text description of the Effect.</td>
</tr>
<tr>
<td>eventRef : LifecycleEvent [0..*]</td>
<td>This allows the BKPMNDefinitions modeler to associate various BPMN events or CMMN milestones to the Effect. These events can be used to define a type of lifecycle for the Effect.</td>
</tr>
<tr>
<td>impactKindRef : ImpactKind [0..1]</td>
<td>This indicates the kind of impact that the Effect will have when the behaviors of the BKPMNDefinitions or Recommendation are executed. This will be defined through an instance of the ImpactKind class as provided by the predefined instance of the BKPMNVocabulary element (see the section entitled “BKPMN Library” for more information).</td>
</tr>
<tr>
<td>measureRef : Measure [0..*]</td>
<td>This identifies any Measures that are defined to determine whether or not the Effect occurs.</td>
</tr>
<tr>
<td>probability : Integer [0..1]</td>
<td>This is a probability rating that defines how likely the Effect will occur when the behaviors of the BKPMNDefinitions or Recommendation are performed.</td>
</tr>
</tbody>
</table>

10.3 ImpactKind

This class is a type of SemanticReference that serves as the terms for an SDMNVocabulary that indicates the kind of impact that the Effect will have when the behaviors of the BPM+ Knowledge Package or Recommendation are executed. Instead of being defined a fixed enumerated list, the kinds can be defined through a class (ImpactKind) and instances of that class. BKPMN does not provide any pre-defined instances of ImpactKind. These can be added by a modeler or an implementation of BKPMN as terms for the instance of the SDMNVocabulary element named “ImpactKinds” (see the section entitled “BKPMN Library” for more information). Example ImpactKind instances could include: negative; neutral; positive; and unknown.

In practice, when a modeler creates a model with an Effect, the ImpactKind will be instantiated by one of the instances in the Library created for the context of the model.

Generalizations

The ImpactKind element inherits the attributes and/or associations of:

- SemanticReference (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The ImpactKind element does not have any additional attributes and/or associations.

10.4 LifecycleEvent

A LifecycleEvent is a significant circumstance or event, as identified by the BKPMNDefinitions developer, that occurs during the performance of the behaviors of a BKPMNDefinitions. LifecycleEvent is contained within a BKPMNDefinitions and can be referenced by a Recommendation.

Note: lifecycle events will also be found in the Pedigree of the BKPMNDefinitions, if the Pedigree is being tracked.

The following figure shows the LifecycleEvent metamodel.
Figure 14: The LifecycleEvents Metamodel

Generalizations
The LifecycleEvent element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Further, the ElementType element inherits the attributes and/or associations of:

- SCEElement (see the SCE Specification for more information).

Properties
The following table presents the additional attributes and/or associations for LifecycleEvent:

Table 8. LifecycleEvent Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventRef : EventRef [0..*]</td>
<td>This relates the BPMN elements, such as Start Events, that are associated with the LifecycleEvent. It is possible that multiple Process Events, across the Processes of the BKPMNDefinitions, may be associated with the LifecycleEvent. Both EventRefs and MilestoneRefs may be associated with a LifecycleEvent.</td>
</tr>
<tr>
<td>milestoneRef : MilestoneRef [0..*]</td>
<td>This relates the CMMN Milestones that are associated with the LifecycleEvent. It is possible that multiple Case elements, across the Cases of the BKPMNDefinitions, may be associated with the LifecycleEvent. Both EventRefs and MilestoneRefs may be associated with a LifecycleEvent.</td>
</tr>
</tbody>
</table>
10.5 Measure

A Measure is a mechanism to calculate something using an expression. In BKPMN, an Effect can utilize a Measure to determine the extent that the Effect has been realized for a particular Recommendation or the overall BPM+ Knowledge Package. The actual expression will be contained within a DMN model as part of a Decision or Business Knowledge Model. A Measure is contained within a BKPMNDefinitions.

The figure below displays the Measure metamodel (which includes the standard instance provided by the BKPMN Library).

![Measure metamodel](image)

Figure 15: The Measure Metamodel

Generalizations

The Measure element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The following table presents the additional attributes and/or associations for Measure:
Table 9. Measure Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>expressionRef : ExpressionRef [0..1]</td>
<td>This identifies the DMN expression that is used to evaluate the Measure. The expression will be contained within a DMN model as part of a Decision or Business Knowledge Model.</td>
</tr>
<tr>
<td>measureKindRef : MeasureKind [0..1]</td>
<td>The MeasureKind will define the kinds of measures that are used in the context of the BPM+ Knowledge Package. This will be defined through an instance of the MeasureKind class as provided by the predefined instance of the BKPMNVocabulary element (see the section entitled “BKPMN Library” for more information).</td>
</tr>
</tbody>
</table>

10.6 MeasureKind

This class is a type of SemanticReference that serves as the terms for an SDMNVocabulary that indicates the kind of measure used to test the Effect will have when the behaviors of the BPM+ Knowledge Package or Recommendation are executed. Instead of being defined a fixed enumerated list, the kinds can be defined through a class (MeasureKind) and instances of that class. BKPMN does not provide any pre-defined instances of MeasureKind. These can be added by a modeler or an implementation of BKPMN as terms for the instance of the SDMNVocabulary element named “MeasureKinds” (see the section entitled “BKPMN Library” for more information). Example MeasureKind instances could include: quality; process; perception; performance; and input.

In practice, when a modeler creates a model with an Effect, the MeasureKind will be instantiated by one of the instances in the Library created for the context of the model.

Generalizations

The MeasureKind element inherits the attributes and/or associations of:

- SemanticReference (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The MeasureKind element does not have any additional attributes and/or associations.

10.7 RiskFactor

Indicates the impact and probability of loss and can be calculated through an expression. The actual expression will be contained within a DMN model as part of a Decision or Business Knowledge Model. It is contained within a BKPMNDefinitions.

The figure below displays the RiskFactor metamodel (which includes the standard instance provided by the BKPMN Library).
Generalizations
The RiskFactor element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties
The following table presents the additional attributes and/or associations for RiskFactor.

Table 10. RiskFactor Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>determinant : DataItemRef [0..*]</td>
<td>This identifies the data item that serves as the determinant for the RiskFactor.</td>
</tr>
<tr>
<td>expressionRef : ExpressionRef [0..1]</td>
<td>A RiskFactor can be expressed through an expression (e.g., “Age &gt; 75 years”). The actual expression will be contained within a DMN model as part of a Decision or Business Knowledge Model.</td>
</tr>
</tbody>
</table>
10.8 RiskKind

This class is a type of \textit{SemanticReference} that serves as the \texttt{terms} for an \texttt{SDMNVocabulary} that indicates the kind of risk that the \textit{RiskFactor} will have when the behaviors of the BPM+ Knowledge Package or \textit{Recommendation} are executed. Instead of being defined a fixed enumerated list, the kinds can be defined through a class (RiskKind) and instances of that class. BKPMN does not provide any pre-defined instances of RiskKind. These can be added by a modeler or an implementation of BKPMN as terms for the instance of the \texttt{SDMNVocabulary} element named "RiskKinds" (see the section entitled “BKPMN Library” for more information). Example RiskKind instances could include: fraction-of-incidences; hazard-ratio; increases-in-incidences; and relative.

In practice, when a modeler creates a model with an Effect, the RiskKind will be instantiated by one of the instances in the Library created for the context of the model.

Generalizations

The RiskKind element inherits the attributes and/or associations of:

- \texttt{SemanticReference} (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The RiskKind element does not have any additional attributes and/or associations.

10.9 Determining Applicability

Applicability describes the population or conditions to which the BPM+ Knowledge Package(or Recommendation within a BPM+ Knowledge Package) is intended. These conditions, if included, must be evaluated before that BPM+ Knowledge Package or Recommendation is enacted. It is possible that multiple BPM+ Knowledge Packages or Recommendations may be triggered and enacted under the same conditions. A DMN Decision Service can be defined to express the formal logic of the applicability conditions.

The following figure shows the Applicability metamodel.
The infrastructure for an organization that will implement the behaviors of a BPM+ Knowledge Package or Recommendation will likely need to include a Process that will evaluate the applicability of the specific context for the triggered BPM+ Knowledge Package or Recommendation.

The evaluation Process occurs outside the context of the BPM+ Knowledge Package or Recommendation and the same Process could be used for many of the BPM+ Knowledge Package or Recommendation the organization manages. (There could be a few variations of the evaluation process, but then there would have to be some mechanism for executing the appropriate one for a particular BPM+ Knowledge Package or Recommendation. This mechanism would have to be in the BPM+ Knowledge Package or Recommendation metadata).

The figure below presents a representative version of that Process (in the context of healthcare):
Using a Service Task to launch the Case (see "Open Appropriate Case" above) allows the Process to be applicable to any Knowledge Package.

The figure below presents a representative version of Decisions that evaluate the conditions that exist for the evaluation Process shown in the figure above:

![Diagram of Decision Model for Determining the Applicability of a BKPMN Package](image)

**Figure 19: Decision Model for Determining the Applicability of a BKPMN Package**

### 10.9.1 Applicability

The applicability of a BPM+ Knowledge Package or Recommendation is a statement of the conditions that must be met in order for the BPM+ Knowledge Package or Recommendation to be applied. The Applicability element is contained within a BKPMNDefinitions.

#### Generalizations

The Applicability element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

#### Properties

The following table presents the additional attributes and/or associations for Applicability:
Table 11. Applicability Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>applicabilityDecision</td>
<td>This references the DecisionServiceRef that will execute the determination whether or not the given situation is applicable to the BPM+ Knowledge Package or Recommendation. The results of the inclusionDecision and exclusionDecision will provide input into the applicabilityDecision.</td>
</tr>
<tr>
<td>applicabilityText</td>
<td>This is a brief summary of what constitutes the applicability for the BPM+ Knowledge Package or Recommendation. Additional description of the applicability can be provided by the Documentation element.</td>
</tr>
<tr>
<td>exclusionDecision</td>
<td>This references the DecisionServiceRef that will execute the criteria that determines if the current context (preconditions) indicates that the current BPM+ Knowledge Package or Recommendation is not applicable (i.e., should not be performed).</td>
</tr>
<tr>
<td>inclusionDecision</td>
<td>This references the DecisionServiceRef that will execute the criteria that determines if the current context (preconditions) indicates that the current BPM+ Knowledge Package or Recommendation is applicable (i.e., should be performed).</td>
</tr>
<tr>
<td>knowledgePackageApplicability</td>
<td>If true, this specifies that this Applicability definition applies to the overall BPM+ Knowledge Package rather than a Recommendation.</td>
</tr>
</tbody>
</table>

10.10 Inputs and Outputs

In the same way that a specific business process has input and output data items, a BPM+ Knowledge Package can have data items that trigger the performance of the entire Knowledge Package (or a Recommendation within) as well as data items that are produced as a result.

The following figure shows the metamodel for BKPMNPackageInputs and BKPONPackageOutputs.
Figure 20: The metamodel for BKPMNPackageInputs and BKPMNPackageOutputs

10.10.1 BKPMNPackageInput

An BKPMNPackageInput represents a key DataItem that is necessary to be present at the start of a performance of a BPM+ Knowledge Package or a Recommendation. It is contained within a BPM+ Knowledge Package or a Recommendation.

Generalizations

The BKPMNPackageInput element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The following table presents the additional attributes and/or associations for BKPMNPackageInput:
Table 12. BKPMNPackageInput Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataItemRef : DataItemRef [0..1]</td>
<td>This attribute is optional since a BKPMNPackage can be developed before all the information is available. However, before a BKPMNPackage is considered complete, the DataItemRef for the BKPMNPackageInput should be set (e.g., there should be a Shared Data Model within the BKPMNPackage that contains the referenced Data Item).</td>
</tr>
</tbody>
</table>

10.10.2 BKPMNPackageOutput

A BKPMNPackageOutput represents a key Data Item that is a product at the end of a performance of a BKPMNPackage. It is contained within a BKPMNPackage.

Generalizations

The BKPMNPackageOutput element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The following table presents the additional attributes and/or associations for BKPMNPackageOutput:

Table 13. BKPMNPackageOutput Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataItemRef : DataItemRef [0..1]</td>
<td>This attribute is optional since a BKPMNPackage can be developed before all the information is available. However, before a BKPMNPackage is considered complete, the DataItemRef for the BKPMNPackageOutput should be set (e.g., there should be a Shared Data Model within the BKPMNPackage that contains the referenced Data Item).</td>
</tr>
</tbody>
</table>

10.11 Recommendations

The following sections describe Recommendations and related elements.

10.11.1 Recommendation

A BPM+ Knowledge Package can be considered a large-scale Recommendation and can contain many internal Recommendations that may or may not occur. A Recommendation defines, under specific circumstances, a set of behaviors that can have expected Effects.

The figure below displays the Recommendations metamodel (which includes the standard instances provided by the BKPMN Library).
Figure 21: The Recommendations Metamodel

Generalizations

The Recommendation element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The following table presents the additional attributes and/or associations for Recommendation:

**Table 14. Recommendation Attributes and/or Associations**

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>applicabilityRef : Applicability [0..*]</td>
<td>The applicability of a Recommendation is a statement of the conditions that must be met, after a BKPMNPackageStartTrigger occurs, in order for the BKPMNDefinitions to be applied. See the section entitled “Applicability” for more information.</td>
</tr>
<tr>
<td>effectRef : Effect [0..*]</td>
<td>The Effects of a Recommendation are statements as to a change in condition or a result that occurs due to enacting of the behaviors of the Recommendation. See the section entitled “Effect” for more information.</td>
</tr>
<tr>
<td><strong>inputRef</strong> : BKPMNPackageInput [0..*]</td>
<td>This is a list of the key Data Items that are necessary to be present at the start of a performance of a Recommendation. See the section entitled “BKPM Knowledge Package Input” for more information.</td>
</tr>
</tbody>
</table>
| **lifecycleEventRef** : LifecycleEvent [0..*] | These are the significant circumstances or events, as identified by the BPM+ Knowledge Package developer, that occur during the enactment of the Recommendation. See the section entitled “Lifecycle Event” for more information.  
*Note: the lifecycle events will also be found in the Pedigree of the BKPMNDefinitions, if the Pedigree is being tracked.* |
| **outputRef** : BKPMNPackageOutput [0..*] | This is a list of the key Data Items that are a product at the end of a performance of a Recommendation. See the section entitled “BKPM Knowledge Package Output” for more information. |
| **recommendationText** : String [1] | This is the written narrative of the Recommendation. It will often include “We recommend…” Additional discussion or documentation can be included in the Documentation element. |
| **recommendedBehaviorRef** : BPMPlusModelElement [0..*] | This identifies BPM+ model elements (e.g., a Process) that is related to the Recommendation. |
| **relatedRecommendationRef** : Recommendation [0..*] | This identifies another Recommendation that is related to this Recommendation. |
| **riskFactorRef** : RiskFactor [0..*] | The RiskFactors of a Recommendation are statements as to conditions that may result in negative effects or losses as a result of enacting the behaviors of the Recommendation. See the section entitled “Risk Factor” for more information. |
| **statusRef** : RecommendationStatus [0..1] | The RecommendationStatus defines how the Recommendation is related to Recommendations the of previous versions of the BKPMNDefinitions. See the section entitled “Recommendation Status” for more information. |
| **strengthRef** : RecommendationStrength [0..1] | The RecommendationStrength is an evaluation of the confidence that the Recommendation developers have regarding how the Recommendation should be applied. See the section entitled “Recommendation Strength” for more information. |
| **subRecommendationRef** : Recommendation [0..*] | This identifies another Recommendation that is related to this Recommendation. |

### 10.11.2 RecommendationStatus

This class is a type of SemanticReference that serves as the terms for an SDMNVocabulary that defines how the Recommendation is related to Recommendations the of previous versions of the BPM+ Knowledge Package. Instead of being defined a fixed enumerated list, the kinds can be defined through a class (RecommendationStatus) and instances of that class. BKPMN does not provide any pre-defined instances of RecommendationStatus. These can be added by a modeler or an implementation of BKPMN as terms for the instance of the SDMNVocabulary element.
named “RecommendationStatuses” (see the section entitled “BKPMN Library” for more information). Example RecommendationStatus instances could include: amended; deleted; new-added; new-replaced; not-changed.

In practice, when a modeler creates a model with a Recommendation, the RecommendationStatus will be instantiated by one of the instances in the Library created for the context of the model.

**Generalizations**

The RecommendationStatus element inherits the attributes and/or associations of:

- **SemanticReference** (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

**Properties**

The RecommendationStatus element does not have any additional attributes and/or associations.

### 10.11.3 RecommendationStrength

This class is a type of SemanticReference that serves as the terms for an SDMNVocabulary that is an evaluation of the confidence that the Recommendation developers have regarding how the Recommendation should be applied. Instead of being defined a fixed enumerated list, the kinds can be defined through a class (RecommendationStrength) and instances of that class. BKPMN does not provide any pre-defined instances of RecommendationStrength. These can be added by a modeler or an implementation of BKPMN as terms for the instance of the SDMNVocabulary element named “RecommendationStrengths” (see the section entitled “BKPMN Library” for more information). Example RecommendationStrength instances could include: strong-against; strong-for; weak-against; weak-for.

In practice, when a modeler creates a model with a Recommendation, the RecommendationStatus will be instantiated by one of the instances in the Library created for the context of the model.

**Generalizations**

The RecommendationStrength element inherits the attributes and/or associations of:

- **SemanticReference** (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

**Properties**

The RecommendationStrength element does not have any additional attributes and/or associations.

### 11 Package Elements

This chapter of the specification will define the PackageElement and its subclasses.

#### 11.1 PackageElement

The PackageElement abstract class is one of two intermediary classes to help with the organization of multiple elements that are contained within a BKPMNInstances. Instead of creating separate containment associations to BKPMNInstances, there is one containment association from PackageElement to BKPMNInstances and then there are a set of concrete subclasses to PackageElement (which can be seen in the figure below).

The class is named PackageElement to indicate that its subclasses are elements that can be categorized as reflecting some aspect of the package itself that may help in the understanding or discovery of the BKPMNModelPackage.

The figure below displays the PackageElement metamodel.
Figure 22: The PackageElement Metamodel

Generalizations

The PackageElement element inherits the attributes and/or associations of:

- TypedElement (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The PackageElement element does not have any additional attributes and/or associations.

11.2 Contributor

A Contributor is a person that assisted in the development of the BKPMNDefinitions. There are different kinds of Contributors, such as editors, reviewers, etc. A Contributor can contribute to the BKPMNDefinitions itself or any of the content that is referenced by the Manifest. It is contained within a BKPMNInstances.

Note: Contributors will also be found in the Pedigree of the BKPMNDefinitions, if the Pedigree is being tracked.

The figure below displays the Contributor metamodel (which includes the standard instance provided by the BKPMN Library).
Figure 23: The Contributor Metamodel

Generalizations

The Contributor element inherits the attributes and/or associations of:

- PackageElement (see the section entitled “PackageElement” for more information).

Further, the PackageElement element inherits the attributes and/or associations of:

- TypedElement (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The following table presents the additional attributes and/or associations for Contributor:
### 11.3 ContributorKind

This class is a type of `SemanticReference` that serves as the terms for an `SDMNVocabulary` that indicates the kind of `Contributor` that was involved in the development of the BPM+ Knowledge Package. Instead of being defined as a fixed enumerated list, the kinds can be defined through a class (`ContributorKind`) and instances of that class. The `ContributorKind` element inherits the attributes and/or associations of:

- `SemanticReference` (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

In practice, when a modeler creates a model with a `Contributor`, the `ContributorKind` will be instantiated by one of the instances in the Library created for the context of the model.

#### Generalizations

The `ContributorKind` element inherits the attributes and/or associations of:

- `SemanticReference` (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

#### Properties

The `ContributorKind` element does not have any additional attributes and/or associations.

### 11.4 Publisher

A Publisher is an organization that will publish and distribute the BPM+ Knowledge Package. It is contained within a `BKPMNInstances`.

Note: the publisher will also be found in the Pedigree of the `BKPMNPackage`, if the Pedigree is being tracked.
Figure 24: Publisher MM

**Generalizations**

The Publisher element inherits the attributes and/or associations of:

- PackageElement (see the section entitled “PackageElement” for more information).

Further, the PackageElement element inherits the attributes and/or associations of:

- TypedElement (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

**Properties**

The following table presents the additional attributes and/or associations for Publisher:

**Table 16. Publisher Attributes and/or Associations**

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>publisher : PMOrganizationRef</td>
<td>[0..*]</td>
</tr>
</tbody>
</table>

11.5 ParticipantRef

A Participant is distinguished from a Performer. Performers are those resources, often Human, that are responsible for performing activities within a Process or a Case. It is contained within a BKPMNDefinitions.

The following figure shows the Participants metamodel.
Figure 25: The Participants Metamodel

Generalizations

The `ParticipantRef` element inherits the attributes and/or associations of:

- `BPMNElementRef` (see the section entitled “BPMNElementRef” for more information).

Further, the `BPMNElementRef` element inherits the attributes and/or associations of:

- `BPMPlusModelElement` (see the section entitled “BPMPlusModelElement” for more information).

Further, the `BPMPlusModelElement` element inherits the attributes and/or associations of:

- `KnowledgeElement` (see the section entitled “KnowledgeElement” for more information).

Further, the `KnowledgeElement` element inherits the attributes and/or associations of:

- `ElementType` (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The `ParticipantRef` element does not have any additional attributes and/or associations.

11.6 PolicyRef

The `PolicyRef` identifies a policy document that is relevant to the BPM+ Knowledge Package. It is contained within a `BKPMNInstances`.

The following figure shows the `PolicyRef` metamodel.
Figure 26: The PolicyRef Metamodel

Generalizations

The PolicyRef element inherits the attributes and/or associations of:

- PackageElement (see the section entitled “PackageElement” for more information).

Further, the PackageElement element inherits the attributes and/or associations of:

- TypedElement (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The following table presents the additional attributes and/or associations for PolicyRef.

Table 17. PolicyRef Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>policyRef : URI [0..1]</td>
<td>This identifies a specific policy (as identified through the URI) that is contained within the Policy Document.</td>
</tr>
</tbody>
</table>

11.7 Purpose

This class is a type of SemanticReference that serves as the terms for an SDMNVocabulary that indicates the kind of Contributor that was involved in the development of the BPM+ Knowledge Package. Instead of being defined a fixed enumerated list, the kinds can be defined through a class (Purpose) and instances of that class. BKPMN does not provide any pre-defined instances of Purpose. These can be added by a modeler or an implementation of BKPMN as terms for the instance of the SDMNVocabulary element named “Purposes” (see the section entitled “BKPMN Library” for more information). Example Purpose instances could include: assessment; education. This
classification could be used as part of a search for a particular \textit{BKPMNModel}, for example.

In practice, when a modeler creates a BPM+ Knowledge Package, the \textit{Purpose} will be instantiated by one of the instances in the Library created for the context of the model.

The following figure shows the \textit{Purpose} metamodel.

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{purpose_metamodel.png}
\caption{The \textit{Purpose} Metamodel}
\end{figure}

\textbf{Generalizations}

The \textit{Purpose} element inherits the attributes and/or associations of:

- \textit{SemanticReference} (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

\textbf{Properties}

The \textit{Purpose} element does not have any additional attributes and/or associations.

\subsection{11.8 UsageTerms}

The \textit{UsageTerms} defines when and where it is appropriate to perform the behaviors of the BPM+ Knowledge Package. It is contained within a \textit{BKPMNInstances}.

\textbf{Generalizations}

The \textit{UsageTerms} element inherits the attributes and/or associations of:

- \textit{PackageElement} (see the section entitled “\textit{PackageElement}” for more information).

Further, the \textit{PackageElement} element inherits the attributes and/or associations of:

- \textit{TypedElement} (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

\textbf{Properties}
The following table presents the additional attributes and/or associations for UsageTerms:

Table 18. UsageTerms Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rightsDeclaration</td>
<td>String [0..1]</td>
</tr>
</tbody>
</table>

12 Manifest Documents

The Manifest is a list of files that contain the components of the BPM+ Knowledge Package. The following sections will define the Manifest and documents it may contain.

12.1 Manifest

Each BKPMNModel will have one Manifest. Most of the time, a Manifest file will reference multiple BPM+ model files for the BKPMNModel Elements. For example, a BPMN xml file can contain many or all of the BPMN Processes that are included in the BKPMNModel. That is why the Manifest is a separate element in the BKPMNModel metamodel rather than just having a list of the BPM+ models. Further, other files, such as narratives, will be in the Manifest.

The figure below displays the Manifest metamodel (which includes the standard instance provided by the BKPMN Library).
Generalizations

The Manifest element inherits the attributes and/or associations of:

- **SCEElement** (see the section entitled “SCEElement” for more information [OMG doc number bmi-2021-12-09]).

Properties

The following table presents the additional attributes and/or associations for Manifest:

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>operationalDocumentRef : OperationalDocumentRef [0..*]</td>
<td>This association provides the list of BKPMN documents (extension “.bkpmn”) that are part of the BKPMNDefinitions.</td>
</tr>
<tr>
<td>resourceDocumentRef : ResourceDocumentRef [0..*]</td>
<td>This association provides the list of documents containing documents that are used a resource information that help in the understanding and purpose of the BKPMNDefinitions.</td>
</tr>
</tbody>
</table>

12.2 Operational Documents

The OperationalDocumentRef element is an element that provides a common set of attributes for the various kinds of operational or behavioral language documents that are referenced by the Manifest. There are currently 7 document types classes that are specializations of the OperationalDocumentRef element. Those classes are:

- BKPMNDocumentRef,
- BPMNDocumentRef,
- CMMNDocumentRef,
- DMNDocumentRef,
- MDMIDocumentRef,
- PPMNDocumentRef, and
- SDMNDocumentRef.

The figure below displays the OperationalDocumentRef metamodel (which includes the standard instance provided by the BKPMN Library).
12.2.1 OperationalDocumentRef

This element is concrete to allow for non-BPM+ operational documents to be included within the BKPMNModel.

Generalizations

The OperationalDocumentRef element inherits the attributes and/or associations of:

- SCEElement (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The following table presents the additional attributes and/or associations for OperationalDocumentRef:

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>documentLocation : URI [1]</td>
<td>The URI where the Operational Document is located. The documentLocation SHALL be specified in a URI format.</td>
</tr>
</tbody>
</table>
implementationKind : String [1]  

The implementation kind could be derived from the extension of the target file. e.g., a file with an extension of ".bpmn" would be listed as a BPMN implementation. This will be defined through an instance of the ImplementationKind class as provided by the predefined instance of the BKPMNVocabulary element (see the section entitled “BKPMN Library” for more information).

profile : LanguageProfile [0..1]  

The languageProfile identifies the published versions of the types documents within the BPM+ Knowledge Package; particularly those that are models of a standard.

operationalModelRef :  
OperationalModelRef [1..*]  

This identifies the relevant operational model.

### 12.2.2 LanguageProfile

A BPM+ Knowledge Package includes (references) multiple types of modeling languages that represent the expected behaviors (e.g., BPMN, CMMN, DMN). Each of the modeling languages evolve over time and can have multiple versions published. The LanguageProfile element identifies the version of the modeling language for an OperationalDocumentRef.

The figure below displays the LanguageProfile metamodel.

![LanguageProfile Metamodel](image)

**Figure 30:** The Profile metamodel

**Generalizations**

The LanguageProfile element inherits the attributes and/or associations of:

- SCEElement (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

**Properties**

The following table presents the additional attributes and/or associations for Profile:
Table 21. LanguageProfile Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>languageName : String [1]</td>
<td>The name of the modeling language.</td>
</tr>
</tbody>
</table>

12.2.3 References to BKPMN Elements

12.2.3.1 BKPMNDocumentRef

The Manifest lists zero or more BKPMN Documents, which are XML documents, usually with the file extension “.bkpmn.” Within a single BKPMN Document there will be one BKPMNDefinitions. A BKPMNDocumentRef element references the XML document, thus including the BKPMN Document in the Manifest. It is contained within a Manifest through the OperationalDocumentRef.

The figure below displays the BKPMNDocumentRef metamodel.

Figure 31: The BKPMNDocument Metamodel
**Generalizations**

The `BKPMNDocumentRef` element inherits the attributes and/or associations of:

- `OperationalDocumentRef` (see the section entitled “OperationalDocumentRef” for more information).

Further, the `OperationalDocumentRef` element inherits the attributes and/or associations of:

- `SCEElement` (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

**Constraints**

A `BKPMNDocumentRef` is a type of `OperationalDocumentRef`, but is distinguished with this constraint:

- The instance for the element’s `ImplementationKind` SHALL be named “BKPMN”.

**Properties**

The following table presents the additional attributes and/or associations for `BKPMNDocumentRef`:

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bkpmnElementRef</code> : BKPMNElementRef [1..*]</td>
<td>This provides the list of the BKPMN elements that are referenced within the <code>BKPMNDefinitions</code>. There SHALL be at least one element referenced otherwise the <code>BPMNDocumentRef</code> does not need to be included in the <code>Manifest</code>. See the section “BKPMNElementRef,” above, for more information.</td>
</tr>
</tbody>
</table>

**12.2.3.2 BKPMNElementRef**

There is currently 1 class that are specializations of the `BKPMNElementRef` element. That class is: `BKPMNPackageRef`. The `BKPMNElement` of BKPMN is used in this context to provide the flexibility to extend BKPMN with references BKPMN elements that are not on this list. It is contained within a `BKPMNDefinitions`.

The figure below displays the `BKPMNElementRef` metamodel.
Generalizations

The `BKPMNElementRef` element inherits the attributes and/or associations of:

- `BPMPlusModelElement` (see the section entitled “BPMPlusModelElement” for more information).

Further, the `BPMPlusModelElement` element inherits the attributes and/or associations of:

- `KnowledgeElement` (see the section entitled “KnowledgeElement” for more information).

Further, the `KnowledgeElement` element inherits the attributes and/or associations of:

- `ElementType` (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The following table presents the additional attributes and/or associations for `BKPMNElementRef`:

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>kpmnDocumentRef</code> : BKPMNDocumentRef [1]</td>
<td>This identifies the BKPMN Document (through the BKPMNDocumentRef) that contains the BKPMN element referenced by the concrete sub-class of this BKPMNModelElementRef. This document SHALL be identified.</td>
</tr>
</tbody>
</table>
12.2.4 References to BPMN Elements

12.2.4.1 BPMNDocumentRef

The Manifest lists zero or more BPMN Documents, which are XML documents, usually with the file extension “.bpmn.” Within a single BPMN Document there can be multiple Processes, Choreographies, and/or Collaborations. A BPMNDocumentRef element references the XML document, thus including the BPMN Document in the Manifest. It is contained within a Manifest through the OperationalDocumentRef.

The figure below displays the BPMNDocumentRef metamodel.

![BPMNDocument Metamodel](image)

**Figure 33: The BPMNDocument Metamodel**

**Generalizations**

The BPMNDocumentRef element inherits the attributes and/or associations of:

- OperationalDocumentRef (see the section entitled “OperationalDocumentRef” for more information).

Further, the OperationalDocumentRef element inherits the attributes and/or associations of:

- SCEElement (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

**Constraints**

A BPMNDocumentRef is a type of OperationalDocumentRef, but is distinguished with this constraint:

- The instance for the element’s ImplementationKind SHALL be named “BPMN”.

**Properties**
The following table presents the additional attributes and/or associations for \textit{BPMNDocumentRef}:

\begin{table}
\centering
\caption{BPMNDocumentRef Attributes and/or Associations}
\begin{tabular}{|l|l|}
\hline
Property/Association & Description \\
\hline
bpmnElementRef : BPMNElementRef [1..*] & This provides the list of the BPMN elements that are referenced within the BKPMNDefinitions. There SHALL be at least one element referenced otherwise the BPMNDocumentRef does not need to be included in the Manifest. See the section “BPMNElementRef,” above, for more information. \\
\hline
\end{tabular}
\end{table}

12.2.4.2 BPMNElementRef

There are currently 6 classes that are specializations of the BPMNElementRef element. Those classes are: ProcessRef, EventRef, CollaborationRef, ChoreographyRef, ResourceRoleRef, and ParticipantRef. These 6 kinds of BPMN elements have been identified as being relevant for the definition of the BKPMNDefinitions. TheRootElement of BPMN is used in this context to provide the flexibility to extend BKPMN with references BPMN elements that are not on this list. It is contained within a BKPMNDefinitions.

The figure below displays the BPMNElementRef metamodel.
The BPMNElemntRef Metamodel

**Generalizations**

The BPMNElemntRef element inherits the attributes and/or associations of:

- **BPMPlusModelElement** (see the section entitled “BPMPlusModelElement” for more information).

Further, the BPMPlusModelElement element inherits the attributes and/or associations of:

- **KnowledgeElement** (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- **ElementType** (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

**Properties**

The following table presents the additional attributes and/or associations for BPMNElemntRef:
### Table 25. BPMNElementRef Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bpmnDocumentRef</td>
<td>This identifies the BPMN Document (through the BPMNDocumentRef) that contains the BPMN element referenced by the concrete sub-class of this BPMPlusModelElementRef. This document SHALL be identified.</td>
</tr>
</tbody>
</table>

**12.2.4.3 EventRef**

This element identifies a significant event that occurs within one of the behaviors of the BPM+ Knowledge Package (e.g., a Process or a Case). It is referenced by a LifecycleEvent and is contained within a BKPMNDefinitions.

The figure below displays the EventRef metamodel.

![EventRef Metamodel](image)

**Figure 35: The EventRef Metamodel**

**Generalizations**

The EventRef element inherits the attributes and/or associations of:

- BPMNElementRef (see the section entitled “BPMNElementRef” for more information).

Further, the BPMNElementRef element inherits the attributes and/or associations of:

- BPMPlusModelElement (see the section entitled “BPMPlusModelElement” for more information).

Further, the BPMPlusModelElement element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).
Properties
The EventRef element does not have any additional attributes and/or associations.

12.2.4.4 ResourceRoleRef
ResourceRoles are contained within a BPMN Process. In the context of BKPMN, they can be associated with a BPM+ Knowledge Package Contributor.

Generalizations
The ResourceRoleRef element inherits the attributes and/or associations of:
- BPMNElementRef (see the section entitled “BPMNElementRef” for more information).

Further, the BPMNElementRef element inherits the attributes and/or associations of:
- BPMPlusModelElement (see the section entitled “BPMPlusModelElement” for more information).

Further, the BPMPlusModelElement element inherits the attributes and/or associations of:
- KnowledgeElement (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:
- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties
The ResourceRoleRef element does not have any additional attributes and/or associations.

12.2.5 References to CMMN Elements

12.2.5.1 CMMNDocumentRef
The Manifest lists zero or more CMMN Documents, which are XML documents, usually with the file extension “.cmmn.” Within a single CMMN Document there can be multiple Cases. A CMMNDocumentRef element references the XML document, thus including the CMMN Document in the Manifest. It is contained within a Manifest through the OperationalDocumentRef.

The figure below displays the CMMNDocumentRef metamodel.
Generalizations
The CMMNDocumentRef element inherits the attributes and/or associations of:

- **OperationalDocumentRef** (see the section entitled “OperationalDocumentRef” for more information).

Further, the OperationalDocumentRef element inherits the attributes and/or associations of:

- **SCEElement** (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Constraints
A CMMNDocumentRef is a type of OperationalDocumentRef, but is distinguished with this constraint:

- The instance for the element’s ImplementationKind SHALL be named “CMMN”.

Properties
The following table presents the additional attributes and/or associations for CMMNDocumentRef:

**Table 26. CMMNDocumentRef Attributes and/or Associations**

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmmnElementRef : CMMNElementRef [1..*]</td>
<td>This provides the list of the CMMN elements that are referenced within the BKPMNDefinitions. There SHALL be at least one element referenced otherwise the CMMNDocumentRef does not need to be included in the Manifest. See the section “CMMNElementRef,” above, for more information.</td>
</tr>
</tbody>
</table>
12.2.5.2 CMMNEElementRef

There are currently 4 classes that are specializations of the CMMNEElementRef element. Those classes are: CaseRef, MilestoneRef, ExitCriterionRef, and CaseRoleRef. These 4 kinds of CMMN elements have been identified as being relevant for the definition of the BKPMNDefinitions. The CMMNEElement of CMMN is used in this context to provide the flexibility to extend BKPMN with references CMMN elements that are not on this list. It is contained within a BKPMNDefinitions.

The figure below displays the CMMNEElementRef metamodel.

![CMMNEElementRef Metamodel](image)

**Generalizations**

The CMMNEElementRef element inherits the attributes and/or associations of:

- BPMPlusModelElement (see the section entitled “BPMPlusModelElement” for more information).

Further, the BPMPlusModelElement element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

**Properties**

The following table presents the additional attributes and/or associations for CMMNEElementRef:
Table 27.  CMMNElementRef Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmmnDocumentRef :</td>
<td>This identifies the CMMN Document (through the CMMNDocumentRef) that contains the CMMN element referenced by the concrete sub-class of this BPMPlusModelElementRef. This document SHALL be identified.</td>
</tr>
</tbody>
</table>

12.2.5.3 MilestoneRef

This element identifies a significant milestone that occurs within one of the behaviors of the BPM+ Knowledge Package (Case). It is referenced by a LifecycleEvent and is contained within a BKPMNDefinitions.

Generalizations

The MilestoneRef element inherits the attributes and/or associations of:

- CMMNElementRef (see the section entitled “CMMNElementRef” for more information).

Further, the CMMNElementRef element inherits the attributes and/or associations of:

- BPMPlusModelElement (see the section entitled “BPMPlusModelElement” for more information).

Further, the BPMPlusModelElement element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The MilestoneRef element does not have any additional attributes and/or associations.

12.2.6 References to DMN Elements

12.2.6.1 DMNDocumentRef

The Manifest lists zero or more DMN Documents, which are XML documents, usually with the file extension “.dmn.” Within a single DMN Document there can be multiple Decision Services. A DMNDocumentRef element references the XML document, thus including the DMN Document in the Manifest. It is contained within a Manifest through the OperationalDocumentRef.

The figure below displays the DMNDocumentRef metamodel.
**Generalizations**

The `DMNDocumentRef` element inherits the attributes and/or associations of:

- `OperationalDocumentRef` (see the section entitled “OperationalDocumentRef” for more information).

Further, the `OperationalDocumentRef` element inherits the attributes and/or associations of:

- `SCEElement` (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

**Constraints**

A `DMNDocumentRef` is a type of `OperationalDocumentRef`, but is distinguished with this constraint:

- The instance for the element’s `ImplementationKind` SHALL be named “DMN”.

**Properties**

The following table presents the additional attributes and/or associations for `DMNDocumentRef`:

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dmnElementRef</code> : DMNElementRef [1..*]</td>
<td>This provides the list of the DMN elements that are referenced within the BKPMNDefinitions. There SHALL be at least one element referenced otherwise the <code>DMNDocumentRef</code> does not need to be included in the Manifest. See the section “DMNElementRef,” above, for more information.</td>
</tr>
</tbody>
</table>
12.2.6.2 DMNElementRef

There are currently 3 classes that are specializations of the DMNElementRef element. Those classes are: DecisionServiceRef, DecisionRef, and ExpressionRef. These 3 kinds of DMN elements have been identified as being relevant for the definition of the BKPMNDefinitions. The DMNElement of DMN is used in this context to provide the flexibility to extend BKPMN with references DMN elements that are not on this list. It is contained within a BKPMNDefinitions.

The figure below displays the DMNElementRef metamodel.

![The DMNElementRef Metamodel](image)

**Generalizations**

The DMNElementRef element inherits the attributes and/or associations of:

- **BPMPlusModelElement** (see the section entitled “BPMPlusModelElement” for more information).

Further, the BPMPlusModelElement element inherits the attributes and/or associations of:

- **KnowledgeElement** (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- **ElementType** (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).
Properties

The following table presents the additional attributes and/or associations for DMNElementRef:

Table 29. DMNElementRef Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dmnDocumentRef : DMNDocumentRef [1]</td>
<td>This identifies the CMMN Document (through the CMMNDocumentRef) that contains the CMMN element referenced by the concrete sub-class of this BPMPlusModelElementRef. This document SHALL be identified.</td>
</tr>
</tbody>
</table>

12.2.6.3 DecisionRef

This element identifies a significant decision that occurs within one of the behaviors of the BPM+ Knowledge Package (a Decision Model). It may be used in place of a DecisionServiceRef and is contained within a BKPMNDefinitions.

Generalizations

The DecisionRef element inherits the attributes and/or associations of:

- DMNElementRef (see the section entitled “DMNElementRef” for more information).

Further, the DMNElementRef element inherits the attributes and/or associations of:

- BPMPlusModelElement (see the section entitled “BPMPlusModelElement” for more information).

Further, the BPMPlusModelElement element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “ KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The DecisionRef element does not have any additional attributes and/or associations.

12.2.6.4 ExpressionRef

BKPMN does not directly support the definition of executable expressions. If a BKPMNDefinitions developer requires an executable expression, then the capabilities of DMN should be used. The BKPMN Expression can then be linked to a DMN Expression that is part of the BKPMNDefinitions. The expression will be contained within a DMN model as part of a Decision or Business Knowledge Model.

Generalizations

The ExpressionRef element inherits the attributes and/or associations of:

- DMNElementRef (see the section entitled “DMNElementRef” for more information).

Further, the DMNElementRef element inherits the attributes and/or associations of:

- BPMPlusModelElement (see the section entitled “BPMPlusModelElement” for more information).

Further, the BPMPlusModelElement element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “ KnowledgeElement” for more information).
Further, the KnowledgeElement element inherits the attributes and/or associations of:

- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The following table presents the additional attributes and/or associations for ExpressionRef:

**Table 30. ExpressionRef Attributes and/or Associations**

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dmnExpressionKind</td>
<td>This identifies the type of expression that is being referenced. See the DMN specification for the possible kinds of expressions.</td>
</tr>
</tbody>
</table>

12.2.7 References to MDMI Elements

12.2.7.1 MDMIDocumentRef

The Manifest lists zero or more MDMI (Model Driven Message Interoperability) Documents, which are XML documents, usually with the file extension “.mdmi.” Within a single MDMI Document there will be multiple MDMI models. A MDMIDocumentRef element references the XML document, thus including the MDMI Document in the Manifest. It is contained within a Manifest through the OperationalDocumentRef.

Generalizations

The MDMIDocumentRef element inherits the attributes and/or associations of:

- OperationalDocumentRef (see the section entitled “OperationalDocumentRef” for more information).

Further, the OperationalDocumentRef element inherits the attributes and/or associations of:

- SCEElement (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Constraints

A MDMIDocumentRef is a type of OperationalDocumentRef; but is distinguished with this constraint:

- The instance for the element’s ImplementationKind SHALL be named “MDMI”.

Properties

The following table presents the additional attributes and/or associations for MDMIDocumentRef:

**Table 31. MDMIDocumentRef Attributes and/or Associations**

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mdmiElementRef</td>
<td>This provides the list of the MDMI elements that are referenced within the BKPMNDefinitions. There SHALL be at least one element referenced otherwise the MDMIDocumentRef does not need to be included in the Manifest. See the section “MDMIElementRef,” above, for more information.</td>
</tr>
</tbody>
</table>
12.2.7.2 MDMIElementRef

The *MDMIElement* of MDMI is used in this context to provide the flexibility to extend BKPMN with references MDMI elements that have not been added to BKPMN. It is contained within a BKPMNDefinitions.

The figure below displays the MDMIElementRef metamodel.

![The MDMIElementRef Metamodel](image)

**Generalizations**

The *MDMIElementRef* element inherits the attributes and/or associations of:

- *BPMPlusModelElement* (see the section entitled “BPMPlusModelElement” for more information).

Further, the *BPMPlusModelElement* element inherits the attributes and/or associations of:

- *KnowledgeElement* (see the section entitled “KnowledgeElement” for more information).

Further, the *KnowledgeElement* element inherits the attributes and/or associations of:

- *ElementType* (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

**Properties**

The following table presents the additional attributes and/or associations for *MDMIElementRef*:
Table 32. MDMIElementRef Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mdmiDocumentRef :</td>
<td>This identifies the MDMI Document (through the MDMIDocumentRef) that contains the MDMI element referenced by the concrete sub-class of this BPMPlusModelElement. This document MUST be identified.</td>
</tr>
</tbody>
</table>

12.2.8 References to PPMN Elements

12.2.8.1 PPMNDocumentRef

The Manifest lists zero or more PPMN Documents, which are XML documents, usually with the file extension “.ppmn.” Within a single PPMN Document there will be one Pedigree and Provenance Model. A PPMNDocumentRef element references the XML document, thus including the PPMN Document in the Manifest. It is contained within a Manifest through the OperationalDocumentRef.

The figure below displays the PPMNDocumentRef metamodel.

![Figure 41: PPMNDocumentRef Metamodel](image)

Generalizations

The PPMNDocumentRef element inherits the attributes and/or associations of:

- OperationalDocumentRef (see the section entitled “OperationalDocumentRef” for more information).
Further, the `OperationalDocumentRef` element inherits the attributes and/or associations of:

- `SCEElement` (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

### Properties

The following table presents the additional attributes and/or associations for `PPMNDocumentRef`:

**Table 33. PPMNDocumentRef Attributes and/or Associations**

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
</table>
| `ppmnElementRef` : PPMNElementRef [1..*] | This provides the list of the PPMN elements that are referenced within the `BKPMNDefinitions`. There SHALL be at least one element referenced otherwise the `PPMNDocumentRef` does not need to be included in the `Manifest`. See the section “PPMNElementRef,” above, for more information.

#### 12.2.8.2 PPMNElementRef

There are currently 3 classes that are specializations of the `PPMNElementRef` element. Those classes are: `PedigreeAndProvenanceRef`, `PMOrganizationRef`, and `PMPersonRef`. The `PPMNElementRef` of PPMN is used in this context to provide the flexibility to extend BKPMN with references PPMN elements that are not on this list. It is contained within a `BKPMNDefinitions`.

The figure below displays the `PPMNElementRef` metamodel.
The \texttt{PPMNElementRef} element inherits the attributes and/or associations of:

- \texttt{BPMPlusModelElement} (see the section entitled “\texttt{BPMPlusModelElement}” for more information).

Further, the \texttt{BPMPlusModelElement} element inherits the attributes and/or associations of:

- \texttt{KnowledgeElement} (see the section entitled “\texttt{KnowledgeElement}” for more information).

Further, the \texttt{KnowledgeElement} element inherits the attributes and/or associations of:

- \texttt{ElementType} (see the \texttt{SCE} Specification for more information [OMG doc number bmi-2021-12-09]).

\textbf{Properties}

The following table presents the additional attributes and/or associations for \texttt{PPMNElementRef}:

\begin{table}[h]
\centering
\begin{tabular}{|l|l|}
\hline
Property/Association & Description \\
\hline \texttt{ppmnDocumentRef} : & This identifies the \texttt{PPMN} Document (through the \texttt{PPMNDocumentRef}) that contains the \texttt{PPMN} element referenced by the concrete sub-class of this \texttt{BPMPlusModelElement}. This document MUST be identified. \\
\texttt{PPMNDocumentRef} [1] & \\
\hline
\end{tabular}
\end{table}

\subsection{12.2.8.3 \texttt{PMOrganizationRef}}

\textbf{Generalizations}

The \texttt{PMOrganizationRef} element inherits the attributes and/or associations of:

- \texttt{PPMNElementRef} (see the section entitled “\texttt{PPMNElementRef}” for more information).

Further, the \texttt{PPMNElementRef} element inherits the attributes and/or associations of:

- \texttt{BPMPlusModelElement} (see the section entitled “\texttt{BPMPlusModelElement}” for more information).

Further, the \texttt{BPMPlusModelElement} element inherits the attributes and/or associations of:

- \texttt{KnowledgeElement} (see the section entitled “\texttt{KnowledgeElement}” for more information).

Further, the \texttt{KnowledgeElement} element inherits the attributes and/or associations of:

- \texttt{ElementType} (see the \texttt{SCE} Specification for more information [OMG doc number bmi-2021-12-09]).

\textbf{Properties}

The \texttt{PMOrganizationRef} element does not have any additional attributes and/or associations.

\subsection{12.2.8.4 \texttt{PMPersonRef}}

\textbf{Generalizations}

The \texttt{PMPersonRef} element inherits the attributes and/or associations of:
- **PPMNElementRef** (see the section entitled “PPMNElementRef” for more information). Further, the **PPMNElementRef** element inherits the attributes and/or associations of:
  - **BPMPlusModelElement** (see the section entitled “BPMPlusModelElement” for more information). Further, the **BPMPlusModelElement** element inherits the attributes and/or associations of:
    - **KnowledgeElement** (see the section entitled “KnowledgeElement” for more information). Further, the **KnowledgeElement** element inherits the attributes and/or associations of:
      - **ElementType** (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

**Properties**

The **PMPersonRef** element does not have any additional attributes and/or associations.

### 12.2.9 References to SDMN Elements

#### 12.2.9.1 SDMNDocumentRef

The **Manifest** lists zero or more SDMN Documents, which are XML documents, usually with the file extension “.sdmn.” Within a single SDMN Document there will be one Shared Data Model. A **SDMNDocumentRef** element references the XML document, thus including the SDMN Document in the **Manifest**. It is contained within a **Manifest** through the **OperationalDocumentRef**.

The figure below displays the **SDMNDocumentRef** metamodel.

![The SDMNDocument Metamodel](image-url)

Figure 43: The SDMNDocument Metamodel
**Generalizations**

The `SDMNDocumentRef` element inherits the attributes and/or associations of:

- `OperationalDocumentRef` (see the section entitled “OperationalDocumentRef” for more information).

Further, the `OperationalDocumentRef` element inherits the attributes and/or associations of:

- `SCEElement` (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

**Constraints**

A `SDMNDocumentRef` is a type of `OperationalDocumentRef`, but is distinguished with this constraint:

- The instance for the element’s `ImplementationKind` SHALL be named “SDMN”.

**Properties**

The following table presents the additional attributes and/or associations for `SDMNDocumentRef`:

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sdmnElementRef</code> : SDMNEElementRef [1..*]</td>
<td>This provides the list of the SDMN elements that are referenced within the <code>BKPMNDefinitions</code>. There SHALL be at least one element referenced otherwise the <code>SDMNDocumentRef</code> does not need to be included in the <code>Manifest</code>. See the section “SDMNElementRef,” above, for more information.</td>
</tr>
</tbody>
</table>

12.2.9.2 `SDMNElementRef`

There are currently 2 classes that are specializations of the `SDMNElementRef` element. Those classes are: `DataItemRef`, and `SharedDataModelRef`. The `SDMNElement` of SDMN is used in this context to provide the flexibility to extend BKPMN with references SDMN elements that are not on this list. It is contained within a `BKPMNDefinitions`.

The figure below displays the `SDMNElementRef` metamodel.
Figure 44: The SDMNElementRef Metamodel

Generalizations

The SDMNElementRef element inherits the attributes and/or associations of:

- **BPMPlusModelElement** (see the section entitled “BPMPlusModelElement” for more information).

Further, the BPMPlusModelElement element inherits the attributes and/or associations of:

- **KnowledgeElement** (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- **ElementType** (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The following table presents the additional attributes and/or associations for SDMNElementRef:
Table 36. SDMNElmentRef Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdmnDocumentRef : SDMNDocumentRef [1]</td>
<td>This identifies the SDMN Document (through the SDMNDocumentRef) that contains the SDMN element referenced by the concrete sub-class of this BPMPlusModelElementRef. This document SHALL be identified.</td>
</tr>
</tbody>
</table>

12.2.9.3 DataItemRef

This element identifies a significant data items that occurs within the behaviors of the BPM+ Knowledge Package (e.g., Process, Case, or Decision Model). It is referenced by a RiskFactor, BKPMNPackageInput, and BKPMNPackageOutput and is contained within a BKPMNDefinitions.

The figure below displays the DataItemRef metamodel.

**Figure 45: The DataItemRef Metamodel**

**Generalizations**

The DataItemRef element inherits the attributes and/or associations of:

- SDMNElmentRef (see the section entitled “SDMNElmentRef” for more information).

Further, the SDMNElmentRef element inherits the attributes and/or associations of:
• **BPMPlusModelElement** (see the section entitled “BPMPlusModelElement” for more information). Further, the **BPMPlusModelElement** element inherits the attributes and/or associations of:
  • **KnowledgeElement** (see the section entitled “KnowledgeElement” for more information). Further, the **KnowledgeElement** element inherits the attributes and/or associations of:
    • **ElementType** (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

**Properties**
The **DataItemRef** element does not have any additional attributes and/or associations.

### 12.3 Resource Documents

The **ResourceDocumentRef** element is an element that provides a common set of attributes for the various kinds of supporting documents that are referenced by the **Manifest**. There are currently 4 document kinds classes that are specializations of the **ResourceDocumentRef** element. Those classes are:

• **BKPMNPackageSummary**
• **InitializationDataRef**,  
• **NarrativeRef**,  
• **PolicyDocumentRef**, and  
• **TestCaseRef**.

The figure below displays the **ResourceDocumentRef** metamodel (which includes the standard instance provided by the **BKPMN Library**).
12.3.1 ResourceDocumentRef

As kind of element, ResourceDocumentRefs are not graphically shown on a KnowledgeModel. The concrete elements that are specializations of ResourceDocumentRef are contained within a Manifest.

Generalizations

The ResourceDocumentRef element inherits the attributes and/or associations of:

- SCEElement (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The following table presents the additional attributes and/or associations for ResourceDocumentRef:
Table 37. ResourceDocumentRef Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>documentLocation : URI [1]</td>
<td>The URI where the RelatedResourceRef is located. The documentLocation SHALL be specified in a URI format.</td>
</tr>
<tr>
<td>documentKindRef : DocumentKind [0..1]</td>
<td>This will be defined through an instance of the DocumentKind class as provided by the predefined instance of the BKPMN Vocabularies element (see the section entitled “BKPMN Library” for more information).</td>
</tr>
</tbody>
</table>

12.3.2 DocumentKind

This class is a type of SemanticReference that serves as the terms for an SDMN Vocabularies that indicates the kind of document that the ResourceDocumentRef is. Instead of being defined as a fixed enumerated list, the kinds can be defined through a class (DocumentKind) and instances of that class. BKPMN does not provide any pre-defined instances of DocumentKind. These can be added by a modeler or an implementation of BKPMN as terms for the instance of the SDMN Vocabularies element named “DocumentKinds” (see the section entitled “BKPMN Library” for more information). Example DocumentKind instances could include: 5WH Library; citation; composed-of; depends-on; derived-from; documentation; education; initialization-data; justification; narrative; policy-document; predecessor; successor; supporting-evidence; test-case.

In practice, when a modeler creates a model with a ResourceDocumentRef, the DocumentKind will be instantiated by one of the instances in the Library created for the context of the model.

Generalizations

The DocumentKind element inherits the attributes and/or associations of:

- SemanticReference (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The DocumentKind element does not have any additional attributes and/or associations.

12.3.3 BKPMN Package Summary

This is a document that provides and description and summary of the BPM+ Knowledge Package. It is contained within a Manifest.

Generalizations

The BKPMN Package Summary element inherits the attributes and/or associations of:

- ResourceDocumentRef (see the section entitled “ResourceDocumentRef” for more information).

Further, the ResourceDocumentRef element inherits the attributes and/or associations of:

- SCEElement (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Constraints

A BKPMN Package Summary is a type of ResourceDocumentRef, but is distinguished with this constraint:

- The instance for the element’s DocumentKind SHALL be named “BKPMN Package Summary”.
Properties
The following table presents the additional attributes and/or associations for BKPMNPackageSummary:

Table 38. BKPMNPackageSummary Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>language : String [0..1]</td>
<td>This attribute sets the written language used for the BKPMNPackageSummary. If this attribute is not set, then the setting from the BKPMNDefinitions element will be used.</td>
</tr>
</tbody>
</table>

### 12.3.4 InitializationDataRef

These documents provide information about the initialization of data items relevant to the BPM+ Knowledge Package. It is contained within a Manifest through the ResourceDocumentRef.

Generalizations
The InitializationDataRef element inherits the attributes and/or associations of:

- ResourceDocumentRef (see the section entitled “ResourceDocumentRef” for more information).

Further, the ResourceDocumentRef element inherits the attributes and/or associations of:

- SCEElement (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Constraints
A InitializationDataRef is a type of ResourceDocumentRef, but is distinguished with this constraint:

- The instance for the element’s DocumentKind SHALL be named “InitializationData”.

Properties
The InitializationDataRef element does not have any additional attributes and/or associations.

### 12.3.5 NarrativeRef

Narratives are text or video documents that provide supporting information to help understand the behaviors of a BKPMNDefinitions. They add supporting descriptions that would help someone navigate the models or understand why they are structured the way that they are. It is contained within a Manifest through the ResourceDocumentRef.

Generalizations
The NarrativeRef element inherits the attributes and/or associations of:

- ResourceDocumentRef (see the section entitled “ResourceDocumentRef” for more information).

Further, the ResourceDocumentRef element inherits the attributes and/or associations of:

- SCEElement (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Constraints
A NarrativeRef is a type of ResourceDocumentRef, but is distinguished with this constraint:
The instance for the element’s DocumentKind SHALL be named “Narrative”.

Properties
The following table presents the additional attributes and/or associations for NarrativeRef:

Table 39. NarrativeRef Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>narrativeKindRef : NarrativeRef[0..*]</td>
<td></td>
</tr>
</tbody>
</table>

12.3.6 NarrativeKind

This class is a type of SemanticReference that serves as the terms for an SDMNVocabulary that indicates the kind of document that the NarrativeRef is. Instead of being defined a fixed enumerated list, the kinds can be defined through a class (NarrativeKind) and instances of that class. BKPMN does not provide any pre-defined instances of NarrativeKind. These can be added by a modeler or an implementation of BKPMN as terms for the instance of the SDMNVocabulary element named “NarrativeKinds” (see the section entitled “BKPMN Library” for more information). Example NarrativeKind instances could include: background; discussion; scope.

In practice, when a modeler creates a model with a NarrativeRef, the NarrativeKind will be instantiated by one of the instances in the Library created for the context of the model.

Generalizations
The NarrativeKind element inherits the attributes and/or associations of:

- SemanticReference (see the section entitled “SemanticReference” for more information).

Further, the SemanticReference element inherits the attributes and/or associations of:

- SCEElement (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties
The NarrativeType element does not have any additional attributes and/or associations.

12.3.7 PolicyDocumentRef

A PolicyDocumentRef is a document that contains one or more policies that are relevant to the BPM+ Knowledge Package. It is contained within a Manifest through the ResourceDocumentRef.

Generalizations
The PolicyDocumentRef element inherits the attributes and/or associations of:

- ResourceDocumentRef (see the section entitled “ResourceDocumentRef” for more information).

Further, the ResourceDocumentRef element inherits the attributes and/or associations of:

- SCEElement (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Constraints
A PolicyDocumentRef is a type of ResourceDocumentRef, but is distinguished with this constraint:
• The instance for the element’s DocumentKind SHALL be named “PolicyDocument”.

Properties
The following table presents the additional attributes and/or associations for PolicyDocumentRef:

Table 40. PolicyDocumentRef Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>policyRef : PolicyRef [0..*]</td>
<td>This identifies the policies used in the BKPMNDefinitions that are contained in the policy document referenced by the PolicyDocumentRef.</td>
</tr>
</tbody>
</table>

12.3.8 TestCaseRef

A TestCaseRef is a document that contains test case data about the topic and influenced the content of the BPM+ Knowledge Package. It is contained within a Manifest through the ResourceDocumentRef.

Generalizations
The TestCaseRef element inherits the attributes and/or associations of:

• ResourceDocumentRef (see the section entitled “ResourceDocumentRef” for more information).

Further, the ResourceDocumentRef element inherits the attributes and/or associations of:

• SCEElement (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Constraints

A TestCaseRef is a type of ResourceDocumentRef, but is distinguished with this constraint:

• The instance for the element’s DocumentKind SHALL be named “TestCase”.

Properties

The TestCaseRef element does not have any additional attributes and/or associations.

13 BPM+ Knowledge Model

The topics addressed by a BPM+ Knowledge Package may be complex and may involve a large number of Process, Cases, and Decisions. These models also interact with each other. A simple Manifest that lists the files that contain all the behavioral models does not provide an adequate view of the scope and nature of the Knowledge Package.

Early work on prototype BPM+ Knowledge Packages found that a diagram of the behavioral model elements (the Processes, Cases, and Decisions) and how they interact provided a good representation of the content and scope of the Knowledge Package.

The following figure shows the BPMPlusKnowledgeModel metamodel.
13.1 BPMPlusKnowledgeModel

A model that can be presented with a visual representation of the breadth and scope of a BKPMNDefinitions in terms of the BPM+ models it contains. It is contained within a BKPMNDefinitions.

**Generalizations**

The BPMPlusKnowledgeModel element inherits the attributes and/or associations of:

- BKPMNDefinitions (see the section entitled “BKPMNDefinitions” for more information).

Further, the BKPMNDefinitions element inherits the attributes and/or associations of:

- SCEDefinitions (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

**Properties**

The following table presents the additional attributes and/or associations for BPMPlusKnowledgeModel:

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bkpmnModelElementRef : BKPMNModelElement [0..*]</td>
<td>This is a list of the BKPMNModelElements (BKPMNPackageRefs, CaseRefs, ChoreographyRefs, CollaborationRefs, DecisionServiceRefs, and ProcessRefs) that are within the BPMPlusKnowledgeModel.</td>
</tr>
<tr>
<td>connector : ModelElementConnector [0..*]</td>
<td>This is a list of the ModelElementConnectors that are within the BPMPlusKnowledgeModel.</td>
</tr>
</tbody>
</table>

13.1.1 Graphical Elements

The table below displays the graphical elements of the representation of a KnowledgeModel:
Table 1: Knowledge Model Graphical Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case</td>
<td>This represents one of the Cases that is part of the BPM+ Knowledge Package. In the BKPMN model, it is a reference to a Case within a CMMN file that can be used in multiple BKPMNDefinitions.</td>
<td><img src="CMMN" alt="Case" /></td>
</tr>
<tr>
<td>Choreography</td>
<td>This represents the Choreography model that is part of the BPM+ Knowledge Package. In the BKPMN model, it is a reference to a Choreography within a BPMN file that can be used in multiple BKPMNDefinitions. There should be only one Choreography model for a given BPM+ Knowledge Package.</td>
<td><img src="BPMN" alt="Choreography" /></td>
</tr>
<tr>
<td>Collaboration</td>
<td>This represents a Collaboration model that is part of the BPM+ Knowledge Package. In the BKPMN model, it is a reference to a Collaboration within a BPMN file that can be used in multiple BKPMNDefinitions.</td>
<td><img src="BPMN" alt="Collaboration" /></td>
</tr>
<tr>
<td>Decision Service</td>
<td>This represents one of the Decision Service that is part of the BPM+ Knowledge Package. In the BKPMN model, it is a reference to a Decision Service within a DMN file that can be used in multiple BKPMNDefinitions.</td>
<td><img src="DMN" alt="Decision Service" /></td>
</tr>
<tr>
<td>BPM+ Knowledge Package Model</td>
<td>This represents other BKPMNPackage that interact with this BKPMNDefinitions.</td>
<td><img src="KPMN" alt="BPM+ Knowledge Package Model" /></td>
</tr>
<tr>
<td>Order Set</td>
<td>An Order Set can be considered a “mini BKPMNPackage.” Because of its special place in healthcare situations, it is given its own model element and notation. Note that this object is specific to the healthcare domain.</td>
<td><img src="KPMN" alt="Order Set" /></td>
</tr>
<tr>
<td>Process</td>
<td>This represents one of the Processes that is part of the BPM+ Knowledge Package. In the BKPMN model, it is a reference to a Process within a BPMN file that can be used in multiple BKPMNPackages.</td>
<td><img src="BPMN" alt="Process" /></td>
</tr>
<tr>
<td>Model Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Pedigree and Provenance Model</td>
<td>This represents a <strong>PPMN</strong> model that is part of the BPM+ Knowledge Package.</td>
<td></td>
</tr>
<tr>
<td>Shared Data Model</td>
<td>This represents an <strong>SDMN</strong> model that is part of the BPM+ Knowledge Package.</td>
<td></td>
</tr>
<tr>
<td>Questionnaire</td>
<td>A Questionnaire can be considered a “mini BKPMNPackage. Because of its special place in healthcare situations, it is given its own model element and notation. While this object is often used in the healthcare domain, it can be used in other business domains.</td>
<td></td>
</tr>
<tr>
<td>Undefined Behavior</td>
<td>The Undefined Behavior element is used as a placeholder for one of the other behavioral elements. It is mainly used early in the development of a <strong>BKPMNDefinitions</strong>. A specific behavior is known to be part of the pathway, but it is not clear at that point whether the behavior is a Process or a Case. This allows the inclusion of the behavior in the overall context before the modeler is ready to set its type.</td>
<td></td>
</tr>
<tr>
<td>Model Link</td>
<td>A <strong>ModelLink</strong> is used to show a relationship (calling) between behavioral model elements of a <strong>KnowledgeModel</strong> (see the section entitled “ModelLink”, below).</td>
<td></td>
</tr>
<tr>
<td>Reference Link</td>
<td>A <strong>ReferenceLink</strong> is used to show a relationship (referencing) between behavioral model elements of a <strong>KnowledgeModel</strong> (see the section entitled “ReferenceLink”, below).</td>
<td></td>
</tr>
<tr>
<td>Association (see the SCE specification)</td>
<td>The <strong>Association</strong> connector, defined in SCE, allows a <strong>KnowledgeModel</strong> developer to connect two objects in the diagram. The connection does not have any semantic or behavioral meaning, but just shows there is a relationship between the two objects. The <strong>Association</strong> is typically used with a <strong>Text Annotation</strong> to association text with an object (see table row below).</td>
<td></td>
</tr>
<tr>
<td>Text Annotation (attached with an Association) (see the SCE specification)</td>
<td><strong>Text Annotations</strong>, defined in SCE, are a mechanism for a modeler to provide additional information for the reader of a <strong>KnowledgeModel</strong>.</td>
<td></td>
</tr>
</tbody>
</table>
13.2 KnowledgeModel

A KnowledgeModel is the specific model that can be notated in BKPMN. It is the concrete specialization of the abstract BPMPlusKnowledgeModel element. In future versions of BKPMN, additional concrete models may be added to represent additional types of information contained in a BPMPlusKnowledgeModel. A KnowledgeModel is contained in a BKPMNDefinitions.

Generalizations

The KnowledgeModel element inherits the attributes and/or associations of:

- BPMPlusKnowledgeModel (see the section entitled “BPMPlusKnowledgeModel” for more information).

Further, the BPMPlusKnowledgeModel element inherits the attributes and/or associations of:

- BKPMNDefinitions (see the section entitled “BKPMNDefinitions” for more information).

Further, the BKPMNDefinitions element inherits the attributes and/or associations of:

- SCEDefinitions (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The KnowledgeModel element does not have any additional attributes and/or associations.

13.3 Model Structural Elements

The following figure shows the Knowledge Model structural elements through BKPMNModelElement metamodel.
13.3.1 BKPMNModelElement

This class is included in the metamodel to serve as a mechanism to group seven other classes and provide a single association for the BKPMNModel class.

Generalizations

The BKPMNModelElement element does not inherit any attributes or associations of from another element.

Properties

The BKPMNModelElement element does not have any additional attributes and/or associations.

13.3.2 BPMPlusModelElement

A BPMPlusModelElement is the abstract class that provides the properties for locating specific BPM+ language elements that can be found in the files referenced within the Manifest. Its concrete subclasses are contained within a BKPMNDefinitions. For example, a ProcessRef is a concrete specialization of BPMPlusModelElement and it is used to represent a Process that is contained within a BPMN document referenced in the Manifest.

Generalizations

The BPMPlusModelElement element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09] [OMG doc number bmi-2021-12-09]).
Properties

The following table presents the additional attributes and/or associations for \texttt{BPMPlusModelElement}:

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{incoming} : Connector [0..*]</td>
<td>This attribute identifies the incoming \texttt{ModelElementConnector} of the \texttt{BKPMNDefinitions}.</td>
</tr>
<tr>
<td>\texttt{modelElement} : QName [0..1]</td>
<td>This identifies the \texttt{id} (QName) of the element that the concrete \texttt{BPMPlusModelElement} represents. The QName will be found in the appropriate model definition file referenced in the \texttt{Manifest}.</td>
</tr>
<tr>
<td>\texttt{modelElementName} : String [0..1]</td>
<td>This identifies the name of the element of the element that the concrete \texttt{BPMPlusModelElement} represents. The name is derived from the details provided by the \texttt{ModelElement}.</td>
</tr>
<tr>
<td>\texttt{modelElementType} : String [0..1]</td>
<td>This identifies the type of the element. The type SHALL match a defined model element type that is derived from the details provided by the \texttt{ModelElement}.</td>
</tr>
<tr>
<td>\texttt{outgoing} : Connector [0..*]</td>
<td>This attribute identifies the outgoing \texttt{ModelElementConnector} of the \texttt{BKPMNDefinitions}.</td>
</tr>
<tr>
<td>\texttt{policyRef} : PolicyRef [0..*]</td>
<td>This identifies any policies that are related to the \texttt{BPMPlusModelElement}.</td>
</tr>
</tbody>
</table>

13.4 Knowledge Model Elements

The figure below displays the metamodel that highlights the \texttt{KnowledgeModel} graphical elements.

![Knowledge Model Elements Metamodel](image)

Figure 49: The Knowledge Model Elements Metamodel
13.4.1  BKPMNPackageRef

The figure below displays the BKPMNPackageRef metamodel.

![BKPMNPackageRef Metamodel](image)

**Generalizations**

The BKPMNPackageRef element inherits the attributes and/or associations of:

- BKPMNElementRef (see the section entitled “BKPMNElementRef” for more information).

Further, the BKPMNElementRef element inherits the attributes and/or associations of:

- BPMPlusModelElement (see the section entitled “BPMPlusModelElement” for more information).

Further, the BPMPlusModelElement element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

In addition, the BKPMNPackageRef element inherits the attributes and/or associations of:

- BKPMNModelElement (see the section entitled “BKPMNModelElement” for more information).

**Properties**
The **BKPMNPackageRef** element does not have any additional attributes and/or associations.

### 13.4.2 CaseRef

This represents one of the Cases that is part of the **BKPMNDefinitions**. In the **BKPMN** model, it is a reference to a Case within a **CMMN** file. It is contained within a **BKPMNDefinitions**.

The figure below displays the **CaseRef** metamodel.

![Figure 51: The CaseRef Metamodel](image)

**Notation**

The following statements define the notation for a **CaseRef**:

- A **CaseRef** is a shape that SHALL be a suitcase shape drawn with a dashed single line (see below).
  - The use of text, color, size, and lines for a **Reference Connector** SHALL follow the rules defined in the section entitled “Use of Text, Color, Size, and Lines in a Diagram” above.

![Figure 52: A CaseRef Object](image)
Generalizations

The **CaseRef** element inherits the attributes and/or associations of:

- **BPMPlusDiagramElement** (see the section entitled “BPMPlusDiagramElement” for more information).

In addition, the **CaseRef** element inherits the attributes and/or associations of:

- **CMMNElemnetRef** (see the section entitled “CMMNElemnetRef” for more information).

Further, the **CMMNElemnetRef** element inherits the attributes and/or associations of:

- **BPMPlusModelElement** (see the section entitled “BPMPlusModelElement” for more information).

Further, the **BPMPlusModelElement** element inherits the attributes and/or associations of:

- **KnowledgeElement** (see the section entitled “KnowledgeElement” for more information).

Further, the **KnowledgeElement** element inherits the attributes and/or associations of:

- **ElementType** (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The following table presents the additional attributes and/or associations for **CaseRef**:

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>startTrigger : LifecycleEvent [0..1]</td>
<td>This identifies the Event that triggers the start of the overall Knowledge Package.</td>
</tr>
</tbody>
</table>

**13.4.3 ChoreographyRef**

This represents the Choreography model that is part of the BKPMNDefinitions. In the BKPMN model, it is a reference to a Choreography within a BPMN file that can be used in multiple BKPMNDefinitions. It is contained within a Definitions.

There should be only one Choreography model for a given BKPMNDefinitions.

The figure below displays the ChoreographyRef metamodel:
The **ChoreographyRef** element inherits the attributes and/or associations of:

- **BPMNElementRef** (see the section entitled “BPMNElementRef” for more information).

Further, the **BPMNElementRef** element inherits the attributes and/or associations of:

- **BPMPlusModelElement** (see the section entitled “BPMPlusModelElement” for more information).

Further, the **BPMPlusModelElement** element inherits the attributes and/or associations of:

- **KnowledgeElement** (see the section entitled “KnowledgeElement” for more information).

Further, the **KnowledgeElement** element inherits the attributes and/or associations of:

- **ElementType** (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

In addition, the **ChoreographyRef** element inherits the attributes and/or associations of:

- **BPMPlusDiagramElement** (see the section entitled “BPMPlusDiagramElement” for more information).

### Properties

The **ChoreographyRef** element does not have any additional attributes and/or associations.

#### 13.4.4 CollaborationRef

This represents a Collaboration model that is part of the **BKPMNDefinitions**. In the **BKPMN** model, it is a reference to a Collaboration within a **BPMN** file that can be used in multiple **BKPMNDefinitions**. A Collaboration will be linked to one or more Process elements in a **BKPMNDefinitions**. It is contained within a **BKPMNDefinitions**.
The figure below displays the CollaborationRef metamodel.

**Generalizations**

The CollaborationRef element inherits the attributes and/or associations of:

- BPMNElementRef (see the section entitled “BPMNElementRef” for more information).

Further, the BPMNElementRef element inherits the attributes and/or associations of:

- BPMPlusModelElement (see the section entitled “BPMPlusModelElement” for more information).

Further, the BPMPlusModelElement element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

In addition, the CollaborationRef element inherits the attributes and/or associations of:

- BPMPlusDiagramElement (see the section entitled “BPMPlusDiagramElement” for more information).
The **CollaborationRef** element does not have any additional attributes and/or associations.

### 13.4.5 DevelopmentMethod

Many **BKPMNDefinitions** (particularly in healthcare) will contain the details of the methodology that was used to create the **BKPMNDefinitions**. This type of information is also captured in the Pedigree of business elements. It is contained within a **BKPMNDefinitions**.

The figure below displays the **DevelopmentMethod** metamodel.

![DevelopmentMethod Metamodel](image)

**Figure 55: The Development Method Metamodel**

**Generalizations**

The **DevelopmentMethod** element inherits the attributes and/or associations of:

- **BKPMNPackageRef** (see the section entitled “BKPMNPackageRef” for more information).

Further, the **BKPMNPackageRef** element inherits the attributes and/or associations of:

- **BKPMNElementRef** (see the section entitled “BKPMNElementRef” for more information).

Further, the **BKPMNElementRef** element inherits the attributes and/or associations of:

- **BPMPlusModelElement** (see the section entitled “BPMPlusModelElement” for more information).

Further, the **BPMPlusModelElement** element inherits the attributes and/or associations of:

- **KnowledgeElement** (see the section entitled “KnowledgeElement” for more information).

Further, the **KnowledgeElement** element inherits the attributes and/or associations of:

- **ElementType** (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

In addition, the **SCEElement** element inherits the attributes and/or associations of:

- **BPMPlusDiagramElement** (see the section entitled “BPMPlusDiagramElement” for more information).

**Properties**

The **DevelopmentMethod** element does not have any additional attributes and/or associations.

### 13.4.6 DecisionServiceRef

This represents one of the Decision Services that is part of the **BKPMNDefinitions**. In the **BKPMN** model, it is a
reference to a Decision Service within a DMN file that can be used in multiple BKPMNDefinitions. It is contained within a BKPMNDefinitions.

The figure below displays the DecisionServiceRef metamodel.

Figure 56: The DecisionServiceRef Metamodel

Generalizations

The DecisionServiceRef element inherits the attributes and/or associations of:

- BPMPlusDiagramElement (see the section entitled “BPMPlusDiagramElement” for more information).

In addition, the DecisionServiceRef element inherits the attributes and/or associations of:

- DMNElementRef (see the section entitled “DMNElementRef” for more information).

Further, the DMNElementRef element inherits the attributes and/or associations of:

- BPMPlusModelElement (see the section entitled “BPMPlusModelElement” for more information).

Further, the BPMPlusModelElement element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:
- **ElementType** (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

**Properties**

The **DecisionServiceRef** element does not have any additional attributes and/or associations.

### 13.4.7 OperationalModelRef

The figure below displays the **OperationalModelRef** metamodel.

![OperationalModelRef Metamodel](image)

**Figure 57: The OperationalModelRef Metamodel**

**Generalizations**

The **OperationalModelRef** element inherits the attributes and/or associations of:

- **KnowledgeElement** (see the section entitled “**KnowledgeElement**” for more information).

Further, the **KnowledgeElement** element inherits the attributes and/or associations of:

- **ElementType** (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

In addition, the **OperationalModelRef** element inherits the attributes and/or associations of:

- **BPMPlusDiagramElement** (see the section entitled “**BPMPlusDiagramElement**” for more information).
Properties

The following table presents the additional attributes and/or associations for OperationalModelRef:

Table 44. OperationalModelRef Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>modelElement : QName [0..1]</td>
<td>This identifies the id (QName) of the element that the concrete OperationalModelRef represents. The QName will be found in the appropriate model definition file referenced in the Manifest.</td>
</tr>
<tr>
<td>modelElementType : String [0..1]</td>
<td>This identifies the type of element that is being referenced (e.g., Task, Decision, CaseFileItem, etc.).</td>
</tr>
<tr>
<td>\operationalDocumentRef : OperationalDocumentRef [1]</td>
<td>The implementationKind of the referenced OperationalDocumentRef SHALL NOT be of the types that are listed as the specializations of the OperationalDocumentRef element, such as BPMN and CMMN, etc. The figure above displays all the document types that are excluded from being referenced by OperationalModelRef element.</td>
</tr>
</tbody>
</table>

13.4.8 ProcessRef

This represents one of the BPMN Processes that is part of the BKPMNDefinitions. In the BKPMN model, it is a reference to a Process within a BPMN file that can be used in multiple BKPMNDefinitions. It is contained within a BKPMNDefinitions.

This element points to a Process that is contained within a BPMN file (BKPMNDefinitions).

This element is graphically represented in a KnowledgeModel.

- The Process Knowledge Element has an “startTrigger” attribute.
  - Any Process where this is True is technically outside the Knowledge Package. But it is the Process that triggers the top-level Case of the BPM+ Knowledge Package. It should be shown in the Knowledge Package for documentation purposes.
  - As small circle (like a BPMN Start Event) is added to the right side of the Process Knowledge Element shape to mark that this is a trigger Process (see figure 1).
  - There can be multiple Processes that are triggers for the Knowledge Package.

The figure below displays the ProcessRef metamodel.
Figure 58: The ProcessRef Metamodel

Generalizations

The ProcessRef element inherits the attributes and/or associations of:

- BPMNElementRef (see the section entitled “BPMNElementRef” for more information).

Further, the BPMNElementRef element inherits the attributes and/or associations of:

- BPMPlusModelElement (see the section entitled “BPMPlusModelElement” for more information).
Further, the **BPMPlusModelElement** element inherits the attributes and/or associations of:

- **KnowledgeElement** (see the section entitled "KnowledgeElement" for more information).

Further, the **KnowledgeElement** element inherits the attributes and/or associations of:

- **ElementType** (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

In addition, the **ProcessRef** element inherits the attributes and/or associations of:

- **BPMPlusDiagramElement** (see the section entitled “BPMPlusDiagramElement” for more information).

**Properties**

The following table presents the additional attributes and/or associations for **ProcessRef**:

**Table 45. ProcessRef Attributes and/or Associations**

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>startTrigger : LifecycleEvent [0..1]</td>
<td>This identifies the Event that triggers the start of the overall Knowledge Package.</td>
</tr>
</tbody>
</table>

### 13.4.9 SharedDataModelRef

The figure below displays the **SharedDataModelRef** metamodel.

![The SharedDataModelRef Metamodel](image)

**Figure 59: The SharedDataModelRef Metamodel**
Generalizations

The `SharedDataModelRef` element inherits the attributes and/or associations of:

- `BPMPlusDiagramElement` (see the section entitled “BPMPlusDiagramElement” for more information).

In addition, the `SharedDataModelRef` element inherits the attributes and/or associations of:

- `SDMNElementRef` (see the section entitled “SDMNElementRef” for more information).

Further, the `SDMNElementRef` element inherits the attributes and/or associations of:

- `BPMPlusModelElement` (see the section entitled “BPMPlusModelElement” for more information).

Further, the `BPMPlusModelElement` element inherits the attributes and/or associations of:

- `KnowledgeElement` (see the section entitled “KnowledgeElement” for more information).

Further, the `KnowledgeElement` element inherits the attributes and/or associations of:

- `ElementType` (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The `SharedDataModelRef` element does not have any additional attributes and/or associations.

13.4.10 UndefinedBehavior

The `UndefinedBehavior` element is used as a placeholder for one of the other behavioral elements. It is mainly used early in the development of a `BKPMNDefinitions`. A specific behavior is known to be part of the pathway, but it is not clear at that point whether the behavior is a Process or a Case. This allows the inclusion of the behavior in the overall context before the modeler is ready to set its type. It is contained within a `BKPMNDefinitions`.

Generalizations

The `UndefinedBehavior` element inherits the attributes and/or associations of:

- `ElementType` (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

In addition, the `UndefinedBehavior` element inherits the attributes and/or associations of:

- `BPMPlusDiagramElement` (see the section entitled “BPMPlusDiagramElement” for more information).

Properties

The `UndefinedBehavior` element does not have any additional attributes and/or associations.
13.4.11 PedigreeAndProvenanceRef

Generalizations

The PedigreeAndProvenanceRef element inherits the attributes and/or associations of:

- BPMPlusDiagramElement (see the section entitled “BPMPlusDiagramElement” for more information).

In addition, the PedigreeAndProvenanceRef element inherits the attributes and/or associations of:

- PPMNElementRef (see the section entitled “PPMNElementRef” for more information).

Further, the PPMNElementRef element inherits the attributes and/or associations of:

- BPMPlusModelElement (see the section entitled “BPMPlusModelElement” for more information).

Further, the BPMPlusModelElement element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The PedigreeAndProvenanceRef element does not have any additional attributes and/or associations.

13.4.12 Order Set Handlers and Questionnaire Handlers

Order Set Handlers and Questionnaire Handlers can be considered to be “mini” BKPMNPackages. Most of their
content is shared data, but there are Process, Case and/or and Decision elements. Order Set Handlers and Questionnaire Handlers can be developed independently or part of a larger BKPMNPackage. These mini BKPMN Packages would have their own life-cycle and metadata. More than one BKPMNPackage may use the same Order Set Handler or Questionnaire Handler.

13.4.12.1 OrderSetHandlerRef

It should be noted that Order Sets are specific to the healthcare domain. Thus, they are appropriate for modeling Shareable Clinical Pathways, but are not appropriate to be defined as an element of a generic BKPMNPackage. We will have to find the appropriate metamodel mechanisms to define generic BKPMN Packages that will allow domain-specific extensions, such as Order Sets. This includes the notation of these elements in a BKPMNPackage.

Generalizations

The OrderSetHandlerRef element inherits the attributes and/or associations of:

- BKPMNPackageRef (see the section entitled “BKPMNPackageRef” for more information).

Further, the BKPMNPackageRef element inherits the attributes and/or associations of:

- BKPMNElementRef (see the section entitled “BKPMNElementRef” for more information).

Further, the BKPMNElementRef element inherits the attributes and/or associations of:

- BPMPlusModelElement (see the section entitled “BPMPlusModelElement” for more information).

Further, the BPMPlusModelElement element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “KnowledgeElement” for more information).

Further, the KnowledgeElement element inherits the attributes and/or associations of:

- ElementType (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

In addition, the SCEElement element inherits the attributes and/or associations of:

- BPMPlusDiagramElement (see the section entitled “BPMPlusDiagramElement” for more information).

Properties

The OrderSetHandlerRef element does not have any additional attributes and/or associations.

13.4.12.2 QuestionnaireHandlerRef

Questionnaire Handlers, on the other hand, are key to healthcare but could also be used for other domains and, thus, can be included as part of a BKPMNPackage definition.

Generalizations

The QuestionnaireHandlerRef element inherits the attributes and/or associations of:

- BKPMNPackageRef (see the section entitled “BKPMNPackageRef” for more information).

Further, the BKPMNPackageRef element inherits the attributes and/or associations of:

- BKPMNElementRef (see the section entitled “BKPMNElementRef” for more information).

Further, the BKPMNElementRef element inherits the attributes and/or associations of:

- BPMPlusModelElement (see the section entitled “BPMPlusModelElement” for more information).

Further, the BPMPlusModelElement element inherits the attributes and/or associations of:

- KnowledgeElement (see the section entitled “KnowledgeElement” for more information).
Further, the KnowledgeElement element inherits the attributes and/or associations of:

- **ElementType** (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

In addition, the SCEElement element inherits the attributes and/or associations of:

- **BPMPlusDiagramElement** (see the section entitled “BPMPlusDiagramElement” for more information).

Properties

The QuestionnaireHandlerRef element does not have any additional attributes and/or associations.

13.4.13 Connectors

The KnowledgeModel currently has only two types of ModelElementConnectors – the ModelLink and the ReferenceLink. This does not include the Association, which is a ModelArtifact. Early drafts of this diagram contained other types of connectors and it is possible that additional connectors will be added to the diagram in the future.

The figure below displays the Connectors metamodel.

![Connectors Metamodel](image)

**Figure 61: The Connectors Metamodel**

13.4.13.1 ModelElementConnector

These types of connector defines a specific model link between elements of different modeling languages. For example, a CMMN ProcessTask links to a specific BPMN Process.

Generalizations

The ModelElementConnector element inherits the attributes and/or associations of:
- `ElementRelationshipType` (see the SCE specification for more information [OMG doc number bmi-2021-12-09]).

## Properties

The following table presents the additional attributes and/or associations for `ModelElementConnector`:

### Table 46. `ModelElementConnector` Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sourceRef</code> : BPMPlusModelElement [1]</td>
<td>This identifies the KnowledgeModel element that is at the source end of the <code>ModelElementConnector</code>. This attribute redefines the <code>sourceRef</code> attribute of SCE <code>ElementRelationshipType</code>.</td>
</tr>
<tr>
<td><code>targetRef</code> : BPMPlusModelElement [1]</td>
<td>This identifies the KnowledgeModel element that is at the target end of the <code>ModelElementConnector</code>. The arrowhead will be at the end of the <code>ModelElementConnector</code> attached to this element. This attribute redefines the <code>targetRef</code> attribute of SCE <code>ElementRelationshipType</code>.</td>
</tr>
</tbody>
</table>

### 13.4.13.2 ModelLink

This type of `ModelElementConnector` defines a specific model link between elements of different modeling languages. For example, a CMMN ProcessTask links to a specific BPMN Process. These kinds of connections can be validated by inspecting the source model file to find the specific link.

A `ModelLink` is used to show a relationship (calling/referencing) relationship between behavioral model elements of a KnowledgeModel. It is contained within a `BKPMNDefinitions`. There are 6 different types of relationships:

- **ProcessRef to ProcessRef**
  - This is realized through a BPMN Call Activity referencing another Process.

- **ProcessRef to CaseRef**
  - This is realized through a BPMN Case Task (extension) referencing a CMMN Case.

- **ProcessRef to DecisionServiceRef**
  - This is realized through a BPMN Business Rule Task referencing a DMN Decision Service.

- **CaseRef to ProcessRef**
  - This can be validated if:
    - The Case referenced by the CaseRef can be found within a CMMN file identified in the Manifest. realized through a CMMN Process Task referencing a BPMN Task.

- **CaseRef to CaseRef**
  - This is realized through a CMMN Case Task referencing another Case.

- **CaseRef to DecisionServiceRef**
  - This is realized through a CMMN Decision Task referencing a DMN Decision Service.

*Note that a BKPMN model does not directly contain the information required to identify these relationships (other than the Connectors). That information is contained within the Tasks of a BPMN.*
Process or a CMMN Case. A property of those Tasks identifies the target of the relationship (within a different BPMN Process, CMMN Case, or DMN model).

The addition of a ModelLink within a KnowledgeModel can be done in two ways:

- A modeler can use a tool mechanism to draw the connector between two behavioral model elements (e.g., between a BPMN Process and a CMMN Case).
- A modeling tool could inspect the BPMN, CMMN, and DMN files referenced by the KnowledgeModel elements (or perhaps the Manifest) and derive the relationships and add the connectors to the diagram.
- A ModelLink is line that SHALL be drawn with a thick dashed single line with a filled arrowhead (see figure below).
  - The use of text, color, size, and lines for an ModelLink SHALL follow the rules defined in the section entitled “Use of Text, Color, Size, and Lines in a Diagram” above.

Figure 62: A ModelLink

Generalizations

The ModelLink element inherits the attributes and/or associations of:

- ModelElementConnector (see the section entitled “ModelElementConnector” for more information).

Further, the Connector element inherits the attributes and/or associations of:

- SCEElement (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Further, the SCEElement element inherits the attributes and/or associations of:

- SCERootElement (see the SCE Specification for more information).

Constraints

A ModelLink is a type of SCE ElementRelationshipType, through the ModelElementConnector element, but is distinguished with this constraint:

- The instance for the element’s RelationshipKind SHALL be named “Dependency”.

Properties

The ModelLink element does not have any additional attributes and/or associations.

13.4.13.3 Connector Rules and Restrictions

Connectors are only allowed between Behavioral Elements, but there are additional restrictions for the connectors.

- A ModelLink MAY connect the following behavioral elements (in these directions):
  - A ProcessRef to another ProcessRef.
  - A ProcessRef to a CaseRef.
  - A ProcessRef to a DecisionServiceRef.
  - A CaseRef to another CaseRef.
  - A CaseRef to a ProcessRef.
  - A CaseRef to a DecisionServiceRef.

- A ModelLink MAY NOT connect other combinations of BPMPlusModelElementRef that are not listed above.
13.4.13.4 ReferenceLink

This type of ModelElementConnector defines a specific model link between elements of different modeling languages. For example, a CMMN ProcessTask links to a specific CQL function. These kinds of connections cannot be validated by inspecting the source model file to find the specific link.

There are 5 different types of known relationships (additional types can be created through adding operational model elements that have not been specified by this version of BKPMN):

- ProcessRef to BKPMNPackageRef, including OrderSets and Questionnaires.
- CaseRef to BKPMNPackageRef, including OrderSets and Questionnaires.
- CollaborationRef to BKPMNPackageRef
- BKPMNPackageRef, including OrderSets and Questionnaires, to ProcessRef
- BKPMNPackageRef, including OrderSets and Questionnaires, to CaseRef

A ReferenceLink is used to show a relationship (calling/referencing) relationship between behavioral model elements of a KnowledgeModel. It is contained within a BKPMNDefinitions.

- A ReferenceLink is line that SHALL be drawn with a thick dashed and dotted single line with a unfilled arrowhead (see figure below).
  o The use of text, color, size, and lines for a ReferenceLink SHALL follow the rules defined in the section entitled “Use of Text, Color, Size, and Lines in a Diagram” above.

Figure 63: A ReferenceLink

Generalizations

The ReferenceLink element inherits the attributes and/or associations of:

- ModelElementConnector (see the section entitled “ModelElementConnector” for more information).

Further, the Connector element inherits the attributes and/or associations of:

- SCEElement (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Further, the SCEElement element inherits the attributes and/or associations of:

- SCERootElement (see the SCE Specification for more information).

Constraints

A ReferenceLink is a type of SCE ElementRelationshipType, through the ModelElementConnector element, but is distinguished with this constraint:

- The instance for the element’s RelationshipKind SHALL be named “Reference”.

Properties

The ReferenceLink element does not have any additional attributes and/or associations.
13.4.14 Model Artifacts

BKPMN provides modelers with the capability of showing additional information about a KnowledgeModel that is not directly related to the KnowledgeModel elements through the capability provided by the ModelArtifact elements that are defined in the SCE specification. BKPMN utilizes the three standard SCE ModelArtifacts: Associations, Groups, and Text Annotations. BKPMN extends the base SCE Group element (see below).

BKPMN does not extend the capabilities of the Association and Text Annotation ModelArtifacts but uses them as-is from the SCE specification. The Group ModelArtifact has been extended for BKPMN (see the section entitled “Group”, below).

The following figure shows how the SCE ModelArtifact is included within BKPMN. ModelArtifacts are contained within a KnowledgeModel.

![Figure 64: The Use of SCE Model Artifacts in BKPMN](image)

A modeler or modeling tool MAY extend a KnowledgeModel and add new types of ModelArtifacts. Any new ModelArtifacts SHALL follow the Connector connection rules (listed below). Associations can be used to link ModelArtifacts to other diagram elements.

**Notation**

Full details of ModelArtifacts are available in the SCE specification, but the notation of the elements is provided here for convenience.

The table below displays the graphical elements of SCE’s ModelArtifacts:
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Notation</th>
</tr>
</thead>
</table>
| Association      | An Association is used to associate Diagram Artifacts (often Text Annotations) or diagram elements to other diagram elements. The connection only specifies that there is some relationship between the two elements, but no model semantics are implied.  
An Association is line that is drawn with a dotted single line. An angle $30^\circ$ arrowhead may optionally be added to either end of the line. |          |
| Group            | The Group object is a Diagram Artifact that provides a visual mechanism to group elements of a diagram informally. Groups are often used to highlight certain sections of a diagram. The highlighted (grouped) section of the diagram can be separated for reporting and analysis purposes.  
BKPMN also allows a ParticipantRef to be associated with a Group.  
A Group is a rounded corner rectangle that is drawn with a solid dashed and dotted line (see figure to the right). | ![Group Diagram] |
| Text Annotation  | Text Annotations are a mechanism for a modeler to provide additional information for the reader of a KnowledgeModel. An Association may be used to connect user-defined text (a Text Annotation) with a diagram element.  
A Text Annotation is an open rectangle that is drawn with a solid single line (see figure to the right). | ![Text Annotation Diagram] |

### 13.4.14.1 Group (BKPMN)

The Group object is a ModelArtifact that provides a visual mechanism to informally group elements of a diagram. Groups are often used to highlight certain sections of a diagram without adding additional constraints or semantics. The highlighted (grouped) section of the diagram can be separated for reporting and analysis purposes. BKPMN extends the SCE Group element by adding an association to a ParticipantRef. This allows provides the capability for Groups in a KnowledgeModel to separate behavioral elements into groups related to a particular participant of the BKPMNDefinitions behaviors (see the figure in the section entitled “BPM+ Knowledge Packages,” above). It is contained within a BKPMNDefinitions.

The following figure shows the Group metamodel.
Generalizations

The Group element inherits the attributes and/or associations of:

- Group (see the SCE Specification for more information [OMG doc number bmi-2021-12-09]).

Properties

The following table presents the additional attributes and/or associations for Group:

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>participantRef : ParticipantRef [0..1]</td>
<td>This identifies the Participant within the BKPMNDefinitions is associated with the Group. The BPMPlusModelRef elements that are contained within the graphical boundaries of the Group should also be associated with the same ParticipantRef.</td>
</tr>
</tbody>
</table>

13.4.14.2 Model Artifact Connection Rules

See the section entitled “Error! Reference source not found.” above for Connector connection rules.

- A DiagramArtifact SHALL NOT be a target for a Model Link or a Reference Link.
- A DiagramArtifact SHALL NOT be a source for a Model Link or a Reference Link.

14 BKPMN Library

A Library is included in BKPMN to provide standard instances that should be implemented by tools supporting BKPMN. Currently, BKPMN defines the instances for 10 sub-packages (see below).

The following figure presents the 10 instances for the BKPMNVocabulary element provided by the BKPMN Library:
Figure 66: The BKPMN Library Instance Model

The following table presents a description for the included instances for BKPMNVocabulary:

Table 48. BKPMNVocabulary Instances

<table>
<thead>
<tr>
<th>Instance</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContributorKinds</td>
<td>This BKPMNVocabulary instance does not have any predefined instances of SemanticReference terms. These can be added by a modeler or tooling environment. Example ContributorKind instances could include: author; editor; endorser; reviewer; subject-matter-expert.</td>
</tr>
<tr>
<td>DocumentKinds</td>
<td>This BKPMNVocabulary instance has five predefined instances of SemanticReference terms. See section below. Additional terms can be added by a modeler or tooling environment.</td>
</tr>
<tr>
<td>ImpactKinds</td>
<td>This BKPMNVocabulary instance does not have any predefined instances of SemanticReference terms. These can be added by a modeler or tooling environment. Example ImpactKind instances could include: negative; neutral; positive; and unknown.</td>
</tr>
<tr>
<td>ImplementationKinds</td>
<td>This BKPMNVocabulary instance has seven predefined instances of SemanticReference terms. See section below. Additional terms can be added by a modeler or tooling environment.</td>
</tr>
<tr>
<td>MeasureKinds</td>
<td>This BKPMNVocabulary instance does not have any predefined instances of SemanticReference terms. These can be added by a modeler or tooling environment. Example MeasureKind instances could include: quality; process; perception; performance; and input.</td>
</tr>
</tbody>
</table>
**NarrativeKinds**

This BKPMNVocabulary instance does not have any predefined instances of SemanticReference terms. These can be added by a modeler or tooling environment. Example NarrativeKind instances could include: background; discussion; scope.

**Purposes**

This BKPMNVocabulary instance does not have any predefined instances of SemanticReference terms. These can be added by a modeler or tooling environment. Example Purpose instances could include: assessment; education.

**RecommendationStatuses**

This BKPMNVocabulary instance does not have any predefined instances of SemanticReference terms. These can be added by a modeler or tooling environment. Example RecommendationStatus instances could include: amended; deleted; new-added; new-replaced; not-changed.

**RecommendationStrengths**

This BKPMNVocabulary instance does not have any predefined instances of SemanticReference terms. These can be added by a modeler or tooling environment. Example RecommendationStrength instances could include: strong-against; strong-for; weak-against; weak-for.

**RiskKinds**

This BKPMNVocabulary instance does not have any predefined instances of SemanticReference terms. These can be added by a modeler or tooling environment. Example RiskKind instances could include: fraction-of-incidences; hazard-ratio; increases-in-incidences; and relative.

### 14.1 DocumentKinds

The following figure presents the instances for the DocumentKind element that are terms for the instance (DocumentKinds) of the BKPMNVocabulary element:

![DocumentKinds Instance Model](image)

**Figure 67: The DocumentKinds Instance Model**

The following table presents a description for the included instances for DocumentKind:
Table 49. ItemKind Literals

<table>
<thead>
<tr>
<th>Literal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BKPMNPackageSummary</td>
<td>This SemanticReference instance is used for the DocumentKind of the BKPMNPackageSummary element.</td>
</tr>
<tr>
<td>InitializationData</td>
<td>This SemanticReference instance is used for the DocumentKind of the InitializationDataRef element.</td>
</tr>
<tr>
<td>Narrative</td>
<td>This SemanticReference instance is used for the DocumentKind of the NarrativeRef element.</td>
</tr>
<tr>
<td>PolicyDocument</td>
<td>This SemanticReference instance is used for the DocumentKind of the PolicyDocumentRef element.</td>
</tr>
<tr>
<td>TestCase</td>
<td>This SemanticReference instance is used for the DocumentKind of the TestCaseRef element.</td>
</tr>
</tbody>
</table>

14.2 ImplementationKinds

The following figure presents the instances for the ImplementationKind element that are terms for the instance (ImplementationKinds) of the BKPMN VOCABULARY element:

![ImplementationKinds Instance Model](image)

Figure 68: The ImplementationKinds Instance Model

The following table presents a description for the included instances for ImplementationKind:
### Table 50. ItemKind Literals

<table>
<thead>
<tr>
<th>Literal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BKPMN</td>
<td>This SemanticReference instance is used for the ImplementationKind of the BKPMNDocumentRef element.</td>
</tr>
<tr>
<td>BPMN</td>
<td>This SemanticReference instance is used for the ImplementationKind of the BKPMNDocumentRef element.</td>
</tr>
<tr>
<td>CMMN</td>
<td>This SemanticReference instance is used for the ImplementationKind of the CMMNDocumentRef element.</td>
</tr>
<tr>
<td>DMN</td>
<td>This SemanticReference instance is used for the ImplementationKind of the DMNDocumentRef element.</td>
</tr>
<tr>
<td>MDMI</td>
<td>This SemanticReference instance is used for the ImplementationKind of the MDMIDocumentRef element.</td>
</tr>
<tr>
<td>PPMN</td>
<td>This SemanticReference instance is used for the ImplementationKind of the PPMNDocumentRef element.</td>
</tr>
<tr>
<td>SDMN</td>
<td>This SemanticReference instance is used for the ImplementationKind of the SDNDocumentRef element.</td>
</tr>
</tbody>
</table>

## 15 BPM+ Models Harmonization

A KnowledgeModel (see the section entitled “KnowledgeModel”, above) defines relationships between core modeling elements of the BPM+ languages (BPMN, CMMN, and DMN) that are used for a BPM+ Knowledge Package. For example, a CMMN Decision Task can be associated to a DMN Decision Service through the ModelLink element. If the CMMN Case was configured correctly and the target DMN Decision Service is present in the DMN models referenced by the Manifest, then a BKPMN tool can validate the ModelLink between the two objects in the KnowledgeModel. All ModelLinks in a KnowledgeModel should be validated before a BPM+ Knowledge Package is considered complete.

However, BPMN was developed before CMMN and DMN, and thus, does not have explicit mechanism to link to CMMN or DMN elements. Thus, BPMN requires additional capabilities to facilitate this validation (see next section). This is the main purpose of the BPM+ Model Harmonization for BKPMN.

### 15.1 BPMN Harmonization

The BPMN harmonization overlays onto BPMN the infrastructure to enable BKPMN to validate ModelLinks from a ProcessRef to a CaseRef or DecisionServiceRef.

The following figure shows the metamodel for harmonizing BPMN within a BPM+ Knowledge Package.
15.1.1 BaseElement

Generalizations

The BaseElement element inherits the attributes and/or associations of:

- BPMN BaseElement (see the BPMN specification for more information).

Properties

The following table presents the additional attributes and/or associations for BaseElement:

Table 51. BaseElement Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>semanticReference : (SCE) SemanticReference [0..*]</td>
<td>A concrete BaseElement can include multiple SCE SemanticReference elements.</td>
</tr>
</tbody>
</table>

15.1.2 BusinessRuleTask

Generalizations

The BusinessRuleTask element inherits the attributes and/or associations of:

- BPMN BusinessRuleTask (see the BPMN specification for more information).

Properties
The following table presents the additional attributes and/or associations for BusinessRuleTask:

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>decisionRef : Decision [0..1]</td>
<td>A reference to a Decision (see below). If decisionRef is not specified, then decisionRefExpression MUST be specified. Only one of the attributes, decisionRefExpression or decisionRef MUST be specified.</td>
</tr>
<tr>
<td>decisionRefExpression : (BPMN) Expression [0..1]</td>
<td>This association defines the details of how a BusinessRuleTask, which is in a Process that is contained in a BPMN file that is referenced in the Manifest, is connected to a DMN DecisionService, which is contained in a DMN file that is referenced in the Manifest. Note that the link may be to a DMN Decision.</td>
</tr>
</tbody>
</table>

15.1.3 CaseTask

Generalizations

The CaseTask element inherits the attributes and/or associations of:

- BPMN Task (see the BPMN specification for more information).

Properties

The following table presents the additional attributes and/or associations for CaseTask:

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>caseRef : Case [0..1]</td>
<td>A reference to a Case (see below). If caseRef is not specified, then caseRefExpression MUST be specified. Only one of the attributes, caseRefExpression or caseRef MUST be specified.</td>
</tr>
<tr>
<td>caseRefExpression: (BPMN) Expression [0..1]</td>
<td>If caseRefExpression is specified, it is assumed that the expression evaluates to a QName which is a valid QName of an existing Process. The process referred to by this QName MUST have compatible Input and Output parameters. The processRefExpression can be used to determine the concrete Process to be invoked by the ProcessTask at runtime. If that attribute is not specified, then processRef MUST refer to a valid Process. Only one of the attributes, processRefExpression or processRef MUST be specified.</td>
</tr>
<tr>
<td>implementation : String [0..1]</td>
<td>This attribute specifies the technology that will be used to link to a CMMN Case Model.</td>
</tr>
</tbody>
</table>
15.1.4 Case

A Case in this context is an abstraction of Cases as they are specified in various Case modeling specifications. By default, a version of the CMMN specification is assumed.

**Generalizations**

The Case element inherits the attributes and/or associations of:

- BPMN RootElement (see the BPMN specification for more information).

**Properties**

The following table presents the additional attributes and/or associations for Case:

**Table 54. Case Attributes and/or Associations**

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>externalRef : QName [0..1]</td>
<td>The concrete Case to be used.</td>
</tr>
<tr>
<td>implementationType : URI</td>
<td>The implementation type of the Case. It MUST be provided in URI format.</td>
</tr>
<tr>
<td>input : CaseParameter [0..*]</td>
<td>Zero or more inputs of the Case.</td>
</tr>
<tr>
<td>output : CaseParameter [0..*]</td>
<td>Zero or more outputs of the Case.</td>
</tr>
</tbody>
</table>

15.1.5 CaseParameter

**Generalizations**

The CaseParameter element inherits the attributes and/or associations of:

- BPMN RootElement (see the BPMN specification for more information).

**Properties**

The following table presents the additional attributes and/or associations for CaseParameter:

**Table 55. CaseParameter Attributes and/or Associations**

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name : String [0..1]</td>
<td>The name of the CaseParameter.</td>
</tr>
</tbody>
</table>

15.1.6 Decision

A Decision in this context is an abstraction of Decisions or Decision Services as they are specified in various Decision modeling specifications. By default, a version of the DMN specification is assumed.
**Generalizations**

The *Decision* element inherits the attributes and/or associations of:

- **BPMN RootElement** (see the BPMN specification for more information).

**Properties**

The following table presents the additional attributes and/or associations for *Decision*:

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>externalRef : QName [0..1]</td>
<td>The concrete <em>Decision</em> to be used.</td>
</tr>
<tr>
<td>implementationType : URI</td>
<td>The implementation type of the <em>Decision</em>. It MUST be provided in URI format.</td>
</tr>
<tr>
<td>input : DecisionParameter [0..*]</td>
<td>Zero or more inputs of the <em>Decision</em>.</td>
</tr>
<tr>
<td>name : String [1]</td>
<td>The name of the <em>Decision</em>.</td>
</tr>
<tr>
<td>output : CaseParameter [0..*]</td>
<td>Zero or more outputs of the <em>Decision</em>.</td>
</tr>
</tbody>
</table>

### 15.1.7 DecisionParameter

**Generalizations**

The *DecisionParameter* element inherits the attributes and/or associations of:

- **BPMN RootElement** (see the BPMN specification for more information).

**Properties**

The following table presents the additional attributes and/or associations for *DecisionParameter*:

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name : String [0..1]</td>
<td>The name of the <em>DecisionParameter</em>.</td>
</tr>
</tbody>
</table>

### 15.2 CMMN Harmonization

The harmonization of CMMN to a BPM+ Knowledge Package is limited to the inclusion of an SCE *SemanticReference* to any CMMN element.

The following figure shows the metamodel for harmonizing CMMN within a BPM+ Knowledge Package.
15.2.1 CMMNElement

Generalizations

The CMMNElement element inherits the attributes and/or associations of:

- CMMN CMMNElement (see the CMMN specification for more information).

Properties

The following table presents the additional attributes and/or associations for CMMNElement:

Table 58. CMMNElement Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>semanticReference : (SCE) SemanticReference [0..*]</td>
<td>A concrete CMMNElement can include multiple SCE SemanticReference elements.</td>
</tr>
</tbody>
</table>

15.3 DMN Harmonization

The harmonization of DMN to a BPM+ Knowledge Package is limited to the inclusion of an SCE SemanticReference to any DMN element.

The following figure shows the metamodel for harmonizing DMN within a BPM+ Knowledge Package.
15.3.1 DMNElement

Generalizations

The DMNElement element inherits the attributes and/or associations of:
- DMN DMNElement (see the DMN specification for more information).

Properties

The following table presents the additional attributes and/or associations for DMNElement:

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>semanticReference : (SCE) SemanticReference [0..*]</td>
<td>A concrete DMNElement can include multiple SCE SemanticReference elements.</td>
</tr>
</tbody>
</table>

16 BKPMN Examples (Informative)

16.1 Use Case: Hello Patient

The BPM+ Health community has been defining Shareable Clinical Pathways by using the current BPM+ standards to define formal and executable versions of current clinical guidelines (e.g., for hypertension, chronic kidney disease, etc.). Current clinical guidelines are usually found in printed or PDF documents and they contain vague and often confusing semantics leading to a great variability in how the guidelines are understood and performed.

This section describes a simple use case that was developed by the BPM+ Health community. At that time there was no concepts of BPM+ Knowledge Packages or Shared Data. The work on this and other use cases was instrumental.
in identifying the need and requirements for a BPM+ Knowledge Package and for Shared Data.

### 16.1.1 Organizing BPM+ Models (A BPM+ Knowledge Package)

The use case defined the Processes, Cases, and Decision Services that are involved in managing a visit to a Doctor’s office. Note that these models were intended to be illustrative rather than an official, comprehensive healthcare guideline.

The following table lists the major BPM+ model elements that made up the use case.

<table>
<thead>
<tr>
<th>Table 60. List of BPM+ Models for the Hello Patient Use Case</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cases</strong></td>
</tr>
<tr>
<td>2. Perform Examination</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

A larger use case for “Antenatal Care” was developed and contained more models than listed above. For that use case there were 9 Cases, 15 Decision Services, and 28 Processes.

Reviewing one of the models listed in the table above does not provide the overall scope and context of the set of models in the use case. While it may be possible to trace through the connections between the BPM+ models, that tracing still does provide the proper context.

This lack of perspective resulted in a new type of diagram included with the use case. It is referred to as a Knowledge Diagram. The diagram provides graphical representations of the BPM+ models and draws connectors to represent how the models can be traced through their connections. The following figure displays the Knowledge Diagram for the Hello Patient use case. Note that all the items listed in the table above have diagram elements associated with them. There are different notations for Processes, Cases, and Decision Services.
Figure 72: Example of a BPM+ Knowledge Diagram

Note that the development of the Knowledge Diagram for the use cases was an indication that something else was needed to fully document the contexts of a set of BPM+ created for a specific topic. This and other factors led to the requirements for a BPM+ Knowledge Package.

16.1.2 Organizing BPM+ Data Elements (A Shared Data Model)

Several elements in BPM+ Models are intended to store or convey data required for the execution of those Models. **BPMN** has Data Objects, Data Inputs, Data Outputs, Data Stores, and Properties. **CMMN** has Case File Items. **DMN** has Information Items that are used for Data Inputs and Decisions. The Hello Patient use case employed many of these types of data elements within its BPM+ models. The following table lists those data elements used within the set of BPM+ models for the Hello Patient use case.
Table 61. List of Data Elements used by the BPM+ Models in the Hello Patient Use Case

<table>
<thead>
<tr>
<th>Case Data</th>
<th>Decision Service Data</th>
<th>Process Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. Encounter</td>
<td>4. BMI Category</td>
<td>4. BMI Category</td>
</tr>
<tr>
<td>5. Exam Data</td>
<td>5. Demographics</td>
<td>5. Demographics</td>
</tr>
<tr>
<td>15. Weight Counseling Referral</td>
<td>15. Treatment Choice</td>
<td>15. Referral</td>
</tr>
<tr>
<td>17. Weight Counseling Referral</td>
<td>17. Weight Counseling Referral Choice</td>
<td>17. Treatment Choice</td>
</tr>
</tbody>
</table>

Note that the data elements listed in **bold** in the table are those that appear in all three types of BPM+ models. The other data elements appear in at least two of the model types.

The set of data elements listed in the above table reflect those data elements that are necessary for only the context of this use case (Hello Patient). They do not represent all the data elements that a Doctor’s Office may require for all of its operations – let alone all the data elements required for the healthcare domain. The use case only specified the data elements required for its particular situation. Hence, we refer to sets of data elements used in such a context as “Shared Data”.

Since the use case employed all three different types of BPM+ models (Process, Case, and Decision Service), the Shared Data of the use case is shared and distributed across the three types of models. While there are some technical differences between how data is structured and used across the BPM+ specifications, at the logical level, they all play the same role within the respective languages. This is evident when a specific conceptual data element (e.g., “Vital Signs and Measures”) can be included in all three BPM+ modeling languages (see figure below). That is, the same data element (and its values during runtime) can be, for example, passed from a CMMN Case to a BPMN Process and then used in a DMN Decision.
Without a Shared Data Model tool, the same data element has to be defined separately in the tools dedicated to each modeling language. There are no standard mechanisms for sharing data elements across the three types BPM+ models.

If there are a lot of data elements that are shared between the models of a BPM+ Knowledge Package, the development and maintenance burden for synchronizing the properties of the data elements will problematic. All of the Hello Patient use data elements were used in at least two types of models. Each time any of the data elements were modified, which can happen multiple times during the BPM+ Knowledge Package development cycle, there would be one or more modifications in the other types of BPM+ models. It would be up to the modeler to ensure that the modifications were made and were consistent.

This maintenance burden was the driver for defining a Shared Data Model, which would be a library of data elements that would readily be available for synchronization with the other BPM+ Models. That is, the Data Items of the Shared Data Model should share the same characteristics as the data elements of the three BPM+ data elements. Further, the modeling experience should be very similar across all four models to ease burdens on the modeler.

The Shared Data Model would provide an environment where data elements can be defined and modified in a single location and the changes could be distributed to the other BPM+ models without additional work and vigilance by the modeler.

17 Exchange Formats

17.1 Interchanging Incomplete Models

In practice, it is common for models to be interchanged before they are complete. This occurs frequently when doing iterative modeling, where one user (such as a subject matter expert or business person) first defines a high-level model, and then passes it on to another user to be completed and refined.

Such “incomplete” models are ones in which all of the mandatory attributes have not yet been filled in, or the cardinality lowerbound of attributes and associations has not been satisfied.

XMI allows for the interchange of such incomplete models. In BKPMN, we extend this capability to interchange of XML files based on the BKPMN XSD. In such XML files, implementers are expected to support this interchange by:

- Disregarding missing attributes that are marked as ‘required’ in the XSD.
- Reducing the lower bound of elements with ‘minOccurs’ greater than 0.
17.2 XSD

17.2.1 Document Structure

A domain-specific set of model elements is interchanged in one or more BKPMN files. The root element of each file SHALL be `<DMN:Definitions>`. The set of files SHALL be self-contained, i.e., all definitions that are used in a file SHALL be imported directly or indirectly using the `<DMN:Import>` element.

Each file SHALL declare a “namespace” that MAY differ between multiple files of one model.

BKPMN files MAY import non-BKPMN files (such as XSDs and PMMLs) if the contained elements use external definitions.

17.2.2 References within the BKPMN XSD

Many BKPMN elements that may need to be referenced contain IDs and within the BKPMN XSD, references to elements are expressed via these IDs. The XSD IDREF type is the traditional mechanism for referencing by IDs, however it can only reference an element within the same file. BKPMN elements of that inherit from SCE RootElement support referencing by ID, across files, by utilizing an href attribute whose value must be a valid URI reference [RFC 3986] where the path components may be absolute or relative, the reference has no query component, and the fragment consists of the value of the id of the referenced BKPMN element.

For example, consider the following Decision:

```xml
<decision name="Pre-Bureau Risk Category"
    id="prebureauriskDec01">…</decision>
```

When this Decision is referenced, e.g. by an InformationRequirement in a Decision that is defined in another file, the reference could take the following form:

```xml
<requiredDecision
    href="http://www.example.org/Definitions01.xml#prebureauriskDec01"/>
```

where “http://www.example.org/Definitions01.xml” is an URI reference to the XML document in which the “Pre-Bureau Risk Category” Decision is defined (e.g. the value of the locationURI attribute in the corresponding Import element), and “prebureauriskDec01” is the value of the id attribute for the Decision.

If the path component in the URI reference is relative, the base URI against which the relative reference is applied is determined as specified in [RFC 3986]. According to that specification, “if no base URI is embedded and the representation is not encapsulated within some other entity, then, if a URI was used to retrieve the representation, that URI shall be considered the base URI” ([RFC 3986], section 5.1.3). That is, if the reference is not in the scope of an xml:base attribute [XBASE], a value of the href attribute that contains only a fragment and no path component references a BKPMN element that is defined in the same instance of XML file as the referencing element. In the example below, assuming that the requiredDecision element is not in the scope of an xml:base attribute, the BKPMN element whose id is “prebureauriskDec01” must be defined in the same XML document:

```xml
<requiredDecision href="#prebureauriskDec01"/>
```

Attribute typeRef references ItemDefinitions and built-in types by name not ID. In order to support imported types, typeRef uses the namespace-qualified name syntax [qualifier].[local-name], where qualifier is specified by the name attribute of the Import element for the imported type. If the referenced type is not imported, the prefix SHALL be omitted.

18 BKPMN Diagram Interchange

18.1 Scope

This chapter specifies the meta-model and schema for BKPMN 1.0 Diagram Interchange (BKPMN DI). The
**BKPMN DI** is meant to facilitate the interchange of BKPMN diagrams between tools rather than being used for internal diagram representation by the tools. The simplest interchange approach to ensure the unambiguous rendering of a KPMN diagram was chosen for **BKPMN DI**. As such, **BKPMN DI** does not aim to preserve or interchange any “tool smarts” between the source and target tools (e.g., layout smarts, efficient styling, etc.). **BKPMN DI** does not ascertain that the BKPMN diagram is syntactically or semantically correct.

### 18.2 Measurement Unit

As per OMG DD, all coordinates and lengths defined by BKPMN DI are assumed to be in user units, except when specified otherwise. A user unit is a value in the user coordinate system, which initially (before any transformation is applied) aligns with the device’s coordinate system (for example, a pixel grid of a display). A user unit, therefore, represents a logical rather than physical measurement unit. Since some applications might specify a physical dimension for a diagram as well (mainly for printing purposes), a mapping from a user unit to a physical unit can be specified as a diagram’s resolution. Inch is chosen in this specification to avoid variability, but tools can easily convert from/to other preferred physical units. Resolution specifies how many user units fit within one physical unit (for example, a resolution of 300 specifies that 300 user units fit within 1 inch on the device).

### 18.3 Diagram Definition and Interchange

The **BKPMN DI** meta-model, similar to the BKPMN abstract syntax meta-model, is defined as a MOF-based meta-model. As such, its instances can be serialized and interchanged using XML. **BKPMN DI** is also defined by an XML schema. Thus its instances can also be serialized and interchanged using XML.

Both **BKPMN DI** meta-model and schema is layered upon the SCE DI (see the SCE 1.0 specification [OMG doc number bmi-2021-12-09]), which is harmonized with the OMG Diagram Definition (DD) standard version 1.1. The referenced DD contains two main parts: the Diagram Commons (DC) and the Diagram Interchange (DI). The DC defines common types like bounds and points, while the DI provides a framework for defining domain-specific diagram models. As a domain-specific DI, **BKPMN DI** defines a few new meta-model classes that derive from the abstract classes from SCE DI and DI.

The focus of **BKPMN DI** is the interchange of laid out shapes and edges that constitute a BKPMN diagram. Each shape and edge references a particular BKPMN model element. The referenced SDMN model elements are all part of the actual BKPMN model. As such, **BKPMN DI** is meant to only contain information that is neither present nor derivable, from the BKPMN model whenever possible. Simply put, to render a BKPMN diagram both the BKPMN DI instance(s) and the referenced BKPMN model are REQUIRED.

From the BKPMN DI perspective, a BKPMN diagram is a particular snapshot of a BKPMN model at a certain point in time. Multiple BKPMN diagrams can be exchanged referencing model elements from the same BKPMN model. Each diagram may provide an incomplete or partial depiction of the content of the BKPMN model. As described in Clause 13, a BKPMN model package consists of one or more files. Each file may contain any number of BKPMN diagrams. The exporting tool is free to decide how many diagrams are exported and the importing tool is free to decide if and how to present the contained diagrams to the user.

### 18.4 BKPMN Diagram Interchange Meta-Model

#### 18.4.1 How to read this Chapter

Clause 18.5.3 describes in detail the meta-model used to keep the layout and the look of BKPMN Diagrams. Clause 18.5.4 presents in tables a library of the BKPMN element depictions and an unambiguous resolution between a referenced BKPMN model element and its depiction.

#### 18.4.2 Overview

The BKPMN DI, which extends the SCE DI, is an instance of the OMG DI meta-model. The basic concept of BKPMN DI, as with diagram interchange in general, is that serializing a diagram [BKPMNDiagram] for interchange requires the specification of a collection of shapes [BKPMNShape] and edges [BKPMNEdge].
The **BKPMN DI** classes only define the visual properties used for depiction. All other properties that are REQUIRED for the unambiguous depiction of the **BKPMN** element are derived from the referenced **BKPMN** element [BKPMNElementRef].

**BKPMN** diagrams may be an incomplete or partial depiction of the content of the **BKPMN** model. Some **BKPMN** elements from a **BKPMN** model may not be present in any of the diagram instances being interchanged.

**BKPMN DI** does not directly provide for any containment concept. The **BKPMNDiagram** is an ordered collection of mixed **BKPMNShape**(s) and **BKPMNEdge**(s). The order of the **BKPMNShape**(s) and **BKPMNEdge**(s) inside a **BKPMNDiagram** determines their Z-order (i.e., what is in front of what). **BKPMNShape**(s) and **BKPMNEdge**(s) that are **BKPMNEdge**(s) MUST appear after them in the **BKPMNDiagram**. Thus, the exporting tool MUST order all **BKPMNShape**(s) and **BKPMNEdge**(s) such that the desired depiction can be rendered.

### 18.4.3 Measurement Unit

As per OMG DD, all coordinates and lengths defined by **BKPMN DI** are assumed to be in user units, except when specified otherwise. A user unit is a value in the user coordinate system, which initially (before any transformation is applied) aligns with the device’s coordinate system (for example, a pixel grid of a display). A user unit, therefore, represents a logical rather than physical measurement unit. Since some applications might specify a physical dimension for a diagram as well (mainly for printing purposes), a mapping from a user unit to a physical unit can be specified as a diagram’s resolution. Inch is chosen in this specification to avoid variability, but tools can easily convert from/to other preferred physical units. Resolution specifies how many user units fit within one physical unit (for example, a resolution of 300 specifies that 300 user units fit within 1 inch on the device).

### 18.4.4 Elements

The following sections define the elements necessary for exchanging the diagrams from an **BKPMN** modeling tool.

#### 18.4.4.1 BKPMNDI

The class **BKPMNDI** is a container for the shared **SCE SCEStyle** and all the **BKPMNDI Diagrams** defined in a **BKPMNDefinitions**.

The following figure shows the **BKPMNDI** metamodel.

![Figure 74: BKPMNDI](image)

**Generalizations**

The **BKPMNDI** element inherits the attributes and/or associations of:

- **SCE SCEDI** (see the **SCE** specification [OMG doc number bmi-2021-12-09] for more information).

**Properties**

The following table presents the additional attributes and/or associations for **BKPMNDI**:

<table>
<thead>
<tr>
<th>Attribute/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Table 62. BKPMNDI Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>diagram : BKPMNDIDiagram [0..*]</td>
<td>A list of BKPMNDIDiagrams. This redefines the diagram association inherited from the SCE SCEDI element.</td>
</tr>
</tbody>
</table>

18.4.4.2 BKPMNDIDiagram

The class BKPMNDIDiagram specializes SCE SCEDiagram. It is a kind of diagram that represents a depiction of all or part of a BKPMN model. It is contained within the BKPMNDI element (see above).

BKPMNDIDiagram is the container of BKPMNDIDiagramElement (BKPMNShape(s) and BKPMNEdge(s)). BKPMNDIDiagram cannot include other BKPMNDIDiagram.

A BKPMNDIDiagram can define an SCE SCEStyle locally and/or it can refer to a shared one defined in the BKPMDI. Properties defined in the local style overrides the one in the referenced shared style. That combined style (shared and local) is the default style for all the BKPMNDIDiagramElement contained in this BKPMNDIDiagram diagram.

The BKPMNDIDiagram class represents a two-dimensional surface with an origin of (0, 0) at the top left corner. This means that the x and y axes have increasing coordinates to the right and bottom. Only positive coordinates are allowed for diagram elements that are nested in a BKPMNDIDiagram.

The following figure shows the BKPMNDIDiagram metamodel.
Generalizations

The BKPMNDIDiagram element inherits the attributes and/or associations of:

- SCE SCEDIDiagram (see the SCE specification [OMG doc number bmi-2021-12-09] for more information).

Further, the SCEDIDiagram element inherits the attributes and/or associations of:

- DD Diagram (see the DD specification [OMG formal-15-06-01] for more information).

Properties

The following table presents the additional attributes and/or associations for BKPMNDIDiagram:

Table 63. BKPMNDIDiagram Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bkpmnDiagramRef : BKPMNDiagram [1]</td>
<td>The diagram that the DI is representing. In the current version, this will be a KnowledgeDiagram.</td>
</tr>
</tbody>
</table>
**18.4.4.3 BKPMNDIDiagramElement**

The **BKPMNDIDiagramElement** class is contained by the **BKPMNDiagram** and is the base class for **BKPMNShape** and **BKPMNEdge**.

**BKPMNDIDiagramElement** inherits its styling from its parent **BKPMNDiagram**. In addition, it can refer to one of the shared **SCE SCEStyle** defined in the **BKPMNDI** and/or it can define a local style. See the **SCE** specification [OMG doc number bmi-2021-12-09] for more details on styling.

**BKPMNDIDiagramElement** MAY also contain a **SCE SCELabel** when it has a visible text label. If no **SCE SCELabel** is defined, the **BKPMNDIDiagramElement** should be depicted without a label.

The following figure shows the **BKPMNDIDiagramElement** metamodel.

![BKPMNDI Diagram Element Metamodel](image)

**Figure 76: BKPMNDI Diagram Element**

**Generalizations**

The **BKPMNDIDiagramElement** element inherits the attributes and/or associations of:

- **SCE SCEDiagramElement** (see the **SCE** specification [OMG doc number bmi-2021-12-09] for more information).

Further, the **SCEDiagram** element inherits the attributes and/or associations of:

- **DD DiagramElement** (see the **DD** specification [OMG formal-15-06-01] for more information).

**Properties**

The **BKPMNDIDiagramElement** element does not have any additional attributes and/or associations.
18.4.4.4 BKPMNShape

The BKPMNShape class specializes SCE SCEShape and BKPMNDIDiagramElement. It is a kind of shape that depicts a concrete specialization of SCE SCEElement from the BKPMN model.

BKPMNShape represents a BKPMNPackageRef, CaseRef, ChoreographyRef, CollaborationRef, Development Method, DecisionServiceRef, QuestionnaireHandlerRef, OperationalModelRef, OrderSetHandlerRef, PedigreeAndProvenanceRef, ProcessRef, SharedDataModelRef, Undefined Behavior, Group, or a Text Annotation that is depicted on the diagram.

BKPMNShape has one additional property (isCollapsed) that are used to further specify the appearance of some shapes that cannot be deduced from the BKPMN model.

The following figure shows the BKPMNShape metamodel.

![BKPMNShape Metamodel](image)

**Figure 77:** BKPMNDI Shape

**Generalizations**

The BKPMNShape element inherits the attributes and/or associations of:

- BKPMNDIDiagramElement (see the section entitled “BKPMNDIDiagramElement” for more information).

Further, the BKPMNDIDiagramElement element inherits the attributes and/or associations of:

- DI::DiagramElement (see the DD specification (OMG formal-15-06-01) for more information).

In addition, the BKPMNShape element inherits the attributes and/or associations of:

- DI::Shape (see the DD specification (OMG formal-15-06-01) for more information).

**Properties**

The following table presents the additional attributes and/or associations for BKPMNShape:
Table 64. BKPMNShape Attributes and/or Associations

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isCollapsed : Boolean</td>
<td>default: false</td>
</tr>
</tbody>
</table>

18.4.4.5 BKPMNEdge

The BKPMNEdge class specializes DI::Edge and BKPMNDIDiagramElement. It is a kind of edge that can depict a relationship between two BKPMN model elements.

BKPMNEdge are used to depict RelationshipConnectors or Associations in the BKPMN model. Since BKPMNDIDiagramElement might be depicted more than once, sourceElementRef and targetElementRef attributes allow to determine to which depiction a BKPMNEdge is connected. When BKPMNEdge has a source, its sourceElementRef MUST refer to the BKPMNDIDiagramElement it starts from. That BKPMNDIDiagramElement MUST resolved to the concrete specialization of the SCE SCEElement that is the actual source of the BKPMNDIDiagramElement where it ends. When it has a target, its targetElementRef MUST refer to the BKPMNDIDiagramElement where it ends. That BKPMNDIDiagramElement MUST resolved to the concrete specialization of the SCE SCEElement that is the actual target of the edge.

The following figure shows the BKPMNEdge metamodel.

Figure 78: BKPMNDI Edge

Generalizations

The BKPMNEdge element inherits the attributes and/or associations of:

- BKPMNDIDiagramElement (see the section entitled “BKPMNDIDiagramElement” for more information).

Further, the BKPMNDIDiagramElement element inherits the attributes and/or associations of:
• *DI::DiagramElement* (see the *DD* specification (OMG formal-15-06-01) for more information).

In addition, the *BKPMNEdge* element inherits the attributes and/or associations of:

• *DI::Edge* (see the *DD* specification (OMG formal-15-06-01) for more information).

**Properties**

The *BKPMNEdge* element does not have any additional attributes and/or associations.

The following table presents the additional attributes and/or associations for *DI::Edge*:

<table>
<thead>
<tr>
<th>Property/Association</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>wayPoints</em>: <em>DC::Point [0..</em>]*</td>
<td>A list of points relative to the origin of its parent <em>BKPMNDIDiagram</em> that specifies the connected line segments of the edge. At least two (2) waypoints MUST be specified.</td>
</tr>
</tbody>
</table>

**18.4.5 Notation**

As a specification that contains notation, *BKPMN* specifies the depiction for *BKPMN* diagram elements, including *SCE DiagramArtifact* elements.

Serializing a *BKPMN* diagram for interchange requires the specification of a collection of *BKPMNShape*(s) and *BKPMNEdge*(s) in the *BKPMNDiagram* (see sections above). The *BKPMNShape*(s) and *BKPMNEdge*(s) attributes must be populated in such a way as to allow the unambiguous rendering of the *BKPMN* diagram by the receiving party. More specifically, the *BKPMNShape*(s) and *BKPMNEdge*(s) MUST reference *BKPMN* model elements. If no *SCEElement* is referenced or if the reference is invalid, it is expected that this shape or edge should not be depicted.

When rendering a *BKPMN* diagram, the correct depiction of a *BKPMNShape* or *BKPMNEdge* depends mainly on the referenced *BKPMN* model element and its particular attributes and/or references. The purpose of this clause is to: provide a library of the *BKPMN* element depictions, and to provide an unambiguous resolution between the referenced *BKPMN* model element [*SCEElement*] and their depiction. Depiction resolution tables are provided below for both *BKPMNShape* and *BKPMNEdge*.

**18.4.5.1 Labels**

Both *BKPMNShape* and *BKPMNEdge* may have labels (its name attribute) placed on the shape/edge, or above or below the shape/edge, in any direction or location, depending on the preference of the modeler or modeling tool vendor.

Labels are optional for *BKPMNShape* and *BKPMNEdge*. When there is a label, the position of the label is specified by the bounds of the *BKPMNLabel* of the *BKPMNShape* or *BKPMNEdge*. Simply put, label visibility is defined by the presence of the *BKPMNLabel* element.

The bounds of the *BKPMNLabel* are optional and always relative to the containing *BKPMNDiagram*’s origin point. The depiction resolution tables provided below exemplify default label positions if no bounds are provided for the *BKPMNLabel* (for *BKPMNShape* kinds and *BKPMNEdge* kinds (see sections above)).

When the *BKPMNLabel* is contained in a *BKPMNShape*, the text to display is the name of the *SCEElement*.

**18.4.5.2 BKPMNShape Resolution**

*BKPMNShape* can be used to represent a *Text Annotation* or a *Group*.
**Dipiction for BPMPlusDiagramElements**

When a BKPMNShape is used to depict a BPMPlusDiagramElement the actual shape is determined by: the referred BPMPlusDiagramElement.

The following table presents the depiction resolutions for DataItems:

**Table 66. Depiction Resolution of DataItems**

<table>
<thead>
<tr>
<th>BKPMNElement</th>
<th>BKPMNShape Attributes</th>
<th>Depiction</th>
</tr>
</thead>
<tbody>
<tr>
<td>BKPMNPackageRef</td>
<td></td>
<td><img src="image" alt="Label" /></td>
</tr>
<tr>
<td>CaseRef</td>
<td></td>
<td><img src="image" alt="Label" /></td>
</tr>
<tr>
<td>ChoreographyRef</td>
<td></td>
<td><img src="image" alt="Label" /></td>
</tr>
<tr>
<td>CollaborationRef</td>
<td></td>
<td><img src="image" alt="Label" /></td>
</tr>
<tr>
<td>DecisionServiceRef</td>
<td></td>
<td><img src="image" alt="Label" /></td>
</tr>
<tr>
<td>OperationalModelRef</td>
<td></td>
<td><img src="image" alt="Label" /></td>
</tr>
<tr>
<td>ProcessRef</td>
<td></td>
<td><img src="image" alt="Label" /></td>
</tr>
<tr>
<td>ServiceRef</td>
<td><img src="image" alt="Diagram" /></td>
<td>Label</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------</td>
<td>-------</td>
</tr>
<tr>
<td>SharedDataModelRef</td>
<td><img src="image" alt="Diagram" /></td>
<td>Label</td>
</tr>
<tr>
<td>UndefinedBehavior</td>
<td><img src="image" alt="Diagram" /></td>
<td>Label</td>
</tr>
<tr>
<td>OrderSetHandlerRef</td>
<td><img src="image" alt="Diagram" /></td>
<td>Label</td>
</tr>
<tr>
<td>PedigreeAndProvinanceRef</td>
<td><img src="image" alt="Diagram" /></td>
<td>Label</td>
</tr>
<tr>
<td>QuestionaireHandlerRef</td>
<td><img src="image" alt="Diagram" /></td>
<td>Label</td>
</tr>
</tbody>
</table>

**Starting Point Decorator**
The following table presents the depiction resolutions for the Multiplicity Decorator:

**Table 67. Multiplicity Decorator Depiction**

<table>
<thead>
<tr>
<th>DataItem Attribute</th>
<th>Depiction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplicity = ZeroOrMore or OneOrMore</td>
<td><img src="image1" alt="Image" /></td>
</tr>
</tbody>
</table>

**Semantic Reference Decorator**

The following table presents the depiction resolutions for the Semantic Reference Decorator:

**Table 68. Semantic Reference Decorator Depiction**

<table>
<thead>
<tr>
<th>DataItem Attribute</th>
<th>Depiction</th>
</tr>
</thead>
<tbody>
<tr>
<td>If the DataItem or any component of its associated ItemDefinition has an defined SemanticReference.</td>
<td><img src="image2" alt="Image" /></td>
</tr>
</tbody>
</table>

**18.4.5.3 BKPMNEdge Resolution**

*BKPMNEdge* can be used to represent an *Ownership Connector*, *Parent-Child Connector*, *Relationship Connector*, or a *Data Association*.

**Relationship Connector**

The following table presents the depiction resolutions for an *Association*:

**Table 69. Depiction Resolution of the Relationship Connector**

<table>
<thead>
<tr>
<th>BKPMN Element</th>
<th>Depiction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationship Connector</td>
<td><img src="image3" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td>Label</td>
</tr>
</tbody>
</table>