Modeling BPMN 2.0 in ARIS

ARIS

Version 9.7 - Service Release 2

February 2015

Product Documentation

This document applies to ARIS Version 9.7 and to all subsequent releases. Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2000 - 2015 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners. Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at http://softwareag.com/licenses.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at http://softwareag.com/licenses and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at http://softwareag.com/licenses and/or in the root installation directory of the licensed product(s).

Contents

1 Introduction 1
1.1 Initial situation and objective 1
1.2 Purpose of this document 1

2 BPMN core elements and their implementation in ARIS 2
2.1 Infrastructure 2
2.2 Foundation 2
2.3 Common elements 5
2.3.1 Artifacts 5
2.3.2 Callable elements 4
2.3.3 Correlation 5
2.3.4 Conversation association 5
2.3.5 Error 5
2.3.6 Event 5
2.3.7 Flow element 6
2.3.8 Flow elements container 7
2.3.9 Gateways 8
2.3.10 Interaction specification 8
2.3.11 Item definition 8
2.3.12 Message 9
2.3.13 Message flow 10
2.3.14 Participant 12
2.3.15 Resource 15
2.3.16 Sequence flow 16
2.4 Services 18

3 BPMN diagrams and ARIS model types: An overview 19

4 Process 20
4.1 Activities 22
4.1.1 Resource assignment 24
4.1.2 Perfomer 24
4.1.3 Activity type: Task 25
4.1.4 Human interactions 29
4.1.5 Activity type: Subprocess 29
4.1.6 Activity type: Call Activity 36
4.1.7 Global task 38
4.1.8 Loop characteristics 38
4.2 Items and Data 43
4.2.1 Data object 43
4.2.2 Data store 45
4.3 Events 46
4.3.1 Catch events and throw events 48
4.3.2 Start event 50
4.3.3 End event 50
4.3.4 Intermediate events 51
4.3.5 Event definitions 52
4.4 Gateways 58
4.4.1 Exclusive gateway 59
4.4.2 Inclusive gateway 60
4.4.3 Parallel gateway 60
4.4.4 Complex gateway 61
4.4.5 Event-based gateways 62

4.5 Lanes 63

5 Collaboration 65

5.1 Pool and participant 67
5.2 Object types and connection types reused from a process 67
5.3 Message flow 68

6 Conversation 69

6.1 Conversation container 69
6.2 Conversation nodes 69
6.3 Participant 70
6.4 Artifacts 71
6.5 Conversation link 72
6.6 Message flow in a conversation 73
6.7 Model assignments 73

1 Introduction

1.1 Initial situation and objective

BPMN (Business Process Modeling and Notation) has emerged as a widely adopted standard for process modeling. Its popularity is based on the fact that it is has been developed by the Object Management Group (OMG), a consortium of organizations that also released other important modeling standards like UML.

The primary goal of BPMN is to provide a notation that is understandable by all users: business analysts designing and documenting business processes, developers implementing these business processes, and business end users executing, managing and monitoring their business processes. Now, the OMG released a new version of BPMN 2.0. This standard shall be supported by ARIS. In a first step, the objective is to focus on process modeling conformance, one of four conformance types defined by the OMG.[1]

1.2 Purpose of this document

Unfortunately the BPMN specification has increased an order of magnitude in technical complexity and fails to distinguish those elements needed for business process modeling from those required for process execution.

The purpose of this document is to describe the ARIS implementation of the BPMN 2.0 elements that are part of business process modeling documenting the process flow. Those parts that are needed for executable design are ignored. The elements relevant for business process modeling are essentially those displayed in a diagram.

The mapping described in the chapters 2 - 6 of this document is based on the BPMN specification **Business process modeling notation (BPMN) – FTF Beta 1 for version 2.0** (http://www.bpmn.org).

The attribute and model association tables are taken from the BPMN 2.0 Beta specification and extended to describe the implementation in ARIS.

[1] The four conformance types are described in detail in the BPMN specification: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 2-6.

2 BPMN core elements and their implementation in ARIS

The BPMN core consists of four packages:

- Foundation,
- Infrastructure,
- Common elements as well as
- Service

It provides the basis for modeling processes, collaborations, choreographies and conversations. These packages are described in detail in chapter 8 of the BPMN specification.

In the following sections the core constructs and their attributes and associations are mapped to ARIS constructs.

2.1 Infrastructure[2]

The infrastructure package consists of two elements which are particularly relevant for import and export. Thus, their attributes and model associations are not considered in the 1st version of the BPMN 2.0 implementation.

2.2 Foundation[3]

The foundation package contains classes which are shared amongst other packages in the BPMN core. The foundation package consists of 8 classes: base element, documentation, root element, extension, extension definition, extension attribute definition, extension attribute value, and relationships.

The mapping of the attributes and model associations of the foundation classes is shown in Table 1.

[2] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 41-44.

[3] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 45-55.

Class	BPMN attribute name	Implementation in ARIS
BaseElement	id: string	The ARIS GUID of the respective modeling construct represents the BPMN ID. For imported BPMN elements an attribute type in the attribute type group **external attributes** will be used.
	documentation: Documentation [0..*]	see below: Documentation
	extensionDefinitions: ExtensionDefinition [0..*]	The ARIS method can be enhanced, e. g. by user defined attributes.
	extensionValues: ExtensionAttributeValue [0..*]	The ARIS method can be enhanced, e. g. by user defined attributes.
Documentation	*inherits from BaseElement*	
	text: string	All ARIS attribute types assigned to model types, object types and connection types can be used for documentation purposes. The attribute types **Description / definition** (AT_DEC) and **Remark / Example** (AT_REM) should be used to for general information. Specific attribute types should be used to store specific information.
Extension	mustUnderstand: boolean [0..1] = False	Extensions are not considered in the 1st version of the BPMN 2.0 implementation.
	definition: ExtensionDefinition	
ExtensionDefinition	name: string	Extension definitions are not considered in the 1st version of the BPMN 2.0 implementation.
	extensionAttributeDefinitions: ExtensionAttributeDefinition [0..*]	

Class	BPMN attribute name	Implementation in ARIS
ExtensionAttribute Definition	name: string	Extension attribute definitions are not considered in the 1st version of the BPMN 2.0 implementation.
	type: string	
	isReference: boolean [0..1] = False	
ExtensionAttribute Value	value: Element [0..1]	Extension attribute values are not considered in the 1st version of the BPMN 2.0 implementation.
	valueRef: Element [0..1]	
	extensionAttributeDefinition: ExtensionAttributeDefinition	
Relationship	*inherits from BaseElement*	(External) relationships are not considered in the 1st version of the BPMN 2.0 implementation.
	type: string	
	direction: RelationshipDirection {none | forward | backward | both}	
	sources: Element [1..*]	
	targets: Element [1..*]	
RootElement	*inherits from BaseElement*	RootElement is an abstract class, it has no direct representation in ARIS. E.g. ARIS object types are root elements, ARIS attribute types are not root elements.

Table 1: Foundation: Mapping the attributes and model associations to ARIS

2.3 Common elements

Common elements are basic elements that may be used in more than one type of diagram (e.g., process, collaboration, conversation, and choreography). The common elements are categorized into seventeen different groups which will be described in the following sub-chapters.

2.3.1 Artifacts[4]

Artifacts are used to depict additional information in a BPMN process or collaboration diagram that is not directly related to the sequence flow or message flow. BPMN 2.0 provides three standard artifacts:

- associations,
- groups, and
- text annotations

Data objects are no longer artifacts, they are concepts of their own (see chapter 4.2).

2.3.1.1 Association

Associations are used to associate information and artifacts with other BPMN elements. Thus, associations are (usually) represented by connection types in ARIS. The relevant connection types are described in the context of the object types being associated.

The attributes and model associations of an association and their mapping to ARIS constructs are listed in Table 2.

[4] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 55-63.

2.3.1.2 Group

BPMN 2.0 uses three different classes to represent groupings, but there is only one symbol: **group**. Thus, a group is the graphical representation of a category value.

Categories and their category values are modeled in an auxiliary model of type **Structuring model** (see Figure 1).

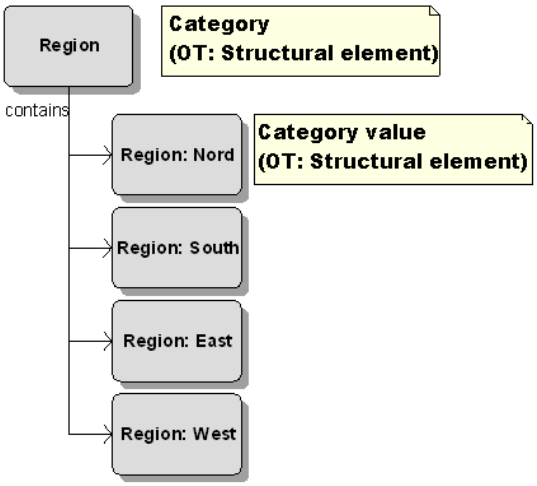

Figure 1: Structuring model: Categories and their values

In ARIS the graphical element **group** is an occurrence copy of a category value object and is depicted by a special symbol in the BPMN 2.0 models. The symbol name is **Group** (see Figure 2).

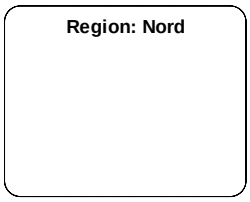

Figure 2: Group symbol

2.3.1.3 Text annotation

Text annotations are used to provide additional textual information for the reader of a BPMN model. They can be associated with graphical elements (ARIS objects and connections) in a model.

The symbol representing a text annotation is shown in Figure 3.

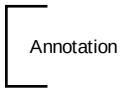

Figure 3: Symbol representing text annotations

Text annotations will be implemented in ARIS in 2 different ways:

Text annotations associated with ARIS objects:

The object type **Text annotation** and the connection type **is associated with** will be used to annotate objects (occurrences) in a model.

Text annotations associated with ARIS connections:

Objects (here: text annotation) cannot be assigned to connections. Thus, the BPMN Designer provides a new functionality: The modeler selects the text annotation symbol in the symbol bar, places it on/near by the connection he/she wants to annotate and enters the text. The BPMN Designer draws a line looking like an association and stores the text in a text annotation attribute of the respective connection. In the first step three text annotation attributes are provided in the attribute type group BPMN 2.0 attributes/Text annotations:

Text annotation 1 (AT_BPMN_TEXT_ANNOTATION_1)

Text annotation 2 (AT_BPMN_TEXT_ANNOTATION_2)

Text annotation 3 (AT_BPMN_TEXT_ANNOTATION_3)

The BPMN attributes and model associations of text annotations and their mapping to ARIS constructs are listed in Table 2.

Class	BPMN attribute name	Implementation in ARIS
Association	*inherits from BaseElement*	different CTs
	associationDirection: AssociationDirection = None {None | One | Both}	This attribute is represented by the direction and the style of the respective ARIS connection type.
	sourceRef: BaseElement	Corresponds to the source object type of the connection type representing the association.
	targetRef: BaseElement	Corresponds to the target object type of the connection type representing the association.
Group	*inherits from BaseElement*	Object type: Structural element (OT_STRCT_ELMT) Symbol: Structural element in model type Structuring model (MT_STRCT_DGM) Symbol: Group (ST_BPMN_GROUPING_1) in BPMN 2.0 diagrams
	categoryValueRef: CategoryValue [0..1]	Attribute type **Name** (AT_NAME) of object type Structural element (OT_STRCT_ELMT)
Category	*inherits from BaseElement*	Object type Structural element (OT_STRCT_ELMT) in model type Structuring model (MT_STRCT_DGM)
	categoryValue: CategoryValue [0..*]	Connection type in model type Structuring model: * Structural element (representing the category) contains structural element (representing the category value).
CategoryValue	*inherits from BaseElement*	Object type: Structural element (OT_STRCT_ELMT) Symbol: Structural element in model type Structuring model (MT_STRCT_DGM) Symbol: Group (ST_BPMN_GROUPING_1) in BPMN 2.0 diagrams
	value: string	Attribute type **Name** of object type Structural element
	category: Category [0..1]	Connection type in model type Structuring model:

Class	BPMN attribute name	Implementation in ARIS
		* Structural element (representing the category) contains structural element (representing the category value).
	categorizedFlowElements: FlowElement [0..*]	Connection type **belongs to** [CT_BELONGS_TO_1] In the BPMN 2.0 process diagram and BPMN 2.0 collaboration diagram: Target object type: Structural element (OT_STRCT_ELMT; ST_BPMN_GROUPING_1) Source object types: * Function (OT_FUNC) representing activities * Event (OT_EVT) * Rule (OT_RULE) representing Gateways * Cluster/data model (OT_CLST) representing data objects * Information carrier (OT_INFO_CARR) representing data stores
Text annotation	*inherits from BaseElement*	
	text: string	For object types in the BPMN 2.0 process diagram, BPMN 2.0 collaboration diagram, and BPMN 2.0 conversation diagram: * Text annotation (OT_BPMN_ANNOTATION, ST_BPMN_ANNOTATION_1), is associated with <target object type>. Target object types are all object types available in the respective model type. For connection types in the BPMN 2.0 process diagram, BPMN 2.0 collaboration diagram, and BPMN 2.0 conversation diagram: three attribute types in the attribute type group BPMN 2.0 attributes/Text annotation attributes: * Text annotation 1 (AT_BPMN_TEXT_ANNOTATION_1) * Text annotation 2 (AT_BPMN_TEXT_ANNOTATION_2) * Text annotation 3 (AT_BPMN_TEXT_ANNOTATION_3)

Table 2: Artifacts: Mapping the attributes and model associations to ARIS

2.3.2 Callable elements[5]

Callable element is an abstract class and has four specialized classes: process, global task, choreography, and choreography task. Only processes and global tasks are relevant for business process modeling compliance. They are represented by the object type **Function**.

The attributes and model associations of callable elements and their mapping to ARIS constructs are listed in Table 3.

Class	BPMN attribute name	Implementation in ARIS
Callable element	*inherits from BaseElement*	Object type: Function (OT_FUNC) Symbol: Call activity (OT_BPMN_CALL_ACTIVITY)
	name: string [0..1]	Attribute type **Name** (AT_NAME) of object type function (OT_FUNC)
	supportedInterfacesRefs: Interface [0..*]	Interfaces are not considered in the 1st version of the BPMN 2.0 implementation.
	ioSpecification: InputOutputSpecification [0..1]	IO specifications are not considered in the 1st version of the BPMN 2.0 implementation.
	ioBinding: InputOutputBinding [0..*]	InputOutput binding is not considered in the 1st version of the BPMN 2.0 implementation.
InputOutputBinding	inputData: DataInput	InputOutput binding is not considered in the 1st version of the BPMN 2.0 implementation.
	outputData: DataOutput	
	operationRef: Operation	

Table 3: Callable element: Mapping the attributes and model associations to ARIS

[5] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 63-65.

2.3.3 Correlation[6]

Correlations belong to the execution design level and are not considered in the 1st version of the BPMN 2.0 implementation.

2.3.4 Conversation association[7]

Conversation associations belong to the execution design level and are not considered in the 1st version of the BPMN 2.0 implementation.

2.3.5 Error

Errors belong to the execution design level and are not considered in the 1st version of the BPMN 2.0 implementation.

2.3.6 Event[8]

Events are described in detail in the context of the BPMN process diagram (see chapter 4.3).Expression[9]

Formal expressions belong to the execution design level and are not considered in the 1st version of the BPMN 2.0 implementation.

However, natural-language expressions are used to allow the modeler to specify conditions. They are described in the context of the respective BPMN elements (object types and connection types).

[6] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 65-70.
[7] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 71-72.
[8] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 72-74.
[9] [9] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 74-75.

2.3.7 Flow element[10]

Flow elements are described in detail in the context of the BPMN process diagram (see chapter 4).

Their general attributes and model associations of flow elements and their mapping to ARIS constructs are shown in Table 4.

Class	BPMN attribute name	Implementation in ARIS
FlowElement	*inherits from BaseElement*	no direct representation in ARIS -> abstract class
	name: string [0..1]	Attribute type **Name** (AT_NAME) of the object types representing flow nodes.
	auditing: Auditing [0..1]	Auditing is not considered in the 1st version of the BPMN 2.0 implementation.
	monitoring: Monitoring [0..1]	Monitoring is not considered in the 1st version of the BPMN 2.0 implementation.

Table 4: Flow element: Mapping the attributes and model associations to ARIS

[10] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 75-77.

2.3.8 Flow elements container[11]

A flow elements container is an abstract super class for BPMN diagrams (or views). So, processes and subprocesses as well as choreographies and choreography subprocess are flow element containers.

The general attributes and model associations of a flow elements container and their mapping to ARIS constructs are listed in Table 5.

The specific attributes and model associations of a process and subprocess are described in detail in the context of the BPMN process diagram.

Class	BPMN attribute name	Implementation in ARIS
FlowElementsContainer	*inherits from BaseElement*	Model type BPMN process diagram (BPMN 2.0) (MT_BPMN_PROCESS_DIAGRAM)
	flowElements: FlowElement [0..*]	Occurrences of the object types and connection types allowed in a BPMN process diagram (BPMN 2.0).
	artifacts: Artifact [0..*]	Occurrences of the object types and attribute types representing groups and text annotations as well as their connection types allowed in the BPMN process diagram (BPMN 2.0).

Table 5: Flow elements container: Mapping the attributes and model associations to ARIS

[11] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 77-78.

2.3.9 Gateways[12]

Gateways are described in detail in the context of the BPMN process diagram (see chapter 4.4).

The attributes and model associations of a gateway are summarized in Table 6.

Class	BPMN attribute name	Implementation in ARIS
Gateways	*inherits from FlowElement*	Object type: Rule (OT_RULE)
	gatewayDirection: GatewayDirection = unspecified { unspecified | converging | diverging | mixed }	The number of incoming and outgoing sequence flows depends on the modeling context, i.e. the position of the gateway in the process. Thus, there is no ARIS attribute type representing the gateway direction. Unspecified and mixed gateways should be avoided.

Table 6: Gateway: Mapping the attributes and model associations to ARIS

2.3.10 Interaction specification[13]

Interaction specifications belong to the execution design level and are not considered in the 1st version of the BPMN 2.0 implementation.

2.3.11 Item definition[14]

Item definitions belong to the execution design level and are not considered in the 1st version of the BPMN 2.0 implementation.

[12] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 78-80.
[13] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 80-81.
[14] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 81-83.

2.3.12 Message[15]

Messages normally represent information exchanged between two participants in a BPMN collaboration diagram.

A message is represented by the symbol **message** of the ARIS object type **Message** (see Figure 4).

Figure 4: Message symbol

The attributes and model associations of a message and their mapping to ARIS are summarized in Table 1.

Class	BPMN attribute name	Implementation in ARIS
Message	*inherits from BaseElement*	Object type: Message (OT_MSG_FLW) Symbol: Message (ST_BPMN_MESSAGE_2)
	name: string	Attribute type **Name** (AT_NAME) of object type Message (OT_MSG_FLW)
	structureRef : ItemDefinition [0..1]	The structure of a message is not considered in the 1st version of the BPMN 2.0 implementation.

Table 7: Message: Mapping the attributes and model associations to ARIS

[15] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 83-86.

2.3.13 Message flow[16]

The message exchange between participants is shown by a message flow that connects two pools (or the objects within the pools).

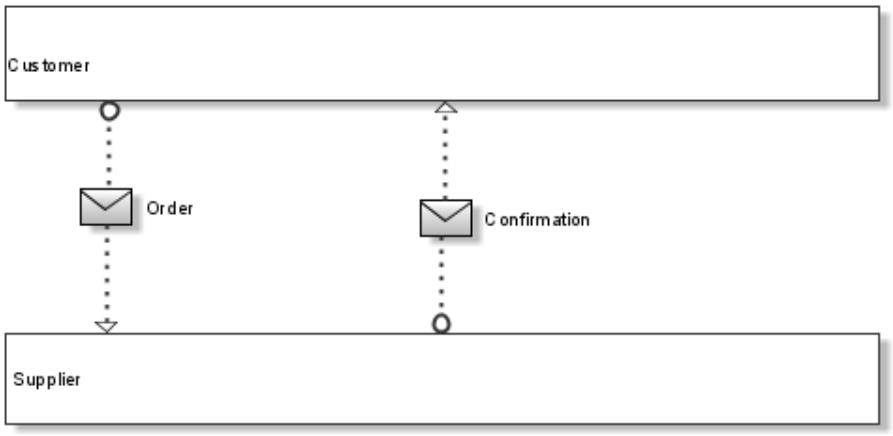

Figure 5: Message flow between participants/pools

A message flow is represented in ARIS by the connection type **Message flow**. If the message sent from one participant to another should be displayed in the diagram, the connection type **message flow** is replaced by the object type **Message** (symbol: message) and two connection types:

- <Source object type> sends message.
- Message is received from <target object type>.

More details can be found in chapter 5.3.

Its specific attributes and model associations as well as the mapping to ARIS constructs are listed in Table 8.

Message flow associations are use to map message flows modeled in two different diagrams, e.g. in a conversation and a collaboration diagram. These associations are realized in ARIS by occurrence copies of the message flow connections.

The attributes and model associations of a message flow association are also shown in Table 8.

Message flow is also described in the context of the BPMN collaboration diagram (chapter 5.3) and the BPMN conversation diagram (chapter 6.6).

[16] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 86-91.

Class	BPMN attribute name	Implementation in ARIS
Message flow	*inherits from BaseElement*	Connection type: Message flow (CT_BPMN_MESAGE_FLOW)
	name: string	Attribute type **Connection role** of connection type Message flow (CT_BPMN_MESSAGE_FLOW)
	sourceRef: MessageFlowNode	Source object type of connection type Message flow (CT_BPMN_MESAGE_FLOW) (Participant, Function, Event)
	targetRef: MessageFlowNode	Target object type of connection type Message flow (CT_BPMN_MESAGE_FLOW) (Participant, Function, Event)
	messageRef: Message [0..1]	Object type: Message (OT_MSG_FLW) Symbol: Message (ST_BPMN_MESSAGE_2) Connection types in the BPMN collaboration diagram: * Participant sends (CT_SENDS_2) message. * Event sends (CT_SENDS_2) message. * Function sends (CT_SENDS_2) message. * Message is received from (CT_IS_RECEIVED_FROM) participant. * Message is received from (CT_IS_RECEIVED_FROM) function. * Message is received from (CT_IS_RECEIVED_FROM) event.
Message flow node		Object types that can be source or target of connection type Message flow (CT_BPMN_MESSAGE_FLOW): Participant (OT_BPMN_POOL), Function (OT_FUNC), Event (OT_EVT)
Message flow association	*inherits from BaseElement*	This association is used to map message flows modeled in a collaboration and a conversation diagram.
	innerMessageFlowRef: Message Flow	Occurrence copy of a message flow connection in a BPMN collaboration diagram and BPMN conversation diagram.
	outerMessageFlowRef: Message Flow	Occurrence copy of a message flow connection in a BPMN collaboration diagram and BPMN conversation diagram.

Table 8: Message flow: Mapping the attributes and model associations to ARIS

2.3.14 Participant[17]

A participant represents a partner entity and/or a partner role that participates in a collaboration. Participants may be modeled in a BPMN collaboration diagram or a BPMN conversation diagram.

The assignment of a partner entity and/or a partner role to a participant is transferred to the BPMN allocation diagram (BPMN 2.0) assigned to the participant (see Figure 6).

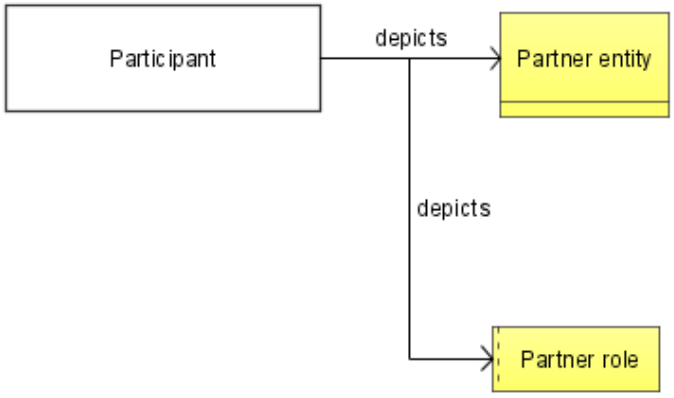

Figure 6: BPMN allocation diagram (BPMN 2.0): Participant and partner entity/partner role.

The usage of participants is described in the context of the BPMN collaboration diagram (see chapter 5.1) and the BPMN conversation diagram (see chapter 6.3).

Participant, partner entity and partner role inherit from base element; their specific attributes and model associations as well as their mapping to ARIS are shown in Table 9.

[17] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 91-96.

Class	BPMN attribute name	Implementation in ARIS
Participant	*inherits from BaseElement*	Object type: Participant (OT_BPMN_POOL) Symbol: Pool (ST_BPMN_POOL_1)
	name: string [0..1]	Attribute type **Name** (AT_NAME) of object type Participant (OT_BPMN_POOL)
	processRef: Process [0..1]	BPMN process diagram (BPMN 2.0) assigned to the participant (OT_BPMN_POOL) Process displayed within in the pool
	partnerRoleRef: PartnerRole [0..1]	Model type: BPMN allocation diagram (BPMN 2.0): Object type: Person type (OT_PERS_TYPE) Symbol: Partner role (ST_BPMN_PARTNER_ROLE) Connection type: Participant depicts (CT_DEPICTS_1) Person type
	partnerEntityRef: PartnerEntity [0..1]	Model type: BPMN allocation diagram (BPMN 2.0): Object type: Organizational unit (OT_ORG_UNIT) Symbol: Partner entity (ST_BPMN_PARTNER_ENTITY) Connection type: Participant depicts (CT_DEPICTS_1) organizational unit
	interfaceRef: Interface [0..*]	Interfaces are not considered in the 1st version of the BPMN 2.0 implementation.
	participantMultiplicity: participantMultiplicity [0..1]	Attribute type in the attribute type group **BPMN 2.0 attributes/Participant multiplicity** of the object type Participant (OT_BPMN_POOL): * Multi-instance participant (AT_MI_PARTICIPANT) The minisymbol (three vertical lines) will be displayed by the BPMN Designer if the value of the attribute type **Multi-instance participant** is set to **true**.
	endpointRefs: EndPoint [0..*]	End points are not considered in the 1st version of the BPMN 2.0 implementation.
Partner Entity	*inherits from BaseElement*	Object type: Organizational unit (OT_ORG_UNIT) Symbol: Partner entity (ST_BPMN_PARTNER_ENTITY)
	name: string	Attribute type **Name** (AT_NAME) of object type Organizational unit (OT_ORG_UNIT)
Partner Role	*inherits from BaseElement*	Object type: Person type (OT_PERS_TYPE) Symbol: Partner role (ST_BPMN_PARTNER_ROLE)
	name: string	Attribute type **Name** of object type Person type (OT_PERS_TYPE)

Class	BPMN attribute name	Implementation in ARIS
Participant Multiplicity	minimum: integer [0..1] = 2	Attribute type in the attribute type group **BPMN 2.0 attributes / Participant multiplicity** of the object type Participant (OT_BPMN_POOL): * Minimum participant multiplicity (AT_MINIMUM_MI_PARTICIPANT)
	maximum: integer [0..1] = 2	Attribute type in the attribute type group **BPMN 2.0 attributes / Participant multiplicity** of the object type Participant (OT_BPMN_POOL): * Maximum participant multiplicity (AT_MAXIMUM_MI_PARTICIPANT)
Participant Association	*inherits from BaseElement*	
	innerParticipantRef: Participant	Occurrence copy of the respective participant.
	outerParticipantRef: Participant	Occurrence copy of the respective participant.

Table 9: Participant: Mapping the attributes and model associations to ARIS

2.3.15 Resource[18]

Resources can be human resources as well as any other resource assigned to activities during process execution. A direct mapping of the BPMN resources to ARIS constructs is not possible - due to the semantically different object types representing resources in ARIS. ARIS does not only provide different object types, but also different connection types.

BPMN 2.0 only knows one object type called **Resource**. The BPMN ActivityResource and its specialized sub-classes correspond to ARIS connection types in combination with object types. Therefore, resources are not considered in the 1st version of the BPMN 2.0 implementation.

[18] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 96-98.

2.3.16 Sequence flow[19]

The BPMN sequence flow is mapped to nine different ARIS connection types, which are used to depict the control flow in traditional ARIS process models (see Table 10).

Source object type	Connection type	Target object type
Event	Occurs before	Event
Event	Activates	Function
Event	Is evaluated by	Rule
Function	Creates	Event
Function	Is predecessor of	Function
Function	Leads to	Rule
Rule	Leads to	Event
Rule	Activates	Function
Rule	Links	Rule

Table 10: Connection types representing sequence flow

BPMN distinguishes three types of sequence flow (see Figure 1):

- **Unconditional sequence flow**

  The unconditional sequence flow means the **normal** flow, no specific conditions apply (in other words: its condition has always the value **true**). It is depicted by a solid line with a solid arrowhead.

- **Conditional sequence flow**

  The conditional sequence flow out of an activity is drawn with a little diamond at the beginning of the connector, signifying a data condition. A conditional sequence flow out of a gateway shares the same shape as a normal sequence flow.

- **Default sequence flow**

  The default sequence flow, denoted by a slash marker at the beginning of the connector means **otherwise**, i. e. it is enabled if no other sequence flow condition evaluates to **true**.

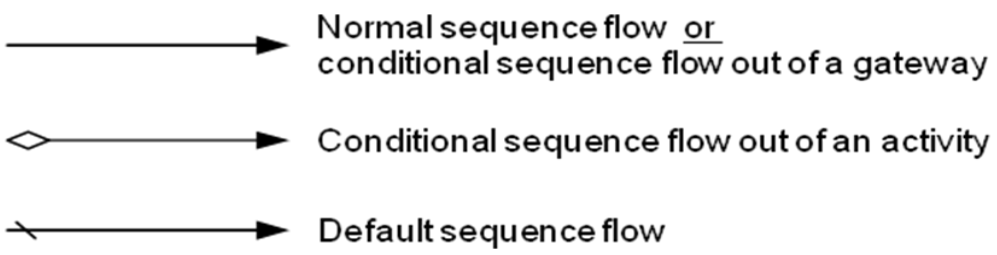

Figure 7: Sequence flow styles

[19] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 98-101.

All connection types used in BPMN models must hold attributes for recording text annotations (see chapter 2.3.1.3). Connection types emerging from activities and gateways need additional attributes for recording sequence flow conditions. The BPMN specific attributes and model associations of sequence flow connections are listed in Table 11.

Class	BPMN attribute name	Implementation in ARIS
Sequence flow	*inherits from FlowElement*	The sequence flow is depicted by nine different connection types in the model types BPMN process diagram (BPMN 2.0) (MT_BPMN_PROCESS_DIAGRAM) and BPMN collaboration diagram (BPMN 2.0) (MT_BPMN_COLLABORATION_DIAGRAM): * event occurs before (CT_SUCCEED) event * event activates (CT_ACTIV_1) function * event is evaluated (CT_IS_EVAL_BY_1) by rule * function creates (CT_CRT_1) event * function is predecessor of (CT_IS_PREDEC_OF_1) function * function leads (CT_LEADS_TO_1) to rule * rule leads to (CT_LEADS_TO_2) event * rule activates (CT_ACTIV_1) function * rule links (CT_LNK_2) rule
	name: string	Attribute type **Connection role** of connection type Message flow (CT_BPMN_MESSAGE_FLOW)
	sourceRef: FlowNode	Source object of a sequence flow connection: Object types are: Function, Event, Rule
	targetRef: FlowNode	Target object of a sequence flow connection: Object types are: Function, Event, Rule
	conditionExpression : Expression [0..1]	Attribute type **Condition expression** (AT_BPMN_CONDITION_EXPRESSION) in attribute type group **BPMN 2.0 attributes** of the following connection types: * activates (CT_ACTIV_1) * creates (CT_CRT_1) * links (CT_LNK_2) * leads to (CT_LEADS_TO_1) * leads to (CT_LEADS_TO_2) * is predecessor of (CT_IS_PREDEC_OF_1) The value of the attribute type **Sequence flow condition** in the attribute type group **BPMN 2.0 attributes** must be set to **Conditional sequence flow**.

Class	BPMN attribute name	Implementation in ARIS
	isImmediate: boolean [0..1]	This attribute type is not considered in the 1st version of the BPMN 2.0 implementation.
Flow node	incoming: Sequence Flow [0..*]	Incoming connections representing the sequence flow of the flow node object (object types: function, event, rule)
	outgoing: Sequence Flow [0..*]	Outgoing connections representing the sequence flow of the flow node object (object types: function, event, rule)

Table 11: Sequence flow: Mapping the attributes and model associations to ARIS

2.4 Services[20]

Services belong to the execution design level and are not considered in the 1st version of the BPMN 2.0 implementation.

[20] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 107-109.

3 BPMN diagrams and ARIS model types: An overview

According to the BPMN 2.0 specification three diagram types are required for process modeling conformance: process diagram, collaboration diagram and conversation diagram.[21]

A careful consideration of these BPMN diagrams shows that the modeling constructs of the process diagram are a subset of the modeling constructs used in the collaboration diagram. There are also overlapping constructs in the collaboration and conversation diagram.

The model types listed in Table 12 will be provided in ARIS.

The BPMN allocation diagram allows the mapping of BPMN attributes and associations to the semantically richer ARIS method where graphical elements are often used to represent BPMN attributes and associations.

BPMN diagram	ARIS model type
Process diagram	BPMN process diagram (BPMN 2.0)
Collaboration diagram	BPMN collaboration diagram (BPMN 2.0)
Conversation diagram	BPMN conversation diagram (BPMN 2.0)
/	BPMN allocation diagram (BPMN 2.0)

Table 12: ARIS model types for process modeling conformance

[21] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 2.

4 Process[22]

The BPMN process diagram depicts a BPMN process. A process is a specialization of a flow elements container. So, it contains the following elements:

- flow nodes (event, activity, and gateway)
- sequence flow
- artifacts (see chapter 2.3.1)

The attributes and model associations specific to a process can be found in Table 13.

Class	BPMN attribute name	ARIS implementation
Process	*from CallableElement* *from FlowElementsContainer*	Model type: BPMN process diagram (BPMN 2.0)
	processType: ProcessType = none { none | executable | non-executable | public }	Attribute type in the attribute type group **BPMN 2.0 attributes** of model type BPMN process diagram (BPMN 2.0): * Process type (AT_BPMN_PROCESS_TYPE) Attribute values: * Undefined (= none), * Executable process (ATV_BPMN_EXECUTABLE), * Non-executable process (AVT_BPMN_NON_EXECUTABLE) * Public process (AVT_BPMN_PUBLIC)
	auditing: Auditing [0..1]	Auditing is not considered in the 1st version of the BPMN 2.0 implementation.
	monitoring: Monitoring [0..1]	Monitoring is not considered in the 1st version of the BPMN 2.0 implementation.
	laneSets: LaneSet [0..*]	Object type **Lane** (OT_BPMN_LANE) Symbol: Lane (ST_BPMN_LANE_1)
	IsClosed: boolean = false	Attribute type **Is closed** (AT_BPMN_IS_CLOSED) in attribute type group **BPMN 2.0 attributes** of the BPMN process diagram
	supports: Process [0..*]	This connection type is not considered in the 1st version of the BPMN 2.0 implementation.
	properties: Property [0..*]	Properties are not considered in the 1st version of the BPMN 2.0 implementation.
	definitionalCollaborationRef: Collaboration [0..1]	The BPMN collaboration diagram (BPMN 2.0) that contains the process

[22] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 121-292.

Table 13: Process: Mapping the attributes and model associations to ARIS

A process is a particular construct: On the one hand it is a model. On the other hand a process can be visualized within a pool in a collaboration. But a pool is not identical with a process, and vice versa. A pool represents a participant in a collaboration (see chapter 5). A pool may contain the process the participant uses in a specific collaboration.

The core elements for modeling a BPMN process are those constructs which can be connected to each other by sequence flow. They are called flow nodes. The corresponding ARIS object types and their symbols provided in the symbol bar are listed in the table below.

BPMN element	ARIS object type	ARIS symbol	API name
Event	Event (OT_EVT)	Start event Intermediate event End event	ST_BPMN_START_EVENT ST_BPMN_INTERMEDIATE_EVENT ST_BPMN_END_EVENT
Activity	Function (OT_FUNC)	Task Subprocess Call activity	ST_BPMN_TASK ST_BPMN_SUBPROCESSS ST_BPMN_CALL_ACTIVITY
Gateway	Rule (OT_RULE)	Gateway	ST_BPMN_RULE_1

Table 14: Object types and their symbols representing elements of a sequence flow

These constructs are described in detail in the separate chapters (see below).

4.1 Activities

The BPMN activity is represented by the ARIS object type **Function**.

BPMN 2.0 differentiates three basic types of activities: task (atomic activity), subprocess (non-atomic activity) and call activity. The symbols depicting these activity types are provided in the ARIS symbol bar (see Figure 8).

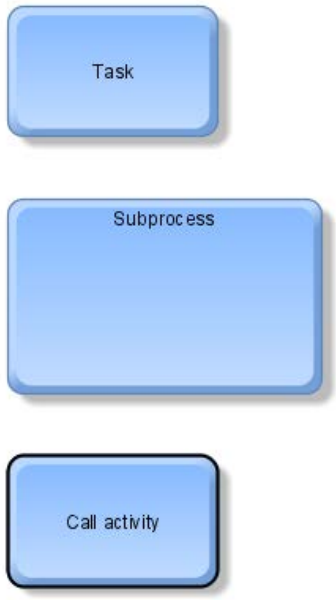

Figure 8: Symbols representing activities in the symbol bar

When the modeler places an activity symbol, the software sets the corresponding value of the ARIS attribute type **Activity type** (AT_BPMN_ACTIVITY_TYPE). This activity type controls the correct behavior of the symbol. E. g: A subprocess may have **embedded** flow elements, a task must not; a call activity may reference another task or process, tasks and subprocesses must not.

The attributes and model associations of an activity and their mapping to ARIS constructs are shown in Table 15.

Class	BPMN attribute name	ARIS implementation
Activity	*inherits from FlowElement*	Object type: Function (OT_FUNC) Attribute type **Activity type** (AT_BPMN_ACTIVITY_TYPE) in the attribute type group **BPMN 2.0 attributes** of object type **Function** Attribute values: * Task (AVT_BPMN_TASK) * Subprocess (AVT_BPMN_SUBPROCESS) * Call activity (AVT_BPMN_CALL_ACTIVITY)
	isForCompensation: boolean = false	Attribute type **Is for compensation** (AT_BPMN_IS_FOR_COMPENSATION) in the attribute type group **BPMN 2.0 attributes** of object type **Function**
	loopCharacteristics: LoopCharacteristics [0..1]	see below: Loop characteristics
	resources: ActivityResource [0..*]	Resources are not considered in the 1st version of the BPMN 2.0 implementation.
	default: SequenceFlow [0..1]	Attribute type **Sequence flow condition** (AT_BPMN_SEQ_FLOW_CONDITION) in the attribute type group **BPMN 2.0 attributes** of the following conenction types: * Activity creates ..., * Activity is predecessor of ..., * Activity leads to ... The attribute value must be set to **Default sequence flow**.
	ioSpecification: InputOutputSpecification [0..1]	IO specification is not considered in the 1st version of the BPMN 2.0 implementation.
	properties: Property [0..*]	Properties are not considered in the 1st version of the BPMN 2.0 implementation.
	boundaryEventRefs: BoundaryEvent [0..*]	Connection type: Function can trigger event CT_BPMN_CAN_TRIGGER
	dataInputAssociations: DataInputAssociation [0..*]	Not considered in the 1st version of the BPMN 2.0 implementation.
	dataOutputAssociations: DataOutputAssociation [0..*]	Not considered in the 1st version of the BPMN 2.0 implementation.

Class	BPMN attribute name	ARIS implementation
	startQuantity: integer = 1	Not considered in the 1st version of the BPMN 2.0 implementation.
	completionQuantity: integer = 1	Not considered in the 1st version of the BPMN 2.0 implementation.

Table 15: Activity: Mapping the attributes and model associations to ARIS

4.1.1 Resource assignment[23]

Resource assignments are not considered in the 1st version of the BPMN 2.0 implementation. They will be dealt with in detail when implementing the execution design level in ARIS.

4.1.2 Perfomer[24]

Performers are not considered in the 1st version of the BPMN 2.0 implementation. They will be dealt with in detail when implementing the execution design level in ARIS.

[23] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 131-133.

[24] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 133.

4.1.3 Activity type: Task[25]

BPMN 2.0 distinguishes eight task types which are represented by different symbols (see Figure 9). Only the abstract task will be available in the symbol bar. The symbols of the remaining seven special task types are not available in the symbol bar, they are handled by the BPMN Designer.

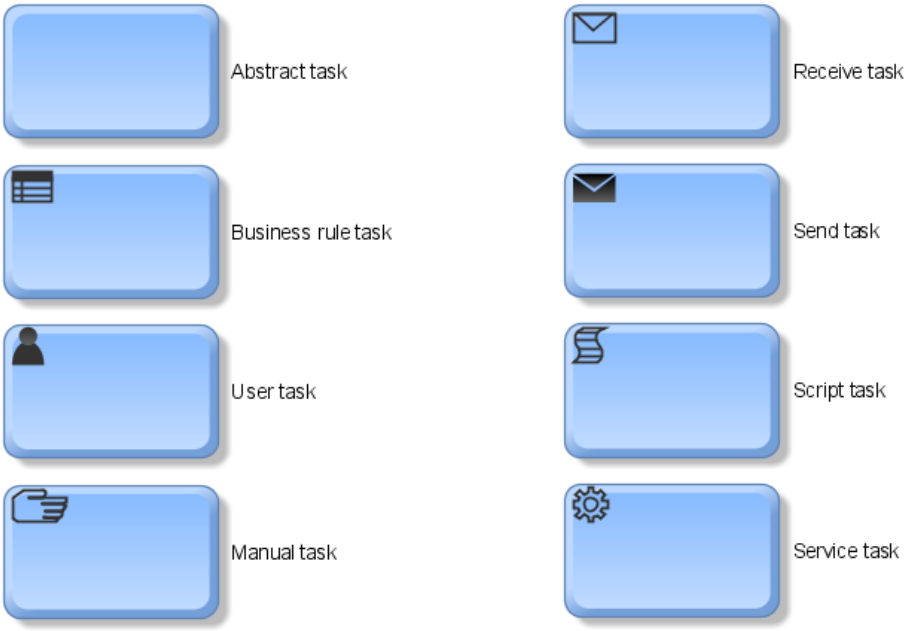

Figure 9: Task symbols

When the modeler selects a specific task symbol the software sets the corresponding value of the ARIS attribute type **Task type**. This attribute type is read-only, but visible; it provides the following values: Abstract task, Business rule task, Manual task, Script task, Send task, Service task, Receive task, and User task.

The attributes and model associations of a task and their mapping to ARIS constructs are shown in Table 16.

[25] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 133-143.

Class	BPMN attribute name	ARIS implementation
Task	*inherits from Activity*	The value of the attribute type **Activity type** (AT_BPMN_ACTIVITY_TYPE) is set to **Task** in the attribute type group **BPMN 2.0 attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: Task (ST_BPMN_TASK) or a special task symbol (see below)
Service task	*inherits from Activity*	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **Service task** in the attribute type group **BPMN 2.0 attributes / Task attributes** of object type **Function** Object type: Function (OT_FUNC) Symbol: Service task (ST_BPMN_SERVICE_TASK)
	implementation: Implementation = Web Service {Web Service | Other | Unspecified}	Not considered in the 1st version of the BPMN 2.0 implementation.
	operationRef: Operation [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
Send task	*inherits from Activity*	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **Send task** in the attribute type group **BPMN 2.0 attributes / Task attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: Send task (ST_SEND_TASK)
	messageRef: Message [0..1]	Connection type in the BPMN collaboration diagram (BPMN 2.0) * Function sends message.
	operationRef: Operation [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
	implementation: Implementation = Web Service {Web Service| Other | Unspecified}	Not considered in the 1st version of the BPMN 2.0 implementation.

Class	BPMN attribute name	ARIS implementation
Receive task	*inherits from Activity*	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **Receive task** in the attribute type group **BPMN 2.0 attributes / Task attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: Receive task (ST_RECEIVE_TASK)
	messageRef: Message [0..1]	Connection type in the BPMN collaboration diagram (BPMN 2.0) * Message is received from function
	Instantiate: boolean = False	Not considered in the 1st version of the BPMN 2.0 implementation.
	operationRef: Operation [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
	implementation: Implementation = Web Service {Web Service | Other | Unspecified}	Not considered in the 1st version of the BPMN 2.0 implementation.
User Task	*inherits from Activity*	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **User task** in the attribute type group **BPMN 2.0 attributes / Task attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: User task (ST_USER_TASK)
	Implementation: UserTaskImplementation = Other {HumanTaskWebService | WebService | Other | Unspecified}	Not considered in the 1st version of the BPMN 2.0 implementation.
	renderings: Rendering [0..*]	Not considered in the 1st version of the BPMN 2.0 implementation.
Manual Task	*inherits from Activity*	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **Manual task** in the attribute type group **BPMN 2.0 attributes / Task attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: Manual task (ST_MANUAL_TASK)

Class	BPMN attribute name	ARIS implementation
Business Rule Task	*inherits from Activity*	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **Business rule task** in the attribute type group **BPMN 2.0 attributes / Task attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: Business rule task (ST_BUSINESS_RULE_TASK)
	Implementation: BusinesRuleTaskImplementation = Other {BusinessRuleWebService | WebService | Other | Unspecified}	Not considered in the 1st version of the BPMN 2.0 implementation.
Script Task	*inherits from Activity*	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **Script task** in the attribute type group **BPMN 2.0 attributes / Task attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: Script task (ST_SCRIPT_TASK)
	scriptLanguage: string [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
	script: string [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.

Table 16: Task: Mapping the attributes and model associations to ARIS

4.1.4 Human interactions[26]

User tasks and manual tasks are relevant for modeling human interactions. Their attributes and model associations can also be found in chapter 4.1.3. Human interactions will be dealt with in detail when implementing the execution design level in ARIS.

4.1.5 Activity type: Subprocess[27]

BPMN 2.0 knows four types of subprocesses:

- (standard) subprocess[28],
- event subprocess.
- transaction, and
- adhoc subprocess

Each type of a subprocess can be displayed as

- collapsed subprocess or
- expanded subprocess

Collapsed subprocesses have a special marker displayed at the bottom of the respective subprocess symbol:

[26] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 143-152.
[27] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 152-164
[28] A (standard) subprocess corresponds to the embedded subprocess in BPMN 1.x.

4.1.5.1 Subprocess type: Subprocess

A standard subprocess shares the same shape as a task. In the collapsed form, the subprocess object uses the **+**-marker to distinguish it from a task. Expanded subprocesses have no marker, they reveal their **embedded** objects (see Figure 10).

The symbol representing the expanded subprocess is available in the symbol bar, the symbol representing the collapsed subprocess will be handled by the software.

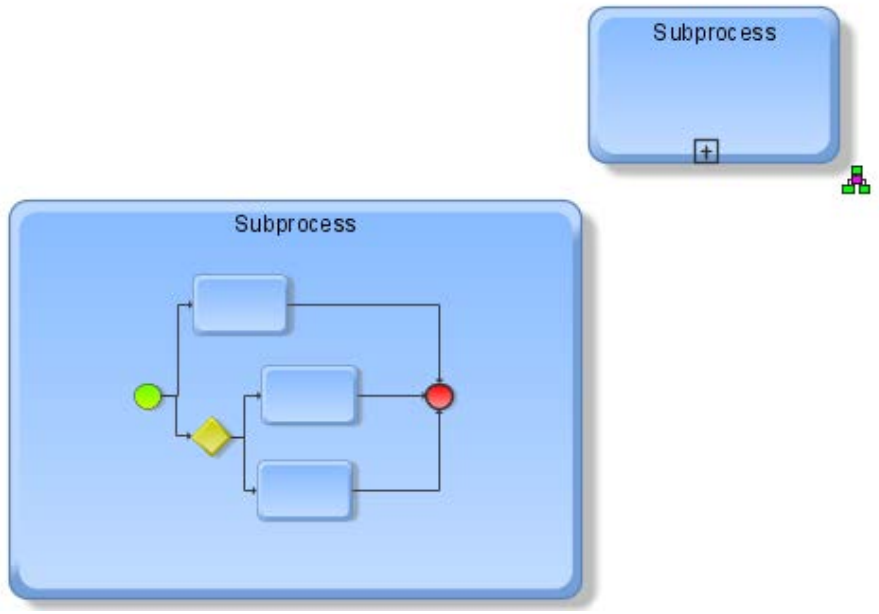

Figure 10: Symbols of a standard subprocess

The attributes and model associations of a subprocess and their mapping to ARIS constructs are listed in Table 17.

Class	BPMN attribute name	ARIS implementation
Subprocess	*inherits from Activity* *inherits from FlowElementsContainer*	The value of the attribute type **Activity type** (AT_BPMN_ACTIVITY_TYPE) is set to **Subprocess** in the attribute type group **BPMN 2.0 attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbols: * Subprocess (ST_BPMN_SUB_PROCESS) * Subprocess collapsed (ST_BPMN_SUB_PROCESS_COLLAPSED) * or a special subprocess symbol (see below)
	triggeredByEvent: boolean = false	Attribute type **Event subprocess** (AT_BPMN_EVENT_SUB_PROCESS) in the attribute type group **BPMN 2.0 attributes/Subprocess attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbols: * Event subprocess (ST_BPMN_EVENT_SUBPROCESS) * Event subprocess (collapsed) (ST_BPMN_EVENT_SUBPROCESS_COLLAPSED) The symbols are rendered by the BPMN Designer.

Table 17: Subprocess: Mapping the attributes and model associations to ARIS

4.1.5.2 Subprocess type: Event subprocess

An event subprocess is a specialized subprocess that is used within a process or a subprocess. Unlike a standard subprocess which uses the flow of its parent process as a trigger, an event subprocess is not part of the normal flow of its parent process, there is no incoming and outgoing sequence flow. An event subprocess has a start event with a trigger. Each time the start event is triggered while the parent process is active, then the event subprocess will start.

The symbols of an event subprocess are shown in Figure 11. If the event subprocess is collapsed, then its start event will be used as a marker in the upper left corner of the symbol. The software will render this marker.

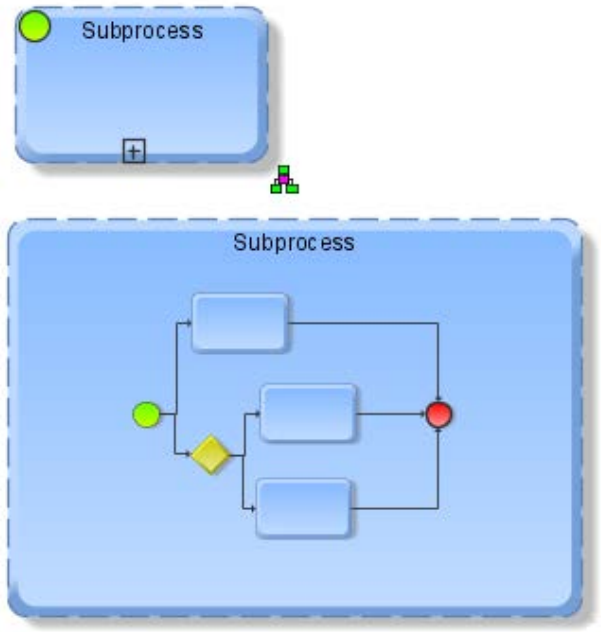

Figure 11: Symbols of an event subprocess

There is a Boolean ARIS attribute type **Event subprocess** representing the BPMN attribute **triggeredByEvent** (see Table 17). This read-only, but visible attribute type is used by the software.

4.1.5.3 Subprocess type: Transaction

A transaction subprocess, denoted with a double-lined boundary, is a specialized type of subprocess. In a transaction subprocess all activities must either complete successfully or the subprocess must be rolled back to its original consistent state. A transaction subprocess has a special behavior: It is associated with a transaction protocol that has to verify that all activities have been successfully completed. The symbols are not available in the symbol bar, they are handled by the software. The program also sets the value of the ARIS attribute type **Subprocess type** to **Transaction**.

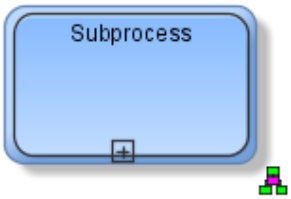

Figure 12: Symbol of a collapsed transaction

A transaction inherits from **Activity**. The attributes and model associations of a transaction subprocess and their mapping to ARIS constructs are shown in Table 18.

Class	BPMN attribute name	ARIS implementation
Transaction	*inherits from Activity*	The value of the attribute type **Subprocess type** (AT_BPMN_SUBPROCESS_TYPE) is set to **Transaction** in the attribute type group **BPMN 2.0 attributes / Subprocess attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbols: * Transaction (ST_BPMN_TRANSACTION) * Transaction (collapsed) (ST_BPMN_TRANSACTION_COLLAPSED_1) The symbols are rendered by the software.
	protocol: string [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
	method: TransactionMethod = compensate { compensate | store | image }	Not considered in the 1st version of the BPMN 2.0 implementation.

Table 18: Transaction: Mapping the attributes and model associations to ARIS

4.1.5.4 Subprocess type: Adhoc subprocess

An adhoc subprocess, denoted with a tilde marker, is a specialized type of subprocess. It contains a set of activities that could be performed. Sequence flow between activities is

optional in an Adhoc subprocess. What activities are performed as well as the sequence and the number of performances is determined by the performers of the activities. During execution of the (parent) process, any one or more of the activities may be active.

The ARIS method provides the tilde marker as mini-symbol of the value **Adhoc subprocess** of the attribute **Subprocess type** (see Figure 13). The program will render the symbols of the adhoc subprocess.

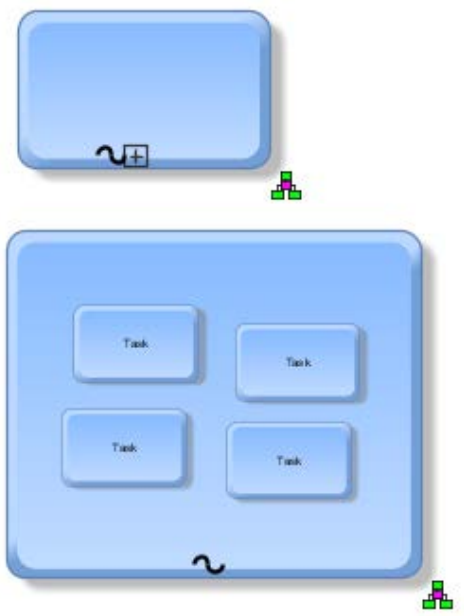

Figure 13: Symbol of a collapsed and expanded adhoc subprocess

The attributes and model associations of an adhoc subprocess and their mapping to ARIS constructs are shown in Table 19.

Class	BPMN attribute name	ARIS implementation
Adhoc sub-process	*inherits from Activity*	The value of the attribute type **Subprocess type** (AT_BPMN_SUB_PROCESS_TYPE) is set to **Adhoc subprocess** in the attribute type group **BPMN 2.0 attributes / Sub-process attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbols: The mini-symbol **tilde** will be rendered by the program.
	completionCondition: Expression	Attribute type **Adhoc completion condition** (AT_BPMN_COMPLETION_CONDI) in the attribute type group **BPMN 2.0 attributes / Subprocess attributes / Adhoc subprocess attributes** of object type **Function**.
	ordering: AdHocOrdering = parallel { parallel | sequential }	Not implemented in the 1st version of the BPMN 2.0 implementation.
	cancelRemainingInstances: Boolean = True	Not implemented in the 1st version of the BPMN 2.0 implementation.

Table 19: Adhoc subprocess: Mapping the attributes and model associations to ARIS

4.1.6 Activity type: Call Activity

A call activity represents the invocation of either a reusable global task or a process. The call activity represents the calling element, and the global task or process represents the called element.

The symbol **Call activity** is available in the symbol bar. If the modeler places this symbol, the value of the attribute **Activity type** is set to **Call activity** and the software provides a dialogue where the modeler selects the task or the process being called. Depending on this selection, the value of the attribute type **Called element** is set to **Global task** or **Global process**. BPMN Designer renders the symbol of the call activity. It corresponds to the symbol of the called task or process, but it is drawn with a thick border.

If a task is selected the program automatically creates a connection (call activity invokes task) on definition level. If a process is selected, the respective process diagram is assigned to the call activity.

The attributes and model associations of a call activity and their mapping to ARIS constructs are shown in Table 20.

Class	BPMN attribute name	ARIS implementation
CallActivity	*inherits from Activity*	The value of the attribute type **Activity type** (AT_BPMN_ACTIVITY_TYPE) is set to **Call activity** in the attribute type group **BPMN 2.0 attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: The symbol depends on the activity being called. The program will render the symbol.
	calledElement: CallableElement [0..1]	**For tasks:** The value of the attribute type **Called element** (AT_BPMN_CALLED_ELEMENT) is set to **Global task**. The BPMN Designer creates the connection type **Function invokes [CT_INVOKES] function** on definition level. **For processes:** The value of the attribute type **Called element** (AT_BPMN_CALLED_ELEMENT) is set to **Global process**. The BPMN process diagram of the called process will be assigned to the call activity. In both cases the software provides an appropriate dialogue.

Table 20: Call activity: Mapping the attributes and model associations to ARIS

4.1.7 Global task

The global task is described in chapter 2.3.2.

A global task has no specific attributes and model associations, it inherits from callable element (see Table 3).

4.1.8 Loop characteristics

BPMN 2.0 provides two alternatives to model repeating activities (both tasks and subprocesses):

- Loop activity (= standard loop)
- Multi-instance activity

An activity can be specified to repeat based on a condition. That is called **standard loop** activity in BPMN. A standard loop is equivalent to the **do while** and **do until** structure in programming. The number of iterations is unknown.

A **multi-instance activity** is another type of repeating activity useful for performing actions on a list of items. A multi-instance activity is equivalent with a **for each** structure in programming. The number of iterations is known when the activity starts. It is the number of items in the list. Iterations of a multi-instance activity can be performed concurrently or sequentially.

The marker for a standard loop is a circular arrow at the bottom center of the activity symbol

(see

Figure 14).

Figure 14: Symbols of standard loop activities

The markers for multi-instance activities are three bars at the bottom center of the task or subprocess symbol.

- Vertical bars are used to represent concurrent/parallel performances (see Figure 15)
- Horizontal bars are used to represent sequential performances (see Figure 16)

Figure 15: Symbols of multi-instance parallel activities

Figure 16: Symbols of multi-instance sequential activities

Loop characteristics has no specific attributes, it inherits the attributes and associations of base element. The attribute type **Loop type** is used in ARIS to specify whether the loop is a standard loop, a multi-instance parallel loop, or a multi-instance sequential loop. The attribute values are visualized by mini-symbols (see Figure 14 - Figure 16).

4.1.8.1 Standard loop characteristics

The attributes and model associations of standard loop activities are summarized in Table 21.

4.1.8.2 Multi-instance loop characteristics

The attributes and model associations of multi-instance activities are shown in Table 21.

4.1.8.3 Complex behavior definition

The additional attributes and model associations of complex behavior definition are summarized in Table 21.

Class	BPMN attribute name	ARIS implementation
LoopCharacteristics	*inherits from BaseElement*	Attribute type group **Loop characteristics** (AT_BPMN_LOOP_CHARACTERISTICS) in attribute type group **BPMN 2.0 attributes** of object type **Function** (OT_FUNC).
StandardLoopCharacteristics	*inherits from BaseElement*	The value of the attribute type **Loop type** (AT_BPMN_LOOP_TYPE_2) is set to **Standard loop** (AVT_BPMN_STANDARD_LOOP) in the attribute type group **BPMN 2.0 attributes / Loop characteristics** of object type **Function**.
	testBefore: boolean = False	Attribute type **Test before** (AT_BPMN_LOOP_TEST_TIME) in the attribute type group **BPMN 2.0 attributes / Loop characteristics / Standard loop attributes** of object type **Function**.
	loopMaximum: Expression [0..1]	Attribute type **Loop maximum** (AT_BPMN_MAX_LOOP) in the attribute type group **BPMN 2.0 attributes / Loop characteristics / Standard loop attributes** of object type **Function**.
	loopCondition: Expression [0..1]	Attribute type **Loop condition** (AT_BPMN_LOOP_CONDITION) in the attribute type group **BPMN 2.0 attributes / Loop characteristics / Standard loop attributes** of object type **Function**.

Class	BPMN attribute name	ARIS implementation
MultiInstanceLoop Characteristics	*inherits from BaseElement*	The value of the attribute type **Loop type** (AT_BPMN_LOOP_TYPE_2) is set to **Multi-instance sequential loop** (AVT_BPMN_MULTI_INSTANCE_SEQUENTIAL_LOOP) or **Multi-instance parallel loop** (AVT_BPMN_MULTI_INSTANCE_PARALLEL_LOOP) in the attribute type group **BPMN 2.0 attributes/Loop characteristics** of object type **Function**.
	isSequential: boolean = False	isSequential = true corresponds to: Loop type = Multi-instance sequential loop isSequential = false corresponds to: Loop type = Multi-instance parallel loop
	loopCardinality: Expression [0..1]	Attribute type **Loop cardinality** (AT_BPMN_LOOP_CARDINALITY) in the attribute type group **BPMN 2.0 attributes/Loop characteristics/Multi-instance loop attributes** of object type **Function**.
	loopDataInput: DataInput [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
	loopDataOutput: DataOutput [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
	inputDataItem: Property [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
	outputDataItem: Property [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
	completionCondition: Expression [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
	behavior : MultiInstanceBehavior = all { none | one | all | complex }	Not considered in the 1st version of the BPMN 2.0 implementation.
	complexBehaviorDefinition: ComplexBehaviorDef	Not considered in the 1st version of the BPMN 2.0 implementation.

Class	BPMN attribute name	ARIS implementation
	inition [0..*]	
	oneBehaviorEventRef: EventDefinition [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
	noneBehaviorEventRef: EventDefinition [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
ComplexBehaviorDefinition	*inherits from BaseElement*	Not considered in the 1st version of the BPMN 2.0 implementation.
	condition: Formal Expression	
	event: ImplicitThrowEvent	

Table 21: Loop characteristics: Mapping the attributes and model associations to ARIS

4.2 Items and Data[29]

As mentioned above, the 1st implementation of BPMN 2.0 in ARIS focuses on the business process level. Therefore, only data objects and data stores are provided - as input or output of activities. Detailed data modeling aspects (e.g. data structures, data states, data associations) are omitted.

4.2.1 Data object

In BPMN 1.x data were considered as an artifact; in BPMN 2.0, data objects were upgraded to objects in the BPMN semantic model.

On the business level, where the data structures and mappings are not considered, data objects are represented in ARIS by six symbols of the object type **Cluster / data model**:

- Data Object
- Data Object Collection
- Data Input
- Data Input Collection
- Data Output
- Data Output Collection.

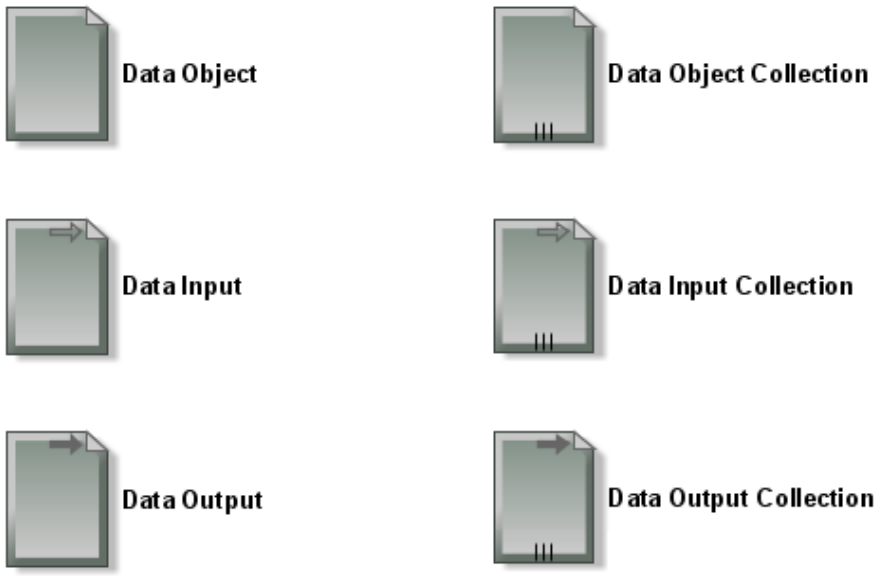

Figure 17: Symbols of data objects

[29] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 181-209..

Only one symbol is available in the symbol bar –the data object.

Data objects can represent input/output of activities by using the following connection types:

- Cluster/data model is input for function
- Function has output of cluster/data model

The data input symbols must not be the target of a **has output of** connection, and the data output symbols must not be the source of an **is input for** connection. The software ensures this.

A data object has the following attributes and model associations (see Table 22).

Class	BPMN attribute name	Implementation in ARIS
ItemAwareElement	*inherits from BaseElement*	
	itemSubjectRef: ItemDefinition [0..1]	The structure of the data is not considered in the 1st version of the BPMN 2.0 implementation.
	dataState: DataState [0..1]	Data states are not considered in the 1st version of the BPMN 2.0 implementation.
Data object	*inherits from FlowElement & ItemAwareElement*	Object type: Cluster/dta model (OT_CLST) 6 Symbols: * data object (ST_BPMN_DATA_OBJECT) * data collection (ST_BPMN_DATA_COLLECTION) * data input (ST_BPMN_DATA_INPUT) * data inpt collection (ST_BPMN_DATA_INPUT_COLELCTION) * data output (ST_BPMN_DATA_OUTPUT) * data output collection (ST_BPMN_DATA_OUTPUT_COLLECTION)
	isCollection: Boolean = False	Represented by special symbols of the object type **cluster/data model** (OT_CLST)

Table 22: Data object: Mapping the attributes and model associations to ARIS

4.2.2 Data store

Unlike data objects, which live only as long as the process instance is running, a data store represents information that persists beyond the lifetime of a particular process. On the business level, a data store is represented by the symbol **data store** of the ARIS object type **information carrier**. This symbol will be available in the symbol bar.

Data store can represent input/output of activities by using the following connection types:

- Information carrier provides input for function
- Function creates output to information carrier.

Figure 18: Symbol of a data store

A data store inherits from flow element and item-aware element. Its specific attributes and model associations are summarized in
Table 23.

Class	BPMN attribute name	Implementation in ARIS
DataStore	*inherits from FlowElement and ItemAwareElement*	Object type: Information carrier (OT_INFO_CARR) Symbol: Data store (ST_BPMN_DATA_STORE)
	name: string	Attribute type **Name** (AT_NAME) of object type information carrier (OT_INFO_CARR)
	capacity: Integer [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
	isUnlimited: Boolean = False	Not considered in the 1st version of the BPMN 2.0 implementation.
DataStoreReference	*inherits from FlowElement and ItemAwareElement*	
	dataStoreRef: DataStore	Occurrence copies of the (referenced) data store.

Table 23: Data store: Mapping the attributes and model associations to ARIS

4.3 Events

BPMN events are represented in ARIS by the object type **Event**. Altogether there are sixty-three symbols available in BPMN 2.0 (see Figure 19). The main types of event are:

- Start event,
- Intermediate event
- End event.

Only these three events are provided in the symbol bar in ARIS (see Figure 19: Type = None). The remaining sixty symbols will be provided as symbols in the ARIS method.

Types	Start			Intermediate				End
	Top-Level	Event Subprocess *Interrupting*	Event Subprocess *Non-Interrupting*	Catching	Boundary *Interrupting*	Boundary *Non-Interrupting*	Throwing	
None								
Message								
Timer								
Error								
Escalation								
Cancel								
Compensation								
Conditional								
Link								
Signal								
Terminate								
Multiple								
Parallel Multiple								

Figure 19: BPMN events: Summary[30]

[30] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 237.

4.3.1 Catch events and throw events

Events can be differentiated in:

- catch events (all start and some intermediate events)
- throw events (some intermediate events and all end events

The attributes and model associations of catch and throw events are listed in Table 24.

Class	BPMN attribute names	Implementation in ARIS
Event		Object type: Event (OT_EVT) Symbols: sixty-three different symbols (see below)
Catch Event	*inherits from FlowElement*	Object type: Event (OT_EVT) Symbols: different start or intermediate event symbols
	eventDefinitionRefs: EventDefinition [0..*]	Occurrence copy of the corresponding throw event.
	eventDefinitions: EventDefinition [0..*]	Attribute type **Event definition** (AT_BPMN_EVENT_DEFINITION) in the attribute type group **BPMN 2.0 attributes** of object type **Event** (OT_EVT). The values of this attribute type are: None, Message, Timer, Error, Escalation, Cancel, Compensation, Conditional, Link, Signal, Multiple, Parallel Multiple (as special case of Multiple). Each event definition has a specific marker inside the event symbol.
	dataOutputAssociations: DataOutputAssociation [0..*]	Not considered in the 1st version of the BPMN 2.0 implementation.
	dataOutput: dataOutput [0..*]	Connection type in the BPMN process diagram (BPMN 2.0) and the BPMN collaboration diagram (BPMN 2.0): Event (symbol: only catch events) has output of Cluster/data model
	outputSet: OutputSet [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
Throw event	*inherits from FlowElement*	Object type: Event (OT_EVT) Symbols: different intermediate or end event symbols
	eventDefinitionRefs:	Occurrence copy of the corresponding catch

Class	BPMN attribute names	Implementation in ARIS
	EventDefinition [0..*]	event.
	eventDefinitions: EventDefinition [0..*]	Attribute type **Event definition** (AT_BPMN_EVENT_DEFINITION) in the attribute type group **BPMN 2.0 attributes** of object type **Event** (OT_EVT). The values of this attribute type are: None, Message, Error, Escalation, Cancel, Compensation, Link, Signal, Terminate, Multiple. Each event definition has a specific marker inside the event symbol.
	dataInputAssociations: DataInputAssociation [0..*]	Not considered in the 1st version of the BPMN 2.0 implementation.
	dataInput: DataInput [0..*]	Connection type in the BPMN process diagram (BPMN 2.0) and the BPMN collaboration diagram (BPMN 2.0): Cluster/data model is input for event (symbol: only throw events)
	inputSet: InputSet [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
Implicit Throw Event	*inherits from ThrowEvent*	Not considered in the 1st version of the BPMN 2.0 implementation.

Table 24: Catch and throw event: Mapping the attributes and model associations to ARIS

4.3.2 Start event

The symbols of start events are depicted in Figure 19. Only the **none** start event is available in the symbol bar. When placing this start event, the modeler is guided by a special functionality of the BPMN Designer.

Their attributes and model associations of a start event are listed in Table 25.

Class	BPMN attribute names	Implementation in ARIS
Start event)	*inherits from CatchEvent*	Object type: Event (OT_EVT) Symbol: Start event (ST_BPMN_START_EVENT)
	isInterrupting: boolean	Interrupting start events are represented by specific event symbols.

Table 25: Start events: Mapping the attributes and model associations to ARIS

4.3.3 End event

The symbols of end events are depicted in Figure 19. Only the **none** end event is available in the symbol bar. When placing this end event, the modeler is guided by a special functionality of the program. End events have no specific attributes and associations.

Class	BPMN attribute names	Implementation in ARIS
EndEvent		Object type: Event (OT_EVT) Symbol: End event (ST_BPMN_END_EVENT)

Table 26: End event: Mapping to ARIS

4.3.4 Intermediate events

The symbols of intermediate events are depicted in Figure 19. Only the **none** intermediate event is available in the symbol bar. When placing this intermediate event, the modeler is guided by a special functionality of the program. Intermediate events have no specific attributes and associations.

Class	BPMN attribute names	Implementation in ARIS
IntermediateEvent		Object type: Event (OT_EVT) Symbol: Intermediate event (ST_BPMN_INTERMEDIATE_EVENT)

Table 27: Intermediate event: Mapping to ARIS

Some types of intermediate events can be attached to the boundary of activities, they are called **boundary events** (see column *Boundary Interrupting* and *Boundary Non-interrupting* in Figure 19). Boundary events are always catch events. Their attributes and model associations are shown in Table 28.

Class	BPMN attribute names	Implementation in ARIS
Boundary events	*inherits from CatchEvent*	Boundary events are specific intermediate events.
	AttachedTo: Activity	Connection type in the BPMN process diagram (BPMN 2.0) and the BPMN collaboration diagram (BPMN 2.0): Function can trigger (CT_BPMN_CAN_TRIGGER) event (symbol: **intermediate event**)
	CancelActivity: boolean	(Non-)Interrupting events are represented by specific event symbols.

Table 28: Intermediate event: Mapping to ARIS

4.3.5 Event definitions

BPMN 2.0 distinguishes the following event definitions: none, message, timer, error, escalation, cancel, compensation, conditional, link, signal, terminate and multiple (**parallel multiple** is a special case thereof). The different definitions are visualized by specific markers placed within the **none** start, intermediate and end event symbol (see Figure 19).

The attributes and model associations of the event definitions and their mapping to ARIS constructs are summarized in Table 29.

Class	BPMN attribute names	Implementation in ARIS
EventDefinition	*inherits from BaseElement*	Attribute type **Event definition** (AT_BPMN_EVENT_DEFINITION) in the attribute type group **BPMN 2.0 attributes** of object type Event (OT_EVT) Attribute values: None, message, timer, error, escalation, cancel, compensation, conditional, link, signal, terminate, multiple. This attribute is read-only, but visible and set automatically by the software.
CancelEventDefinition	*inherits from BaseElement*	Object type: Event (OT_EVT) Symbols: * Cancel intermediate event (ST_BPMN_CANCEL_INTERMEDIATE_EVENT) * Cancel end event (ST_BPMN_CANCEL_END_EVENT)
CompensationEvent Definition	*inherits from BaseElement*	Object type: Event (OT_EVT) Symbols: * Compensation start event (ST_BPMN_COMPENSATION_START * Compensation intermediate event (catch)(ST_BPMN_COMPENSATION_INTERMEDIATE_CATCH * Compensation intermediate event (throw) (ST_BPMN_COMPENSATION_INTERMEDIATE_THROW) * Compensation end event (ST_BPMN_COMPENSATION_END_EVENT)

Class	BPMN attribute names	Implementation in ARIS
	activityRef: Activity [0..1]	There is a catch compensation event (start or intermediate) which is an occurrence copy of the throw compensation event. This occurrence copy is used to find the activity to be compensated. (start event: the process the event sub-process is embedded in; Catch boundary intermediate event: the activity the event is attached to).
	waitForCompletion: boolean = True	Attribute type **Wait for completion** (AT_BPMN_WAIT_FOR_COMPLETION) in the attribute type group **BPMN 2.0 attributes/Compensation event attributes** of object type **Event** (OT_EVT).
ConditionalEvent Definition	*inherits from BaseElement*	Object type: Event (OT_EVT) Symbols: * Conditional start event (ST_BPMN_RULE_START_EVENT) * Conditional start event (non-interrupting) (ST_BPMN_CONDITIONAL_START_NI) * Conditional intermediate event (ST_BPMN_RULE_INTERMEDIATE_EVENT) * Conditional intermediate event (non-interrupting) (ST_BPMN_CONDITIONAL_INTERMEDIATE_NI)
	condition: Expression	Attribute type **Condition (natural language)** (AT_BPMN_RULE_EXPRESSION) in the attribute type group **BPMN 2.0 attributes/Conditional event attributes** of object type **Event** (OT_EVT).
ErrorEventDefinition	*inherits from BaseElement*	Object type: Event (OT_EVT) Symbols: * Error start event (ST_BPMN_ERROR_START) * Error intermediate event (ST_BPMN_ERROR_INTERMEDIATE_EVENT * Error end event (ST_BPMN_ERROR_END_EVENT)

Class	BPMN attribute names	Implementation in ARIS
	errorCode: string	Not implemented in the 1st version of the BPMN 2.0 implementation.
	error: Error [0..1]	Not implemented in the 1st version of the BPMN 2.0 implementation.
EscalationEvent Definition	*inherits from BaseElement*	Object type: Event (OT_EVT) Symbols: * Escalation start event (ST_BPMN_ESCALATION_START) * Escalation start event (non-interrupting) (ST_BPMN_ESCALATION_START_NI) * Escalation intermediate event (catch) (ST_BPMN_ESCALATION_INTERMEDIATE_CATCH) * Escalation intermediate event (non-interrupting) (ST_BPMN_ESCALATION_INTERMEDIATE_NI) * Escalation intermediate event_throw (ST_BPMN_ESCALATION_INTERMEDIATE_THROW) * Escalation end event (ST_BPMN_ESCALATION_END)
	escalationCode: string	Not considered in the 1st version of the BPMN 2.0 implementation.
	escalationRef: Escalation [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
LinkEventDefinition	*inherits from BaseElement*	Object type: Event (OT_EVT) Symbols: * Link intermediate event (catch) (ST_BPMN_LINK_INTERMDIATE_CATCH) * Link intermediate event (throw) (ST_BPMN_LINK_INTERMDIATE_THROW) Catch and throw link events are referred to each other by occurrence copies.
	name: string	Attribute type **Name** (AT_NAME) of object type **Event** (OT_EVT)

Class	BPMN attribute names	Implementation in ARIS
MessageEvent Definition	*inherits from BaseElement*	Object type: Event (OT_EVT) Symbols: * Message start event (ST_BPMN_MESSAGE_START_EVENT) * Message start event (non-interrupting) (ST_BPMN_MESSAGE_START_NI) * Message intermediate event (catch) (ST_BPMN_MESSAGE_INTERMEDIATE_CATCH) * Message intermediate event (non-interrupting) (ST_BPMN_MESSAGE_INTERMEDIATE_NI) * Message intermediate event (throw) (ST_BPMN_MESSAGE_INTERMEDIATE_THROW) * Message end event (ST_BPMN_MESSAGE_END_EVENT)
	MessageRef: Message [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
	operationRef: Operation [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
Multiple event		Object type: Event (OT_EVT) Symbols: * Multiple start event (ST_BPMN_MULTIPLE_START_EVENT) * Multiple start event (non-interrupting) (ST_BPMN_MULTIPLE_START_NI) * Multiple intermediate event (catch) (ST_BPMN_MULTIPLE_INTERMEDIATE_CATCH) * Multiple intermediate event (non-interrupting)(ST_BPMN_MULTIPLE_INTERMEDIATE_NI) * Multiple intermediate event (throw) (ST_BPMN_MULTIPLE_INTERMEDIATE_THROW) * Multiple end event (ST_BPMN_MULTIPLE_END_EVENT)

Class	BPMN attribute names	Implementation in ARIS
None event		Object type: Event (OT_EVT) Symbols: * Start event (ST_BPMN_SE) * Intermediate event (ST_BPMN_IE) * End event (ST_BPMN_EE) These symbols are available in the symbol bar.
Parallel multiple event		Object type: Event (OT_EVT) Symbols: * Parallel multiple start event (ST_BPMN_PARALLEL_MULTIPLE_START) * Parallel multiple start event (non-interrupting) (ST_BPMN_PARALLEL_MULTIPLE_START_NI) * Parallel multiple intermediate event (ST_BPMN_PARALLEL_MULTIPLE_INTERMEDIATE) * Parallel multiple intermediate event (non-interrupting) (ST_BPMN_PARALLEL_MULTIPLE_INTERMEDIATE_NI)
SignalEventDefinition	*inherits from BaseElement*	Object type: Event (OT_EVT) Symbols: * Signal start event (ST_BPMN_SIGNAL_START_EVENT) * Signal start event (non-interrupting) (ST_BPMN_SIGNAL_START_NI) * Signal intermediate event (catch) (ST_BPMN_SIGNAL_INTERMEDIATE_EVENT) * Signal intermediate event (non-interrupting) (ST_BPMN_SIGNAL_INTERMEDIATE_NI) * Signal intermediate event (throw) (ST_BPMN_SIGNAL_INTERMEDIATE_THROW) * Signal end event (ST_BPMN_SIGNAL_END_EVENT)

Class	BPMN attribute names	Implementation in ARIS
	signalRef: Signal	Not considered in the 1st version of the BPMN 2.0 implementation.
TerminateEvent Definitin	*inherits from BaseElement*	Object type: Event (OT_EVT) Symbol: * Terminate end event (ST_BPMN_TERMINATE_END_EVENT)
TimerEventDefini tion	*inherits from BaseElement*	Object type: Event (OT_EVT) Symbols: * Timer start event (ST_BPMN_TIMER_START_EVENT) * Timer start event (non-interrupting) (ST_BPMN_TIMER_START_NI) * Timer intermediate event (ST_BPMN_TIMER_INTERMEDIATE_EVENT) * Timer intermediate event (non-interrupting) (ST_BPMN_TIMER_INTERMEDIATE_NI).
	timeDate: Expression [0..1]	Attribute type **Time date** (AT_BPMN_TIMEDATE) in the attribute type group **BPMN 2.0 attributes/Timer event attributes** of object type **Event** (OT_EVT).
	timeCycle: Expression [0..1]	Attribute type **Time cycle** (AT_BPMN_TIMECYCLE) in the attribute type group **BPMN 2.0 attributes/Timer event attributes** of object type **Event** (OT_EVT).

Table 29: Event types: Mapping the attributes and model associations to ARIS

4.4 Gateways

The ARIS object type **Rule** depicts BPMN gateways. Although BPMN 2.0 knows five different gateway types, only one symbol will be available in the symbol bar:

The remaining gateway symbols will be handled by the BPMN Designer. Figure 20 depicts all (basic) gateway symbols.

Figure 20: BPMN gateway types

For event-based gateways there will be two additional symbols which are used to start a process:

Instantiating event-based gateway:

Instantiating parallel event-based gateway:

All in all the ARIS method will provide eight gateway symbols. Contrary to events, an ARIS attribute recording the gateway type is not required. It is up to the modeler to ensure that gateways are used in a semantically correct way. The modeler should not reuse gateways.

4.4.1 Exclusive gateway

The attributes and model associations of exclusive gateways and their mapping to ARIS constructs are shown in Table 30.

Class	BPMN attribute name	Implementation in ARIS
Exclusive gateway	*inherits from Gateway*	Object type: Rule (OT_RULE) Symbols: * Gateway (ST_BPMN_RULE_1) * Exclusive gateway (ST_BPMN_RULE_XOR_3) * Event-based gateway (ST_BPMN_RULE_XOR_4)
	default: SequenceFlow [0..1]	Attribute type **Sequence flow condition** (AT_BPMN_SEQ_FLOW_CONDITION) in the attribute type group **BPMN 2.0 attributes** of the following connection types: * Rule leads to (CT_LEADSTO_2) event * Rule activates (CT_ACTIV_1) function * Rule links (CT_LNK_2) rule The attribute value must be set to **Default sequence flow**. The symbol (slash) will be automatically set by the software.

Table 30: Exclusive gateway: Mapping the attributes and model associations to ARIS

4.4.2 Inclusive gateway

The attributes and model associations of inclusive gateways and their mapping to ARIS constructs are shown in Table 31.

Class	BPMN attribute name	Implementation in ARIS
Inclusive gateway	*inherits from Gateway*	Object type: Rule (OT_RULE) Symbol: * Inclusive gateway (ST_BPMN_RULE_OR_1)
	default: SequenceFlow [0..1]	See: exclusive gateway

Table 31: Inclusive gateway: Mapping the attributes and model associations to ARIS

4.4.3 Parallel gateway

Parallel gateways have no specific attributes.

Class	BPMN attribute name	Implementation in ARIS
Parallel gateway	*inherits from Gateway*	Object type: Rule (OT_RULE) Symbol: * Parallel gateway (ST_BPMN_RULE_AND_1)

Table 32: Parallel gateway: Mapping to ARIS

4.4.4 Complex gateway

The attributes and model associations of complex gateways and their mapping to ARIS constructs are shown in Table 33.

Class	BPMN attribute name	Implementation in ARIS
Complex gateway	*inherits from Gateway*	Object type: Rule (OT_RULE) Symbol: * Complex gateway (ST_BPMN_RULE_COMPLEX_1)
	activationCondition: Expression [0..1]	Attribute type **Activation condition** (AT_ACTIVATION_CONDITION) in the attribute type group **BPMN 2.0 attributes/Complex gateway attributes** of object type Rule (OT_RULE)

Table 33: Inclusive gateway: Mapping the attributes and model associations to ARIS

4.4.5 Event-based gateways

The attributes and model associations of complex gateways and their mapping to ARIS constructs are shown in Table 34. All attributes are represented by specific gateway symbols.

Class	BPMN attribute name	Implementation in ARIS
Event-based gateway	*inherits from Gateway*	Object type: Rule (OT_RULE) Symbols: * Event-based gateway (ST_BPMN_RULE_XOR_4) * Instantiating event-based gateway (ST_BPMN_RULE_XOR_START) * Instantiating parallel event-based gateway (ST_BPMN_RULE_XOR_PARALLEL)
	instantiate: boolean = False	Represented by symbols. **True**, if: * Instantiating event-based gateway (ST_BPMN_RULE_XOR_START) * Instantiating parallel event-based gateway (ST_BPMN_RULE_XOR_PARALLEL) **False** if: * Event-based gateway (ST_BPMN_RULE_XOR_4)
	eventGatewayType: EventGatewayType = Exclusive { Exclusive | Parallel }	Represented by symbols: **Exclusive** if: * Event-based gateway (ST_BPMN_RULE_XOR_4) * Instantiating event-based gateway (ST_BPMN_RULE_XOR_START) **Parallel** if: * Instantiating parallel event-based gateway (ST_BPMN_RULE_XOR_PARALLEL)

Table 34: Event-based gateway: Mapping the attributes and model associations to ARIS

4.5 Lanes[31]

A lane is a subdivision of a process or a pool. Lanes have no semantics in BPMN. BPMN 2.0 uses lanes as a way to categorizes flow elements. Most often lanes represent organizational elements, but in principle any categorization may be used for lanes. Lanes may contain nested sub-lanes. A lane set specifies the categorization represented by the lanes.

Like a pool a lane is drawn as a rectangular box, its label is not boxed off.

Figure 21: Nested lanes

The attributes and model associations of a lane are summarized in Table 35.

Class	BPMN attribute name	Implementation in ARIS
LaneSet	*inherits from BaseElement*	Object type: Lane (OT_BPMN_LANE) Symbol: Lane (ST_BPMN_LANE_1)
	process: Process	The BPMN process model that contains the lane(s).
	lanes: Lane [0..*]	Object type: Lane (OT_BPMN_LANE) Symbol: Lane (ST_BPMN_LANE_1) The source objects in the connection type: Lane belongs to (CT_BELONGS_TO_1) lane
	parentLane: Lane [0..1]	The target object in the connection type: Lane belongs to (CT_BELONGS_TO_1) lane CT: Lane belongs to lane
Lane	*inherits from BaseElement*	Object type: Lane (OT_BPMN_LANE) Symbol: Lane (ST_BPMN_LANE_1)
	name: string	Attribute type **Name** (AT_NAME) of object type Lane (OT_BPMN_LANE)
	partitionElement: BaseElement [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
	partitionElementRef: BaseElement [0..1]	Not considered in the 1st version of the BPMN 2.0 implementation.
	childLaneSet: LaneSet [0..1]	The source objects in the connection type: Lane belongs to (CT_BELONGS_TO_1) lane

[31] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 282-286.

Class	BPMN attribute name	Implementation in ARIS
	flowElementRefs: FlowElement [0..*]	The source objects in the following **belongs to** (CT_BELONGS_TO_1) connection types: * Function belongs to lane * Event belongs to lane * Rule belongs to lane * Cluster/data model belongs to lane * Information carrier belongs to lane

Table 35: Lane set and lane: Mapping the attributes and model associations to ARIS

5 Collaboration[32]

A collaboration shows message exchanges between participants. A collaboration contains at least two pools representing the participants. A pool may include a process (white box) or may be shown as a black box with all details hidden. The message exchanges between the participants are represented by message flows that connect two pools (or the objects within the pools). Only one pool may be represented without a boundary.

The new model type **BPMN collaboration diagram (BPMN 2.0)** has been introduced to model collaborations.

The attributes and model associations of a collaboration and their mapping to ARIS is listed in Table 36.

Class	BPMN attribute names	Implementation in ARIS
Collaboration	*inherits from BaseElement and InteractionSpecification*	Model type: BPMN collaboration diagram (BPMN 2.0) (MT_BPMN_COLLABORATION_DIAGRAM)
	name: string	Attribute type **Name** of the BPMN collaboration diagram (BPMN 2.0)
	choreographyRef: Choreography [0..1]	Choreography is not considered in the 1st version of the BPMN 2.0 implementation.
	conversationAssociations: ConversationAssociation [0..*]	The relationships to conversations are represented by occurrence copies of participants (OT_BPMN_POOL; ST_BPMN_POOL_1), occurrence copies of message flow connections and the assignment of a BPMN collaboration model (BPMN 2.0) to the object type **Conversation** (OT_BPMN_CONVERSATION).
	conversations: Conversation [0..*]	BPMN collaboration diagram (BPMN 2.0) assigned to the object type **Conversation** (OT_BPMN_CONVERSATION)
	artifacts: Artifact [0..*]	See chapter 2.3.1
	participantAssociations: ParticipantAssociations [0..*]	The relationships to participants are represented by occurrence copies of participants (OT_BPMN_POOL; ST_BPMN_POOL_1),

[32] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 111-119.

Class	BPMN attribute names	Implementation in ARIS
	messageFlowAssociations: Message Flow Association [0..*]	The relationships to message flows are represented by occurrence copies of message flow connections (and the involved participants).
	IsClosed: boolean = false	Attribute type **Is closed** in the attribute type group **BPMN 2.0 attributes** of model type the BPMN Collaboration diagram (BPMN 2.0).

Table 36: Collaboration

The object types and connection types of the BPMN collaboration diagram are detailed in the following chapters.

5.1 Pool and participant

Pools and participants play a central role in collaborations. They are described in detail in chapter 2.3.14.

5.2 Object types and connection types reused from a process

As a pool may show a process (white box) all object types and connection types that are allowed in the BPMN process diagram (BPMN 2.0) are also available in the BPMN collaboration diagram (BPMN 2.0).

The object types and connection types taken over from the BPMN process diagram (BPMN 2.0) are described in detail in the chapter 4.

The connection type **belongs to** is used to embed the object types of a visible process into a pool.

Source object type	Connection type	Target object type
Event (OT_EVT)	Belongs to (CT_BELONGS_TO_1)	Participant (OT_BPMN_POOL)
Function (OT_FUNC)	Belongs to (CT_BELONGS_TO_1)	Participant (OT_BPMN_POOL)
Rule (OT_RULE)	Belongs to (CT_BELONGS_TO_1)	Participant (OT_BPMN_POOL)
Lane (OT_BPMN_LANE)	Belongs to (CT_BELONGS_TO_1)	Participant (OT_BPMN_POOL)
Cluster/data model (OT_CLST)	Belongs to (CT_BELONGS_TO_1)	Participant (OT_BPMN_POOL)
Information carrier (OT_INFO_CARR)	Belongs to (CT_BELONGS_TO_1)	Participant (OT_BPMN_POOL)

Table 37: Association of process object types to pools

5.3 Message flow

The message flow between different participants is represented by an ARIS connection type of the same name. It connects two pools or the objects within a pool. The attributes and model associations of message flow are described in chapter 2.3.13.

To show the messages being exchanged in message flows the ARIS object type **message** represented by a message symbol is used. The message flow connection type is replaced by two connection types: **sends** and **is received from**. This work around is required due to the fact that it is not possible in ARIS to assign object types to connection types.

The BPMN Designer will display the **sends** and **is received from** connection types like a normal message flow (see Figure 5). Details can be found in chapter 2.3.13.

6 Conversation[33]

The conversation diagram has been introduced with BPMN 2.0 to provide a big picture of the interactions (in terms of related message exchanges) between collaborating participants.

The conversation diagram is similar to the collaboration diagram, but its pools are not allowed to contain a process a choreography is not allowed between the pools.

The BPMN conversation diagram differentiates three basic elements.

- conversation nodes
- participants
- conversation links.

They are described in the next chapters.

6.1 Conversation container

The attributes and model associations of a conversation and conversation container are

Class	Attributes & Associations	Implementation in ARIS
Conversation	*inherits from CallableElement, InteractionSpecification, ConversationContainer*	
	correlationKeys: CorrelationKey [0..*]	Correlation keys are not considered in the 1st version of the BPMN 2.0 implementation.
	messageFlowRefs: MessageFlow [0..*]	Occurrence copies of message flows (and the involved participants).
Conversation Container	*inherits from BaseElement*	Model type: BPMN conversation diagram (BPMN 2.0) MT_BPMN_CONVERSATION_DIAGRAM
	conversationNodes: ConversationNode [0..*]	see below
	artifacts: Artifact [0..*]	see chapter 2.3.1

summarized in Table 38.

Table 39: Conversation: Mapping the attributes and model associations to ARIS

6.2 Conversation nodes

BPMN 2.0 distinguishes three sub-types of conversation nodes:

- Communication
- Sub-conversation
- Call conversation[34]

[33] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 293-305.

Conversation nodes are represented in ARIS by the new object type **Conversation**.

A **communication** is an atomic conversation element in a BPMN conversation diagram, it represents a set of message flow grouped together based on a single correlation key. A communication will involve at least two participants.

The symbol of a communication (see Figure 22) is available in the symbol bar of the BPMN conversation diagram (BPMN 2.0).

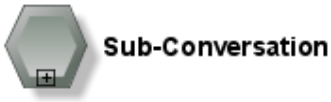

Figure 22: Symbols of conversation nodes

A **sub-conversation** is a conversation which consists of lower-level conversations[35] which are modeled in a separate BPMN conversation diagram assigned to the sub-conversation. A sub-conversation shares the participants of its parent conversation.

The ARIS method provides a sub-conversation symbol (see Figure 22), it also shown in the symbol bar.

A **call conversation** identifies a place in a conversation where a global conversation or a global communication is used. A **global communication** is a reusable atomic communication definition that can be called from within any conversation by a call conversation.

The concepts of call conversations and global communication are very vague. Thus, the ARIS method does not provide specific symbols. But, there is an ARIS attribute type which allows the modeler to flag call conversations:

Call conversation

This is a Boolean attribute. If the value is **true**, the call conversation symbol will be rendered by automatically by the software.

6.3 Participant

Participants are represented by the ARIS object type **Participant**. The **Pool** symbol is available in the symbol bar. If the ARIS attribute type **Multi-instance participant** is set to **true** the program will render the symbol: three vertical lines will be displayed at the bottom of the pool symbol.

Participants/pools are described in detail in chapter 2.3.14.

---

[34] The concept of call conversation is not clear, thus, call conversations are ignored in the 1st iimplementation of BPMN 2.0 in ARIS.

[35] A sub-conversation can be compared with a subprocess in a BPMN process diagram.

6.4 Artifacts

According to the metamodel artifacts are allowed in a conversation diagram. However, the relevance of groupings in a conversation diagram is not evident. For that reason only text annotations will be implemented in the 1st version of the BPMN conversation diagram.

The symbol **Text annotation** will be available in the symbol bar. Artifacts and their usage are described in detail in chapter 2.3.1.

6.5 Conversation link

A conversation link[36] is used to link participants with conversation nodes. A conversation node has at least two participants.
N-ary (n > 2) conversations are allowed.

In ARIS a new connection type:

- Participant participates in conversation has been introduced. The passive name of the connection type is:
- Conversation has conversation link to participant. Specific attributes are not required.

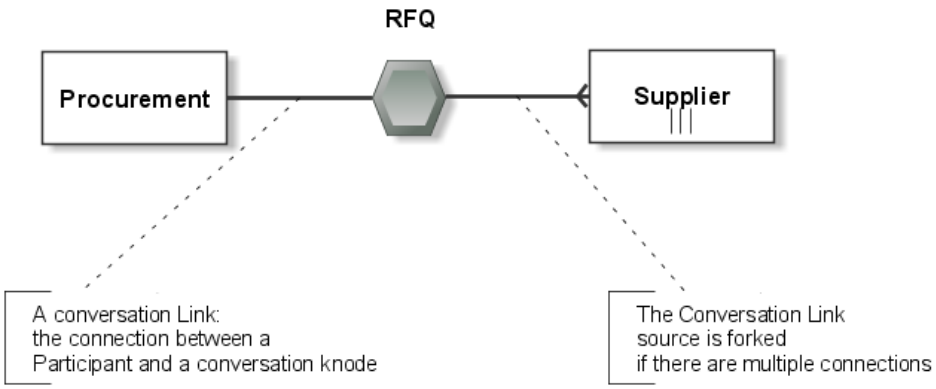

Figure 23: Conversation link with **participant multiplicity**

The fork shown at the source of a conversation link (see Figure 23) must be manually set by the modeler using the property dialogue of the respective connection type.

[36] There is an inconsistency in the specification: Sometimes the name “Communication links” is used, sometimes the name “Conversation link”.

6.6 Message flow in a conversation

According to the specification, it is allowed in the BPMN conversation diagram to model message exchanges between participants using message flows.[37]

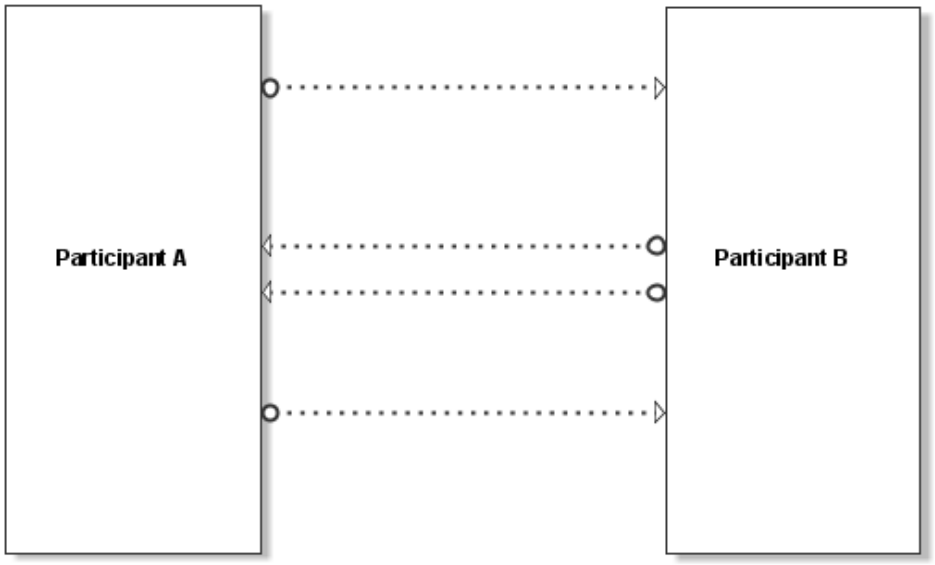

Figure 24: Message flow between participants in a conversation diagram

Thus, the ARIS connection type **Participant message flow participant** will be available in the BPMN conversation diagram.

6.7 Model assignments

The object type “Conversation” will have the following assignments:

- BPMN conversation diagram
- BPMN collaboration diagram.

Only one model of each type can be assigned to a conversation object.

[37] See: Business Process Modeling Notation (BPMN), FTF Beta 1 for version 2.0. Page 294.