BAMBERGER BEITRÄGE

ZUR WIRTSCHAFTSINFORMATIK UND ANGEWANDTEN INFORMATIK

ISSN 0937-3349

Nr. 92

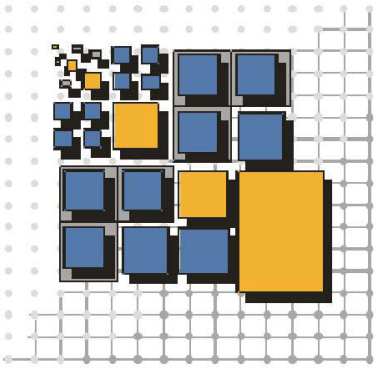
BPMN 2.0 Process Model Serialization Constraints

Matthias Geiger

May 2013

FAKULTÄT WIRTSCHAFTSINFORMATIK UND ANGEWANDTE INFORMATIK

OTTO-FRIEDRICH-UNIVERSITÄT BAMBERG



BPMN 2.0 Process Model Serialization Constraints

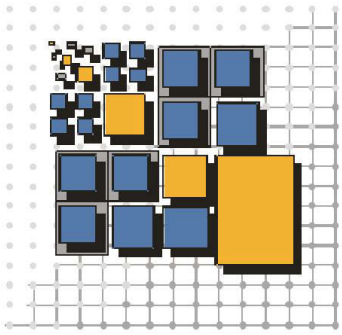
Matthias Geiger

Lehrstuhl für Praktische Informatik, Fakultät WIAI
Otto-Friedrich-Universität Bamberg
An der Weberei 5, 96047 Bamberg

`http://www.uni-bamberg.de/pi/bpmn-constraints`

Abstract Correct and standard compliant serializations of BPMN process models are crucial for model exchange between tools, automatic application of academic verification approaches and executability on BPMN engines. The official standard document does not provide an extensive set of all constraints regarding the correctness of model serializations. This technical reports fills this gap by presenting a categorized list of generic, technology independent constraints stated by the standard. Furthermore it is analyzed which rules are already covered when the standardized XSD-based serialization format is used.

Keywords BPMN 2.0, Serialization, Standard analysis, Standard compliance, Constraints



Distributed Systems Group

Otto-Friedrich Universität Bamberg

Prof. Dr. rer. nat. Guido Wirtz

http://www.uni-bamberg.de/pi/

Due to hardware developments, strong application needs and the overwhelming influence of the net in almost all areas, distributed systems have become one of the most important topics for nowadays software industry. Owing to their ever increasing importance for everyday business, distributed systems have high requirements with respect to dependability, robustness and performance. Unfortunately, distribution adds its share to the problems of developing complex software systems. Heterogeneity in both, hardware and software, permanent changes, concurrency, distribution of components and the need for inter-operability between different systems complicate matters. Moreover, new technical aspects like resource management, load balancing and guaranteeing consistent operation in the presence of partial failures and deadlocks put an additional burden onto the developer.

The long-term common goal of our research efforts is the development, implementation and evaluation of methods helpful for the realization of robust and easy-to-use software for complex systems in general while putting a focus on the problems and issues regarding distributed systems on all levels. Our current research activities are focussed on different aspects centered around that theme:

- *Reliable and inter-operable Service-oriented Architectures*: Development of design methods, languages, tools and middle-ware to ease the development of SOAs with an emphasis on provable correct systems that allow for early design-evaluation due to rigorous development methods. Additionally, we work on approaches and standards to provide truly inter-operable platforms for SOAs.
- *Implementation of Business Processes and Business-to-Business-Integration (B2Bi)*: Starting from requirements for successful B2Bi development processes, languages and systems, we investigate the practicability and inter-operability of different approaches and platforms for the design and implementation of business processes with a focus on combining processes from different business partners.
- *Quality-of-Service (QoS) Aspects for SOA and B2Bi*: QoS aspects, especially reliability and security, are indispensable when putting distributed systems into practical use. We work on methods that allow for a seamless observance of QoS aspects during the entire development process from high-level business processes down to implementation platforms.
- *Agent and Multi-Agent (MAS) Technology:* Development of new approaches to use Multi-Agent-Systems for designing, organizing and optimizing complex systems ranging from service management and SOA to electronic markets and virtual enterprises.
- *Visual Programming- and Design-Languages*: The goal of this long-term effort is the utilization of visual metaphors and languages as well as visualization techniques to make design- and programming languages more understandable and, hence, more easy-to-use.

More information about our work, i.e., projects, papers and software, is available at our homepage (see above). If you have any questions or suggestions regarding this report or our work in general, don't hesitate to contact me at guido.wirtz@uni-bamberg.de

Guido Wirtz

Bamberg, January 2010

Contents

1 Introduction **1**

2 Extracting Constraints from the Standard Document **3**

2.1 Extraction Approach 3

2.2 Assumptions and Limitations 5

3 Overview of All Extracted Constraints **6**

3.1 Basic Attribute/Sub Elements Cardinality 6

3.2 Basic Value Restrictions and Default Values 14

3.3 Basic Reference Constraints 16

3.4 Extended Constraints 20

3.4.1 Constraint Structure and Important Attributes 20

3.4.2 Tabular Overview 21

4 Analysis of the XSD-based Serialization **27**

4.1 Basic Attribute/Sub Elements Cardinality 27

4.2 Basic Value Restrictions and Default Values 30

4.3 Basic Reference Constraints 31

4.4 Extended Constraints 32

5 Discussion and Comparison to Another Rule Collection **34**

6 Summary **36**

Bibliography **38**

A Detailed Description of All Extended Constraints **40**

B List of previous University of Bamberg reports **103**

List of Figures

1 Element *definitions*; excerpt from the BPMN standard 4

2 *Association* attribute definition 5

List of Tables

1 Overview: Basic Attributes and Sub Elements and their Cardinality 13

2 Overview: Basic Value Restrictions and Default Values 15

3 Overview: Basic Reference Constraints 19

4 Example to illustrate the extensive description of all Extended Constraints . . . 20

5 Overview: Extended and complex constraints 26

6 Mapping of Silver's rules to the corresponding Constraint Numbers in this work 35

List of Acronyms

BPMN	Business Process Model and Notation
OMG	Object Management Group
WfMC	Workflow Management Coalition
WS-BPEL	Web Services Business Process Execution Language
XMI	XML Metadata Interchange
XML	eXtensible Markup Language
XPDL	XML Process Definition Language
XSD	XML Schema Definition

1 Introduction

The Business Process Model and Notation (BPMN) [10] as a standard for a (business) process notation is widely used in practice and academia. In practice BPMN is used for modeling purposes such as visualizing and documenting business processes in most cases. But also BPMN engines for executing BPMN models[1] are gaining more and more importance.

Especially for saving and exchanging models between different tools and engines, for process execution and for applying scientific approaches for checking the semantics (such as [3–5, 14, 20]) automatically a serialized form of BPMN models is needed. Unfortunately BPMN versions prior to Version 2.0 ([1, 8, 9]) the focus was to define a graphical set of shapes for process modeling. A consistent meta-model and a serialiazion format was left out completely which led to a plethora of different proposals to serialize BPMN models.

Besides proprietary formats two main serialization formats were used for BPMN 1.1 [8] and BPMN 1.2 [9]:
For executable processes often a serialization in the Web Services Business Process Execution Language (WS-BPEL)[7] was proposed and analyzed. Due to the different paradigms (block-structured vs. graph-based) and the missing ability to save the graphical information of BPMN models in WS-BPEL a roundtripping between BPMN and WS-BPEL is hard and not always possible ([6, 12, 13, 15, 19]).

The second main serialization format which is better aligned with BPMN is the XML Process Definition Language (XPDL)[21, 22] which is still promoted and developed further by the Workflow Management Coalition (WfMC)[2]. Various tools still support XPDL as an export and import format for BPMN process models.

In [2] a XML-based serialization is proposed. The authors furthermore provide checking mechanisms based on XPath and XQuery to check for rule violations in the models. Unfortunately the approach is outdated by now as a new version of BPMN [10] is available.

With Version 2.0 two serialization mechanisms based on the eXtensible Markup Language (XML) have been added and are therefore standardized: XML Metadata Interchange (XMI)[11] and XML Schema Definition (XSD)[17, 18] (see Chap. 15 in [10]). The default serialization format which is linked directly with and often referred to in [10] is the XSD-based XML serialization.

As mentioned a BPMN serialization may be realized in different ways, but all variants must respect the stated rules in the standard document to generate and save standard compliant BPMN models.

The standard document [10] consists of about 480 pages running text, hundreds of figures and tables and various appendices. Furthermore some *normative* files are also referred to. All these sources contain guidelines, restrictions and constraints to define requirements for *correct* BPMN models. These requirements range from graphical constraints regarding the appearance

[1] such as Activiti (`http://activiti.org`)
[2] `http://www.wfmc.org`

of BPMN shapes, via constraints regarding the serialization of models to constraints which are only relevant during process execution on BPMN engines.

All these rules must be fulfilled and respected if *i)* a specific BPMN model should be valid or *ii)* a tool or an engine claims to be BPMN compliant. Inexplicably the standardization organization Object Management Group (OMG)[3] neither provides a complete list of all stated BPMN constraints nor tools or tests to check correctness and compliance of models and tools. Therefore it is currently hard to decide whether a concrete model or a "BPMN compliant" editor in fact conforms to all restrictions stated in [10].

The aim of our work is to provide an extensive and (hopefully) complete list of all serialization constraints defined in the standard document [10]. The revealed rules are generic and independent from the used serialization format. This rule set is a generic basis and the first step to develop specific compliance checking mechanisms for BPMN process models.

In this report we focus on serialization constraints and therefore the list does not cover other BPMN aspects such as graphical constraints (e.g., thickness of shape borders) and execution semantics rules. More insight in the assumptions and limitations as well as a description of our approach of detecting and extracting constraints is presented in the following section 2.

The results of the application, that is an overview over all detected constraints is given in the subsequent section 3. In section 4 "Analysis of the XSD-based Serialization" it is analyzed which rules are already covered by the usage of the standardized XSD-based exchange format. A discussion of our results and a comparison to existing constraint collections is presented in section 5 and the last section 6 concludes our work with a summary and an overview of planned future work.

[3] `http://www.omg.org`

2 Extracting Constraints from the Standard Document

Before presenting the main contribution of our work namely an overview over all extracted constraints (see section 3) it is important to clarify our extraction approach, how the extracted rules can be categorized and all assumptions and limitations of this approach.

2.1 Extraction Approach

Regarding model constraints the standard document consists of three important sources: The running text, class diagrams and tables specifying attributes of BPMN elements.

When analyzing the standard documents four main constraint categories have been identified:

- **Basic Attribute/Sub Element Cardinality (`CARD`):** This category consists of all constraints regarding cardinality of attributes and sub elements of BPMN elements, i.e., which attributes are allowed for all BPMN elements and how often are they allowed or required to occur. Cardinality constraints are mostly denoted in attribute tables in [10] using the standard notation in square brackets (e.g., `[1...*]`). Of peculiar interest are mandatory attributes/elements and minimum/maximum occurrence constraints.
- **Basic Value Restrictions and Default Values (`VAL`):** For various attributes/elements default values and restrictions of allowed values are stated in [10].
- **Basic Reference Constraints (`REF`):** Constraints regarding references to other BPMN Elements: When a BPMN element references another element, the reference must be resolvable, i.e., the referenced element must exist. Moreover, in most cases only specific elements are allowed to be referenced. Such constraints are also categorized in this category.
- **Extended Constraints (`EXT`):** All other constraints which could not be assigned to the other basic categories will be categorized in this category. Especially constraints which depend on concrete values for other elements and attributes fall in this category.

Depending on the category the extraction approach is different:

Cardinality (`CARD`) and value restrictions (`VAL`) can be determined by simply analyzing the tables in [10] which define the *"the attributes and model associations"* of all BPMN elements. These tables contain all needed information such as attribute names, type definitions, value and cardinality restrictions.

Figure 1 shows an excerpt of Table 8.1 from [10, p.83] which introduces the attributes and their constraints for the BPMN base element *definitions*. The cardinality constraints CARD.001-CARD.006 (see Sec. 3.1) can be derived from this short extraction: The list of allowed attributes is determined by the column "Attribute Name" which shows the allowed attributes named *name*,

Table 8.1 - Definitions attributes and model associations

Attribute Name	Description/Usage
name: string	The name of the `Definition`.
targetNamespace: string	This attribute identifies the namespace associated with the `Definition` and follows the convention established by XML Schema.
expressionLanguage: string [0..1]	This attribute identifies the formal `Expression` language used in `Expressions` within the elements of this `Definition`. The Default is "http://www.w3.org/1999/XPath". This value MAY be overridden on each individual formal `Expression`. The language MUST be specified in a URI format.
typeLanguage: string [0..1]	This attribute identifies the type system used by the elements of this `Definition`. Defaults to http://www.w3.org/2001/XMLSchema. This value can be overridden on each individual `ItemDefinition`. The language MUST be specified in a URI format.
rootElements: RootElement [0..*]	This attribute lists the root elements that are at the root of this `Definitions`. These elements can be referenced within this `Definitions` and are visible to other `Definitions`.
diagrams: BPMNDiagram [0..*]	This attribute lists the `BPMNDiagrams` that are contained within this `Definitions` (see page 367 for more information on

Figure 1: Element *definitions*; excerpt from the BPMN standard ([10, p.83])

targetNamespace, *expressionLanguage*, *typeLanguage*, *rootElements* and *diagrams* (**bold** label followed by colon).

The colon is followed by a denotion of the allowed datatype for each attribute which is *"string"* for the former four attributes and *RootElement* resp. *BPMNDiagram* for the latter two. Moreover also the cardinality of each attribute can be derived from Fig. 1: Using the standard notation in square brackets the lowest and the highest number of occurrences is defined. For example, the attribute *expressionLanguage* must occur at most once $[0 \dots 1]$, whereas the number of rootElements is unlimited ($[0 \dots *]$). A missing cardinality information denotes the cardinality [1], that is the attribute must occur exactly once.

The value constraints VAL.001-002 are also defined in Table 8.1: For the attributes *expressionLanguage* and *typeLanguage* a default value is specified in the column "Description/Usage". Another way to define value define a default is shown in Figure 2: Using the equality sign the default value *"None"* is defined for the *association*'s mandatory element *associationDirection*. Moreover using the notation `{value 1 | value 2 | ...}` the set of allowed values is restricted to three specific strings: *"None"*, *"One"*, *"Both"*.

The same tables contain the reference type information which is essential for all reference rules (`REF`). References can be identified as attributes with the suffix *"...Ref"* and in the normative XSD files through the usage of the datatypes *xsd:IDREF* and *xsd:QName*. An Example for this can also be observed in Figure 2 namely the two attributes *sourceRef* and *targetRef* which both refer to a *BaseElement* definition.

Some extended constraints (`EXT`) also can be revealed by analyzing the tables just mentioned - especially in column *"Description/Usage"*. But it is also often the case that the running text states further restrictions and rules which have to be respected. Therefore the whole body of

Table 8.20 – Association attributes and model associations

Attributes	Description
associationDirection: AssociationDirection = None {None | One | Both}	`associationDirection` is an attribute that defines whether or not the Association shows any directionality with an arrowhead. The default is `None` (no arrowhead). A value of `One` means that the arrowhead SHALL be at the Target Object. A value of `Both` means that there SHALL be an arrowhead at both ends of the Association line.
sourceRef: BaseElement	The `BaseElement` that the **Association** is connecting from.
targetRef: BaseElement	The `BaseElement` that the **Association** is connecting to.

Figure 2: *Association* attribute definition ([10, p.98])

running is text is also taken into consideration to identify extended rules.

Generally extracting rules from the running text is not as trivial as extracting rules from the tables because here often some interpretation is needed. Moreover some rules are not explicitly mentioned but are only implicit rules and it is not always the case that the different rule sources are well aligned. Instead of this the running text, the class diagrams and tables are often inconsistent.

In such cases we used another source to evaluate which constraint was intended by the standard authors or how the constraint is interpreted in practice. These additional source was most frequently the normative XSD files or - when not applicable - we analyzed how tool vendors implemented the issue in their modeling or execution tools and engines.

2.2 Assumptions and Limitations

As already mentioned in the Introduction our aim is to provide an technology agnostic list of all model serialization rules or constraints defined in the BPMN standard document. Therefore the main restriction of our approach is the concentration of rules regarding the serialization of BPMN models. The serialization of BPMN diagrams (see [10, Chap.12]) and all graphical constraints such as the appearance of shapes, their nesting, etc. is out the scope of this work.

Other aspects which are not covered in our list of constraints are all rules regarding process execution and runtime constraints. This includes instance attributes of BPMN models and particularly constraints regarding the execution semantics of BPMN.

In contrast to this limitations our approach does not assume the usage of the XSD serialization of BPMN models but is technology agnostic. This is needed in order to check *all* rules - regardless of the underlying serialization format. In section 4 we discuss how the normative XSD files already cover the extracted constraints.

3 Overview of All Extracted Constraints

This section presents an overview over all extracted constraints ordered by their categorization into the four main categories already introduced in section 2.1. For all categories a short introduction is given and than all correspondent rules are listed tabularly.

3.1 Basic Attribute/Sub Elements Cardinality

The most elementary rules are the basic attribute and element cardinality constraints which declare for each type of BPMN element which attributes and model associations apply. Respecting these constraints it is already possible to create structural correct BPMN models. For each identified attribute (and sub element) a name, a datatype and a occurrence multiplicity has to be defined.

These details are required to describe all cardinality rules:

- **running number (#):** unique number for each constraint (CARD.001-CARD.311)
- **Element:** name of the affected BPMN element
- **Attribute/Sub Element:** name of the affected attribute or sub element
- **Type:** required data type of the referenced or attached attribute/sub element
- **Cardinality:** required attribute/element multiplicity (`[0..1]`, `[0..*]`, `[1]`, `[1..*]` or non-standard cardinality)

Table 1 gives an overview of all 311 cardinality constraints.

#	Element	Attribute/Sub Element	Type	Cardinality
CARD.001	definitions	name	String	1
CARD.002	definitions	targetNamespace	String	1
CARD.003	definitions	expressionLanguage	String	0..1
CARD.004	definitions	typeLanguage	String	0..1
CARD.005	definitions	rootElements	RootElement	0..*
CARD.006	definitions	diagrams	BPMNDiagram	0..*
CARD.007	definitions	imports	Import	0..*
CARD.008	definitions	extensions	Extension	0..*
CARD.009	definitions	relationships	Relationship	0..*
CARD.010	definitions	exporter	String	0..1
CARD.011	definitions	exporterVersion	String	0..1
CARD.012	import	importType	String	1
CARD.013	import	location	String	0..1
CARD.014	import	namespace	String	1
CARD.015	baseElement	id	String	1
CARD.016	baseElement	documentation	Documentation	0..*
CARD.017	baseElement	extensionDefinitions	ExtensionDefininition	0..*

#	Element	Attribute/Sub Element	Type	Cardinality
CARD.018	baseElement	extensionValues	ExtensionAttributeValue	0..*
CARD.019	documentation	text	String	1
CARD.020	documentation	textFormat	String	1
CARD.021	extension	mustUnderstand	boolean	0..1
CARD.022	extension	definition	ExtensionDefininition	1
CARD.023	extensionDefinition	name	String	
CARD.024	extensionDefinition	extensionAttributeDefinitions	ExtensionAttributeDefinition	0..*
CARD.025	extensionAttributeDefinition	name	String	1
CARD.026	extensionAttributeDefinition	type	String	1
CARD.027	extensionAttributeDefinition	isReference	boolean	0..1
CARD.028	extensionAttributeValue	value	Element	0..1
CARD.029	extensionAttributeValue	valueRef	Element	0..1
CARD.030	extensionAttributeValue	extensionAttributeDefinition	ExtensionAttributeDefinition	1
CARD.031	relationship	type	String	1
CARD.032	relationship	direction	RelationshipDirection	1
CARD.033	relationship	sources	Element	1..*
CARD.034	relationship	targets	Element	1..*
CARD.035	association	associationDirection	AssociationDirection	1
CARD.036	association	sourceRef	BaseElement	1
CARD.037	association	targetRef	BaseElement	1
CARD.038	group	categoryValueRef	CategoryValue	0..1
CARD.039	category	name	String	1
CARD.040	category	categoryValue	CategoryValue	0..*
CARD.041	categoryValue	value	String	1
CARD.042	categoryValue	category	Category	0..1
CARD.043	categoryValue	categorizedFlowElements	FlowElement	0..*
CARD.044	textAnnotation	text	String	1
CARD.045	textAnnotation	textFormat	String	1
CARD.046	correlationKey	name	String	0..1
CARD.047	correlationKey	correlationPropertyRef	CorrelationProperty	0..*
CARD.048	correlationProperty	name	String	0..1
CARD.049	correlationProperty	type	String	0..1
CARD.050	correlationProperty	correlationPropertyRetrievalExpression	CorrelationPropertyRetrievalExpression	1..*
CARD.051	correlationPropertyRetrievalExpression	messagePath	FormalExpression	1
CARD.052	correlationPropertyRetrievalExpression	messageRef	Message	1
CARD.053	correlationSubscription	correlationKeyRef	CorrelationKey	1
CARD.054	correlationSubscription	correlationPropertyBinding	CorrelationPropertyBinding	0..*
CARD.055	correlationPropertyBinding	dataPath	FormalExpression	1
CARD.056	correlationPropertyBinding	correlationPropertyRef	CorrelationProperty	1
CARD.057	error	structureRef	ItemDefinition	0..1
CARD.058	error	name	String	1
CARD.059	error	errorCode	String	0..1

#	Element	Attribute/Sub Element	Type	Cardinality
CARD.060	escalation	structureRef	ItemDefinition	0..1
CARD.061	escalation	name	String	1
CARD.062	escalation	escalationCode	String	0..1
CARD.063	formalExpression	language	String	0..1
CARD.064	formalExpression	body	Element	1
CARD.065	formalExpression	evaluatesToTypeRef	ItemDefinition	1
CARD.066	flowElement	name	String	0..1
CARD.067	flowElement	categoryValueRef	CategoryValue	0..*
CARD.068	flowElement	auditing	Auditing	0..1
CARD.069	flowElement	monitoring	Monitoring	0..1
CARD.070	flowElementsContainer	flowElements	FlowElement	0..*
CARD.071	flowElementsContainer	laneSets	LaneSet	0..*
CARD.072	gateway	gatewayDirection	GatewayDirection	1
CARD.073	itemDefinition	itemKind	ItemKind	1
CARD.074	itemDefinition	structureRef	Element	0..1
CARD.075	itemDefinition	import	Import	0..1
CARD.076	itemDefinition	isCollection	boolean	1
CARD.077	message	name	String	1
CARD.078	message	itemRef	ItemDefinition	0..1
CARD.079	resource	name	String	1
CARD.080	resource	resourceParameters	ResourceParameter	0..*
CARD.081	resourceParameter	name	String	1
CARD.082	resourceParameter	type	ItemDefinition	1
CARD.083	resourceParameter	isRequired	boolean	1
CARD.084	sequenceFlow	sourceRef	FlowNode	1
CARD.085	sequenceFlow	targetRef	FlowNode	1
CARD.086	sequenceFlow	conditionExpression	Expression	0..1
CARD.087	sequenceFlow	isImmediate	boolean	0..1
CARD.088	flowNode	incoming	SequenceFlow	0..*
CARD.089	flowNode	outgoing	SequenceFlow	0..*
CARD.090	interface	name	String	1
CARD.091	interface	operations	Operation	1..*
CARD.092	interface	callableElements	CallableElement	0..*
CARD.093	interface	implementationRef	Element	0..1
CARD.094	operation	name	String	1
CARD.095	operation	inMessageRef	Message	1
CARD.096	operation	outMessageRef	Message	0..1
CARD.097	operation	errorRef	Error	0..*
CARD.098	operation	implementationRef	Element	0..1
CARD.099	collaboration	name	String	1
CARD.100	collaboration	choreographyRef	Choreography	0..*
CARD.101	collaboration	correlationKeys	CorrelationKey	0..*
CARD.102	collaboration	conversationAssociations	ConversationAssociation	0..*
CARD.103	collaboration	conversations	ConversationNode	0..*
CARD.104	collaboration	conversationLinks	ConversationLink	0..*
CARD.105	collaboration	artifacts	Artifact	0..*
CARD.106	collaboration	participants	Participant	0..*
CARD.107	collaboration	participantAssociations	ParticipantAssociation	0..*
CARD.108	collaboration	messageFlow	MessageFlow	0..*
CARD.109	collaboration	messageFlowAssociations	messageFlowAssociation	0..*
CARD.110	collaboration	isClosed	boolean	1
CARD.111	participant	name	String	0..1
CARD.112	participant	processRef	Process	0..1

#	Element	Attribute/Sub Element	Type	Cardinality
CARD.113	participant	partnerRoleRef	PartnerRole	0..*
CARD.114	participant	partnerEntityRef	PartnerEntity	0..*
CARD.115	participant	interfaceRef	Interface	0..*
CARD.116	participant	participantMultiplicity	ParticipantMultiplicity	0..1
CARD.117	participant	endPointRefs	EndPoint	0..*
CARD.118	partnerEntity	name	String	1
CARD.119	partnerEntity	participantRef	Participant	0..*
CARD.120	partnerRole	name	String	1
CARD.121	partnerRole	participantRef	Participant	0..*
CARD.122	participantMultiplicity	minimum	integer	1
CARD.123	participantMultiplicity	maximum	integer	0..1
CARD.124	participantAssociation	innerParticipantRef	Participant	1
CARD.125	participantAssociation	outerParticipantRef	Participant	1
CARD.126	messageFlow	name	String	1
CARD.127	messageFlow	sourceRef	InteractionNode	1
CARD.128	messageFlow	targetRef	InteractionNode	1
CARD.129	messageFlow	messageRef	Message	0..1
CARD.130	messageFlowAssociation	innerMessageFlowRef	MessageFlow	1
CARD.131	messageFlowAssociation	outerMessageFlowRef	MessageFlow	1
CARD.132	conversationNode	name	String	0..1
CARD.133	conversationNode	participantRefs	Participant	2..*
CARD.134	conversationNode	messageFlowRefs	MessageFlow	0..*
CARD.135	conversationNode	correlationKeys	CorrelationKey	0..*
CARD.136	subConversation	conversationNodes	ConversationNode	0..*
CARD.137	callConversation	calledCollaborationRef	Collaboration	0..1
CARD.138	callConversation	participantAssociations	ParticipantAssociation	0..*
CARD.139	conversationLink	name	String	0..1
CARD.140	conversationLink	sourceRef	InteractionNode	1
CARD.141	conversationLink	targetRef	InteractionNode	1
CARD.142	conversationAssociation	innerConversationNodeRef	ConversationNode	0..1
CARD.143	conversationAssociation	outerConversationNodeRef	ConversationNode	0..1
CARD.144	process	processType	ProcessType	1
CARD.145	process	isExecutable	boolean	0..1
CARD.146	process	auditing	Auditing	0..1
CARD.147	process	monitoring	Monitoring	0..1
CARD.148	process	artifacts	Artifact	0..1
CARD.149	process	isClosed	boolean	1
CARD.150	process	supports	Process	0..*
CARD.151	process	properties	Property	0..*
CARD.152	process	resources	ResourceRole	0..*
CARD.153	process	correlationSubscriptions	CorrelationSubscription	0..*
CARD.154	process	definitionalCollaborationRef	Collaboration	0..1
CARD.155	activity	isForCompensation	boolean	1
CARD.156	activity	loopCharecteristics	LoopCharacteristics	0..1
CARD.157	activity	resources	ResourceRole	0..*
CARD.158	activity	default	SequenceFlow	0..1
CARD.159	activity	ioSpecification	InputOutputSpecification	0..1
CARD.160	activity	properties	Property	0..*
CARD.161	activity	boundaryEventRefs	BoundaryEvent	0..*
CARD.162	activity	dataInputAssociations	DataInputAssociation	0..*

#	Element	Attribute/Sub Element	Type	Cardinality
CARD.163	activity	dataOutputAssociations	DataOutputAssociation	0..*
CARD.164	activity	startQuantity	integer	1
CARD.165	activity	completionQuantity	integer	1
CARD.166	resourceRole	resourceRef	Resource	0..1
CARD.167	resourceRole	resourceAssignmentExpression	RessourceAssignmentExpression	0..1
CARD.168	resourceRole	resourceParameterBindings	ResourceParameterBinding	0..*
CARD.169	resourceAssignmentExpression	expression	Expression	1
CARD.170	resourceParameterBinding	parameterRef	ResourceParameter	1
CARD.171	resourceParameterBinding	expression	Expression	1
CARD.172	serviceTask	implementation	String	1
CARD.173	serviceTask	operationRef	Operation	0..1
CARD.174	sendTask	messageRef	Message	0..1
CARD.175	sendTask	operationRef	Operation	0..1
CARD.176	sendTask	implementation	String	1
CARD.177	receiveTask	messageRef	Message	0..1
CARD.178	receiveTask	instantiate	boolean	1
CARD.179	receiveTask	operationRef	Operation	0..1
CARD.180	receiveTask	implementation	String	1
CARD.181	businessRuleTask	implementation	String	1
CARD.182	scriptTask	scriptFormat	String	0..1
CARD.183	scriptTask	script	String	0..1
CARD.184	userTask	implementation	String	1
CARD.185	userTask	renderings	Rendering	0..*
CARD.186	subProcess	triggeredByEvent	boolean	1
CARD.187	subProcess	artifacts	Artifact	0..*
CARD.188	transaction	method	TransactionMethod	1
CARD.189	adHocSubProcess	completionCondition	Expression	1
CARD.190	adHocSubProcess	ordering	AdHocOrdering	1
CARD.191	adHocSubProcess	cancelRemainingInstances	boolean	1
CARD.192	callActivity	calledElement	CallableElement	0..1
CARD.193	callableElement	name	String	0..1
CARD.194	callableElement	supportedInterfaceRefs	Interface	0..*
CARD.195	callableElement	ioSpecification	InputOutputSpecification	0..1
CARD.196	callableElement	ioBinding	InputOutputBinding	0..*
CARD.197	inputOutputBinding	inputDataRef	DataInput	1
CARD.198	inputOutputBinding	outputDataRef	DataOutput	1
CARD.199	inputOutputBinding	operationRef	Operation	1
CARD.200	GlobalTask	resources	ResourceRole	0..*
CARD.201	StandardLoopCharacteristics	testBefore	boolean	1
CARD.202	StandardLoopCharacteristics	loopMaximum	integer	0..1
CARD.203	StandardLoopCharacteristics	loopCondition	Expression	0..1
CARD.204	MultiInstanceLoopCharacteristics	isSequential	boolean	1
CARD.205	MultiInstanceLoopCharacteristics	loopCardinality	Expression	0..1

#	Element	Attribute/Sub Element	Type	Cardinality
CARD.206	MultiInstanceLoopCharacteristics	loopDataInputRef	ItemAwareElement	0..1
CARD.207	MultiInstanceLoopCharacteristics	loopDataOutputRef	ItemAwareElement	0..1
CARD.208	MultiInstanceLoopCharacteristics	inputDataItem	DataInput	0..1
CARD.209	MultiInstanceLoopCharacteristics	outputDataItem	DataOutput	0..1
CARD.210	MultiInstanceLoopCharacteristics	behavior	MultiInstanceBehavior	1
CARD.211	MultiInstanceLoopCharacteristics	complexBehaviorDefinition	ComplexBehaviorDefinition	0..*
CARD.212	MultiInstanceLoopCharacteristics	completionCondition	Expression	0..1
CARD.213	MultiInstanceLoopCharacteristics	oneBehaviorRef	EventDefinition	0..1
CARD.214	MultiInstanceLoopCharacteristics	noneBehaviorRef	EventDefinition	0..1
CARD.215	ComplexBehaviorDefinition	condition	FormalExpression	1
CARD.216	ComplexBehaviorDefinition	event	ImplicitThrowEvent	1
CARD.217	ItemAwareElement	itemSubjectRef	ItemDefinition	0..1
CARD.218	ItemAwareElement	dataState	DataState	0..1
CARD.219	DataObject	isCollection	boolean	1
CARD.220	DataObjectReference	dataObjectRef	DataObject	1
CARD.221	DataState	name	String	1
CARD.222	DataStore	name	String	1
CARD.223	DataStore	capacity	integer	0..1
CARD.224	DataStore	isUnlimited	boolean	1
CARD.225	DataStoreReference	dataStoreRef	DataStore	1
CARD.226	Property	name	String	1
CARD.227	InputOutputSpecification	inputSets	InputSet	1..*
CARD.228	InputOutputSpecification	outputSets	OutputSet	1..*
CARD.229	InputOutputSpecification	dataInputs	DataInput	0..*
CARD.230	InputOutputSpecification	dataOutputs	DataOutput	0..*
CARD.231	DataInput	name	String	0..1
CARD.232	DataInput	inputSetRefs	InputSet	1..*
CARD.233	DataInput	inputSetwithOptional	InputSet	0..*
CARD.234	DataInput	inputSetWithWhileExecuting	InputSet	0..*
CARD.235	DataInput	isCollection	boolean	1
CARD.236	DataOutput	name	String	0..1
CARD.237	DataOutput	outputSetRefs	OutputSet	1..*
CARD.238	DataOutput	ioutputSetwithOptional	OutputSet	0..*
CARD.239	DataOutput	outputSetWithWhileExecuting	OutputSet	0..*
CARD.240	DataOutput	isCollection	boolean	1
CARD.241	InputSet	name	String	0..1
CARD.242	InputSet	dataInputRefs	DataInput	0..*
CARD.243	InputSet	optionalInputRefs	DataInput	0..*
CARD.244	InputSet	whileExecutingInputRefs	DataInput	0..*
CARD.245	InputSet	outputSetRefs	OutputSet	0..*

#	Element	Attribute/Sub Element	Type	Cardinality
CARD.246	OutputSet	name	String	0..1
CARD.247	OutputSet	dataOutputRefs	DataOutput	0..*
CARD.248	OutputSet	optionalOutputRefs	DataOutput	0..*
CARD.249	OutputSet	whileExecutingOutput-Refs	DataOutput	0..*
CARD.250	OutputSet	inputSetRefs	InputSet	0..*
CARD.251	DataAssociation	transformation	Expression	0..1
CARD.252	DataAssociation	assignment	Assignment	0..*
CARD.253	DataAssociation	sourceRef	ItemAwareElement	0..*
CARD.254	DataAssociation	targetRef	ItemAwareElement	1
CARD.255	Assignment	from	Expression	1
CARD.256	Assignment	to	Expression	1
CARD.257	Event	properties	Property	0..*
CARD.258	CatchEvent	eventDefinitionRefs	EventDefinition	0..*
CARD.259	CatchEvent	eventDefinitions	EventDefinition	0..*
CARD.260	CatchEvent	dataOutputAssociations	DataOutputAssociation	0..*
CARD.261	CatchEvent	dataOutputs	DataOutput	0..*
CARD.262	CatchEvent	outputSet	OutputSet	0..1
CARD.263	CatchEvent	parallelMultiple	boolean	1
CARD.264	ThrowEvent	eventDefinitionRefs	EventDefinition	0..*
CARD.265	ThrowEvent	eventDefinitions	EventDefinition	0..*
CARD.266	ThrowEvent	dataInputAssociations	DataInputAssociation	0..*
CARD.267	ThrowEvent	dataInuts	DataInput	0..*
CARD.268	ThrowEvent	inputSet	InputSet	0..1
CARD.269	StartEvent	isInterrupting	boolean	1
CARD.270	BoundaryEvent	attachedTo	Activity	1
CARD.271	BoundaryEvent	cancelActivity	boolean	1
CARD.272	CompensationEvent-Definition	activityRef	Activity	0..1
CARD.273	CompensationEvent-Definition	waitForCompletion	boolean	1
CARD.274	ConditionalEvent-Definition	condition	Expression	1
CARD.275	ErrorEventDefinition	error	Error	0..1
CARD.276	EscalationEventDefinition	escalationRef	Escalation	0..1
CARD.277	LinkEventDefinition	name	String	1
CARD.278	LinkEventDefinition	sources	LinkEventDefinition	1..*
CARD.279	LinkEventDefinition	target	LinkEventDefinition	1
CARD.280	MessageEventDefinition	messageRef	Message	0..1
CARD.281	MessageEventDefinition	operationRef	Operation	0..1
CARD.282	SignalEventDefinition	signalRef	Signal	0..1
CARD.283	*Signal*[4]	*name*	*String*	*0..1*
CARD.284	*Signal*[3]	*structureRef*	*ItemDefinition*	*0..1*
CARD.285	TimerEventDefinition	timeDate	Expression	0..1
CARD.286	TimerEventDefinition	timeDuration	Expression	0..1
CARD.287	TimerEventDefinition	timeCycle	Expression	0..1
CARD.288	ExclusiveGateway	default	SequenceFlow	0..1
CARD.289	InclusiveGateway	default	SequenceFlow	0..1
CARD.290	ComplexGateway	activationCondition	Expression	0..1
CARD.291	ComplexGateway	default	SequenceFlow	0..1
CARD.292	EventBasedGateway	instantiate	boolean	1

[4]Element and attributes not directly defined in [10]; see description in section 4.1 on page 30

#	Element	Attribute/Sub Element	Type	Cardinality
CARD.293	EventBasedGateway	eventGatewayType	EventGatewayType	1
CARD.294	LaneSet	name	String	0..1
CARD.295	LaneSet	process	Process	1
CARD.296	LaneSet	lanes	Lane	0..*
CARD.297	LaneSet	parentLane	Lane	0..1
CARD.298	Lane	name	String	1
CARD.299	Lane	partitionElement	BaseElement	0..1
CARD.300	Lane	partitionElementRef	BaseElement	0..1
CARD.301	Lane	childLaneSet	LaneSet	0..1
CARD.302	Lane	flowNodeRefs	FlowNode	0..*
CARD.303	ChoreographyActivity	participantRefs	Participant	2..*
CARD.304	ChoreographyActivity	initiatingParticipantRef	Participant	1
CARD.305	ChoreographyActivity	loopType	ChoreographyLoopType	1
CARD.306	ChoreographyActivity	correlationKeys	CorrelationKey	0..*
CARD.307	ChoreographyTask	messageFlowRef	MessageFlow	1..*
CARD.308	SubChoreography	artifacts	Artifact	0..*
CARD.309	CallChoreography	calledChoreographyRef	CallableElement	0..1
CARD.310	CallChoreography	participantAssiociations	ParticipantAssociation	0..*
CARD.311	GlobalChoreographyTask	initiatingParticipantRef	Participant	1

Table 1: Overview: Basic Attributes and Sub Elements and their Cardinality

3.2 Basic Value Restrictions and Default Values

Especially for attributes which refer to basic datatypes (such as *boolean* or *integer*) or to datatypes derived from such datatypes (e.g., *String* enumerations) the BPMN standard [10] defines value restrictions and default values.

These details are required to describe all value restriction rules:

- **running number (#):** unique number for each constraint (VAL.001-VAL.041)
- **Element:** name of the affected BPMN element
- **Attribute:** affected attribute or sub element name
- **Type:** required data type of the attribute/sub element
- **Default Value:** required default value ("-", if no default value is required)
- **Value Restriction:** list of all allowed values; following the notation in [10]: ({ Value1 | Value2 | ... }) ("-", if there is no value restriction)

The list of all value constraints follows in Table 2.

#	Element	Attribute	Type	Default Value	Value Restriction
VAL.001	definitions	expressionLanguage	String	http://www.w3.org/1999/Xpath	-
VAL.002	definitions	typeLanguage	String	http://www.w3.org/2001/XMLSchema	-
VAL.003	documentation	textFormat	String	"text/plain"	-
VAL.004	extension	mustUnderstand	boolean	false	-
VAL.005	extensionAttributeDefinition	isReference	boolean	false	-
VAL.006	relationship	direction	RelationshipDirection	-	{None | Forward | Backward | Both}
VAL.007	association	associationDirection	AssociationDirection	"None"	{None | One | Both}
VAL.008	textAnnotation	textFormat	String	"text/plain"	-
VAL.009	gateway	gatewayDirection	GatewayDirection	"Unspecified"	{ Unspecified | Converging | Diverging | Mixed}
VAL.010	itemDefinition	itemKind	ItemKind	"Information"	{Information | Physical}
VAL.011	itemDefinition	isCollection	boolean	false	-
VAL.012	collaboration	isClosed	boolean	false	-
VAL.013	participantMultiplicity	minimum	integer	0	-
VAL.014	participantMultiplicity	maximum	integer	1	-
VAL.015	process	processType	ProcessType	"None"	{None | Private | Public}
VAL.016	process	isClosed	boolean	false	-
VAL.017	activity	isForCompensation	boolean	false	-

#	Element	Attribute	Type	Default Value	Value Restriction
VAL.018	activity	startQuantity	integer	1	-
VAL.019	activity	completion-Quantity	integer	1	-
VAL.020	serviceTask	implementation	String	"##WebService"	-
VAL.021	sendTask	implementation	String	"##WebService"	-
VAL.022	receiveTask	instantiate	boolean	false	-
VAL.023	receiveTask	implementation	String	"##WebService"	-
VAL.024	businessRule-Task	implementation	String	"##unspecified"	-
VAL.025	userTask	implementation	String	"##unspecified"	-
VAL.026	subProcess	triggeredBy-Event	boolean	false	-
VAL.027	adHocSubPro-cess	ordering	AdHoc-Ordering	"Parallel"	{ Parallel | Sequential }
VAL.028	adHocSubPro-cess	cancelRemain-ingInstances	boolean	true	-
VAL.029	StandardLoop-Characteristics	testBefore	boolean	false	-
VAL.030	MultiInstance-LoopCharacter-istics	isSequential	boolean	false	-
VAL.031	MultiInstance-LoopCharacter-istics	behavior	MultiIn-stance-Behavior	"all"	{ None | One | All | Complex }
VAL.032	DataObject	isCollection	boolean	false	-
VAL.033	DataStore	isUnlimited	boolean	false	-
VAL.034	DataInput	isCollection	boolean	false	-
VAL.035	DataOutput	isCollection	boolean	false	-
VAL.036	CatchEvent	parallelMultiple	boolean	false	-
VAL.037	StartEvent	isInterrupting	boolean	true	
VAL.038	Compensation-EventDefinition	waitForComple-tion	boolean	true	-
VAL.039	EventBased-Gateway	instantiate	boolean	false	-
VAL.040	EventBased-Gateway	eventGateway-Type	Event-Gateway-Type	"Exclusive"	{Exclusive | Parallel }
VAL.041	Choreography-Activity	loopType	Choreo-graphy-Loop-Type	"None"	{ None | Standard | Mul-tiInstanceSequential | MultiInstanceParallel }

Table 2: Overview: Basic Value Restrictions and Default Values

3.3 Basic Reference Constraints

References are used heavily in BPMN models. The two main sources for reference usage are on the one hand the ability of BPMN to define reusable elements that might be used in various referencing elements (e.g., various *Message* definitions might refer to the same *ItemDefinition*) and on the other hand flow definitions. All flow definitions are also implemented using references to sources, targets and flow definitions.

For all references two aspects are important: First, the reference must be resolvable, i.e., the referenced element exists either in the same model or has been imported. Second, for most references only particular elements are allowed. Using the example from before: The attribute *itemRef* of the element *Message* must refer to an *ItemDefintion* (see REF.017). All other element references violate this constraint.

For a technology agnostic view on the references it is sufficient to know which attribute is a reference and what type of BPMN elements is referenced. However, as the information is needed for further checks (see Sec. 4.3), in this case also the XSD-specific attribute naming and implementation is added to the tabular overview of all reference constraints.

Therefore, the structure of the reference constraint table is:

- **running number (#):** unique number for each constraint (REF.001-REF.107)
- **Element:** name of the affected BPMN element
- **Attribute/Sub Element:** name of the referencing attribute or sub element
- **Type:** data type of the referenced element
- **Attribute/Element name in XSD:** name used in the XSD files
- **XSD Type:** denotes whether an *xsd:IDREF* or *xsd:QName* is used to implement the reference in the XSD files

The list of all reference constraints follows in Table 3.

#	Element	Attribute/Sub Element	Type	Attribute/Element name in XSD	XSD Type
REF.001	extension	definition	ExtensionDefinition	definition	QName
REF.002	relationship	sources	Element	source	QName
REF.003	relationship	targets	Element	target	QName
REF.004	association	sourceRef	BaseElement	sourceRef	QName
REF.005	association	targetRef	BaseElement	targetRef	QName
REF.006	group	categoryValueRef	CategoryValue	categoryValueRef	QName
REF.007	correlationKey	correlationPropertyRef	CorrelationProperty	correlationPropertyRef	QName
REF.008	correlationProperty	type	String	type	QName
REF.009	correlationPropertyRetrievalExpression	messageRef	Message	messageRef	QName

#	Element	Attribute/Sub Element	Type	Attribute/Element name in XSD	XSD Type
REF.010	correlationSubscription	correlationKeyRef	CorrelationKey	correlationKeyRef	QName
REF.011	correlationPropertyBinding	correlationPropertyRef	CorrelationProperty	correlationPropertyRef	QName
REF.012	error	structureRef	ItemDefinition	structureRef	QName
REF.013	escalation	structureRef	ItemDefinition	structureRef	QName
REF.014	formalExpression	evaluatesToTypeRef	ItemDefinition	evaluatesToTypeRef	QName
REF.015	flowElement	categoryValueRef	CategoryValue	categoryValueRef	QName
REF.016	itemDefinition	structureRef	Element	structureRef	QName
REF.017	message	itemRef	ItemDefinition	itemRef	QName
REF.018	resourceParameter	type	ItemDefinition	type	QName
REF.019	sequenceFlow	sourceRef	FlowNode	sourceRef	IDREF
REF.020	sequenceFlow	targetRef	FlowNode	targetRef	IDREF
REF.021	flowNode	incoming	SequenceFlow	incoming	QName
REF.022	flowNode	outgoing	SequenceFlow	outgoing	QName
REF.023	interface	implementationRef	Element	implementationRef	QName
REF.024	operation	inMessageRef	Message	inMessageRef	QName
REF.025	operation	outMessageRef	Message	outMessageRef	QName
REF.026	operation	errorRef	Error	errorRef	QName
REF.027	operation	implementationRef	Element	implementationRef	QName
REF.028	collaboration	choreographyRef	Choreography	choreographyRef	QName
REF.029	participant	processRef	Process	processRef	QName
REF.030	participant	interfaceRef	Interface	interfaceRef	QName
REF.031	participant	endPointRefs	EndPoint	endPointRef	QName
REF.032	partnerEntity	participantRef	Participant	participantRef	QName
REF.033	partnerRole	participantRef	Participant	participantRef	QName
REF.034	participantAssociation	innerParticipantRef	Participant	innerParticipantRef	QName
REF.035	participantAssociation	outerParticipantRef	Participant	outerParticipantRef	QName
REF.036	messageFlow	sourceRef	InteractionNode	sourceRef	QName
REF.037	messageFlow	targetRef	InteractionNode	targetRef	QName
REF.038	messageFlow	messageRef	Message	messageRef	QName
REF.039	messageFlowAssociation	innerMessageFlowRef	MessageFlow	innerMessageFlowRef	QName
REF.040	messageFlowAssociation	outerMessageFlowRef	MessageFlow	outerMessageFlowRef	QName
REF.041	conversationNode	participantRefs	Participant	participantRef	QName
REF.042	conversationNode	messageFlowRefs	MessageFlow	messageFlowRef	QName
REF.043	callConversation	calledCollaborationRef	Collaboration	calledCollaborationRef	QName
REF.044	conversationLink	sourceRef	InteractionNode	sourceRef	QName
REF.045	conversationLink	targetRef	InteractionNode	targetRef	QName
REF.046	conversationAssociation	innerConversationNodeRef	ConversationNode	innerConversationNodeRef	QName
REF.047	conversationAssociation	outerConversationNodeRef	ConversationNode	outerConversationNodeRef	QName
REF.048	process	supports	Process	supports	QName
REF.049	process	definitionalCollaborationRef	Collaboration	definitionalCollaborationRef	QName
REF.050	activity	default	SequenceFlow	default	IDREF
REF.051	resourceRole	resourceRef	Resource	resourceRef	QName

#	Element	Attribute/Sub Element	Type	Attribute/Element name in XSD	XSD Type
REF.052	resourceParameterBinding	parameterRef	ResourceParameter	parameterRef	QName
REF.053	serviceTask	operationRef	Operation	operationRef	QName
REF.054	sendTask	messageRef	Message	messageRef	QName
REF.055	sendTask	operationRef	Operation	operationRef	QName
REF.056	receiveTask	messageRef	Message	messageRef	QName
REF.057	receiveTask	operationRef	Operation	operationRef	QName
REF.058	callActivity	calledElement	CallableElement	calledElement	QName
REF.059	callableElement	supportedInterfaceRefs	Interface	supportedInterfaceRefs	QName
REF.060	inputOutputBinding	inputDataRef	DataInput	inputDataRef	IDREF
REF.061	inputOutputBinding	outputDataRef	DataOutput	outputDataRef	IDREF
REF.062	inputOutputBinding	operationRef	Operation	operationRef	QName
REF.063	MultiInstanceLoopCharacteristics	loopDataInputRef	ItemAwareElement	loopDataInputRef	QName
REF.064	MultiInstanceLoopCharacteristics	loopDataOutputRef	ItemAwareElement	loopDataOutputRef	QName
REF.065	MultiInstanceLoopCharacteristics	oneBehaviorRef	EventDefinition	oneBehaviorRef	QName
REF.066	MultiInstanceLoopCharacteristics	noneBehaviorRef	EventDefinition	noneBehaviorRef	QName
REF.067	DataObject	-	-	itemSubjectRef	QName
REF.068	DataObjectReference	dataObjectRef	DataObject	dataObjectRef	IDREF
REF.069	DataObjectReference	-	-	itemSubjectRef	QName
REF.070	DataStore	-	-	itemSubjectRef	QName
REF.071	DataStoreReference	dataStoreRef	DataStore	dataStoreRef	QName
REF.072	DataStoreReference	-	-	itemSubjectRef	QName
REF.073	Property	-	-	itemSubjectRef	QName
REF.074	DataInput	-	-	itemSubjectRef	QName
REF.075	DataOutput	-	-	itemSubjectRef	QName
REF.076	InputSet	dataInputRefs	DataInput	dataInputRefs	IDREF
REF.077	InputSet	optionalInputRefs	DataInput	optionalInputRefs	IDREF
REF.078	InputSet	whileExecutingInputRefs	DataInput	whileExecutingInputRefs	IDREF
REF.079	InputSet	outputSetRefs	OutputSet	outputSetRefs	IDREF
REF.080	OutputSet	dataOutputRefs	DataOutput	dataOutputRefs	IDREF
REF.081	OutputSet	optionalOutputRefs	DataOutput	optionalOutputRefs	IDREF
REF.082	OutputSet	whileExecutingOutputRefs	DataOutput	whileExecutingOutputRefs	IDREF
REF.083	OutputSet	inputSetRefs	InputSet	inputSetRefs	IDREF
REF.084	DataAssociation	sourceRef	ItemAwareElement	sourceRef	IDREF
REF.085	DataAssociation	targetRef	ItemAwareElement	targetRef	IDREF
REF.086	CatchEvent	eventDefinitionRefs	EventDefinition	eventDefinitionRef	QName
REF.087	ThrowEvent	eventDefinitionRefs	EventDefinition	eventDefinitionRef	QName

#	Element	Attribute/Sub Element	Type	Attribute/Element name in XSD	XSD Type
REF.088	BoundaryEvent	attachedTo	Activity	attachedToRef	QName
REF.089	CompensationEventDefinition	activityRef	Activity	activityRef	QName
REF.090	ErrorEventDefinition	error	Error	errorRef	QName
REF.091	EscalationEventDefinition	escalationRef	Escalation	escalationRef	QName
REF.092	LinkEventDefinition	sources	LinkEventDefinition	source	QName
REF.093	LinkEventDefinition	target	LinkEventDefinition	target	QName
REF.094	MessageEventDefinition	messageRef	Message	messageRef	QName
REF.095	MessageEventDefinition	operationRef	Operation	operationRef	QName
REF.096	SignalEventDefinition	signalRef	Signal	signalRef	QName
REF.097	Signal			structureRef	QName
REF.098	ExclusiveGateway	default	SequenceFlow	default	IDREF
REF.099	InclusiveGateway	default	SequenceFlow	default	IDREF
REF.100	ComplexGateway	default	SequenceFlow	default	IDREF
REF.101	Lane	partitionElementRef	BaseElement	partitionElementRef	QName
REF.102	Lane	flowNodeRefs	FlowNode	flowNodeRef	IDREF
REF.103	ChoreographyActivity	participantRefs	Participant	participantRef	QName
REF.104	ChoreographyActivity	initiatingParticipantRef	Participant	initiatingParticipantRef	QName
REF.105	ChoreographyTask	messageFlowRef	MessageFlow	messageFlowRef	QName
REF.106	CallChoreography	calledChoreographyRef	CallableElement	calledChoreographyRef	QName
REF.107	GlobalChoreographyTask	initiatingParticipantRef	Participant	initiatingParticipantRef	QName

Table 3: Overview: Basic Reference Constraints

3.4 Extended Constraints

In contrast to the previously introduced constraint categories which can be described with a few attributes as they all follow the same pattern the following extended constraints (`EXT`) need to be described in more detail. This is mainly due to the fact that the rules in this section cover more complex requirements, often depicted in the running standard text.

The detailed descriptions of all extracted extended constraints follows a common structure which is outlined in the following section. The subsequent section shows an overview of all extended constraints in tabular form, whereas the actual extensive descriptions can be found in Appendix A.

3.4.1 Constraint Structure and Important Attributes

Although the extended constraint cannot be categorized as they focus on different aspects, some describing attributes are relevant for all of them. Most rules concern a specific BPMN element and one of its attributes. In order to describe the constraint a textual description is always needed. And to provide an ability to reproduce the (potential) interpretation and rule extraction the standard document is always quoted and referenced.

The common tabular structure which is used to provide a clear representation is shown exemplarily in table 4.

Rule #			Conf.Level	
Label:				
	Affected Element:			
	Attribute/Sub Element:			
Constraint:				
	(Pre-) Condition:			
	Source:			
			Chapter	pg.

Table 4: Example to illustrate the extensive description of all Extended Constraints

All used aspects in table 4 are introduced and summarized in the following bullet list:

- **Rule #:** unique number for each constraint (EXT.001-EXT.145)
- **Conformance Level (Conf.Level):** indication for which BPMN conformance class [10, Sec. 2] the actual rule is relevant; possible values:
  - *all:* relevant for all conformance classes
  - *proc:* relevant for *Process Modeling Conformance* [10, Sec. 2.1]

  - *exec:* relevant for *Process Execution Conformance* [10, Sec. 2.2]
  - *chor:* relevant for *Choreography Modeling Conformance* [10, Sec. 2.4]
  - arbitrary combinations of the three former values

- **Label:** short summarizing rule label
- **Affected Element:** name of the affected BPMN element (might be empty, denoted as "-", if a rule does not apply to a single element)
- **Attribute/Sub Element:** affected attribute or sub element name ("-", if no single attribute is affected)
- **Constraint:** textual specification of the actual constraint
- **(Pre-) Condition:** condition under which the rule applies
- **Source:** quotation of the corresponding statement in the standard document [10]
- **Chapter & Page:** corresponding chapter and page in [10] where the quotation can be found

3.4.2 Tabular Overview

As mentioned before, the following table 5 gives an overview of all extended constraints. The compact description of the extended rules consists of:

- **running number (#):** unique number for each constraint (EXT.001-EXT.145)
- **Element:** name of the affected BPMN element ("-", if rule does not apply to single element)
- **Attribute/Sub Element:** name of the affected attribute or sub element ("-", if rule does not apply to specific attribute)
- **Label:** short summarizing name for each actual rule
- **Details:** reference[5] to the section in the appendix A containing the detailed constraint description

#	Element	Attribute/Sub Element	Label	Details
EXT.001	import	-	ImportProcessing	A.1
EXT.002	baseElement	id	BaseElementUniqueId	A.2
EXT.003	baseElement	id	BaseElementId	A.3
EXT.004	documentation	textFormat	DocumentationTextFormatMimeType	A.4
EXT.005	artifact	-	ArtifactExtensibility	A.5

[5]hyperlink in the digital version of this document

#	Element	Attribute/Sub Element	Label	Details
EXT.006	artifact/SequenceFlow	-	ArtifactSeqFlowNoTarget	A.6
EXT.007	artifact/SequenceFlow	-	ArtifactSeqFlowNoSource	A.7
EXT.008	artifact/MessageFlow	-	ArtifactMsgFlowNoTarget	A.8
EXT.009	artifact/MessageFlow	-	ArtifactMsgFlowNoSource	A.9
EXT.010	textAnnotation	textFormat	TextAnnotationTextFormatMimeType	A.10
EXT.011	Escalation	escalationCode	EscalationEscalationCodeRules	A.11
EXT.012	Expression		ExpressionNaturalLanguageNotExecutable	A.12
EXT.013	FormalExpression		FormalExpressionBodyMustBePresent	A.13
EXT.014	FormalExpression	language	FormalExpressionLanguageUriFormat	A.14
EXT.015	FlowElementContainer		FlowElementContainerLaneSetConstraint	A.15
EXT.016	Gateway	gatewayDirection	GatewayGatewayDirectionUnspecifiedConstraint	A.16
EXT.017	Gateway	gatewayDirection	GatewayGatewayDirectionConvergingConstraint	A.17
EXT.018	Gateway	gatewayDirection	GatewayGatewayDirectionDivergingConstraint	A.18
EXT.019	Gateway	gatewayDirection	GatewayGatewayDirectionMixedConstraint	A.19
EXT.020	ItemDefinition	isCollection	ItemDefinitionIsCollectionCheck	A.20
EXT.021	SequenceFlow	sourceRef	SequenceFlowSourceRefConstraints	A.21
EXT.022	SequenceFlow	targetRef	SequenceFlowTargetRefConstraints	A.22
EXT.023	SequenceFlow	-	SequenceFlowUsageConsistencyCheck	A.23
EXT.024	SequenceFlow	isImmediate	SequenceFlowIsImmediateExecutableConstraint	A.24
EXT.025	SequenceFlow		SequenceFlowConditionalActivityConstraint	A.25
EXT.026	SequenceFlow	(sourceRef)	SequenceFlowDefaultSeqFlowCheck	A.26
EXT.027	Collaboration	choreographyRef	CollaborationChoreographyRefConstraints	A.27
EXT.028	SequenceFlow	sourceRef / targetRef	PoolNoSequenceFlowBetweenPools	A.28
EXT.029	ParticipantMultiplicity	minimum	ParticipantMultiplicityMinimumValidValueCheck	A.29
EXT.030	ParticipantMultiplicity	maximum	ParticipantMultiplicityMaximumValidValueCheck	A.30
EXT.031	MessageFlow	sourceRef / targetRef	MessageFlowConnectionOfPools	A.31
EXT.032	GlobalConversation	-	GlobalConversationConstraints	A.32
EXT.033	Collaboration	isClosed	CollaborationIsClosedAdditionalConstraint	A.33
EXT.034	-	-	LaneConsistentToConversation	A.34
EXT.035	Process	isExecutable	ProcessIsExecutableConstraint	A.35
EXT.036	Process	flowElement	ProcessFlowElementConstraints	A.36
EXT.037	Activity	startQuantity	ActivityStartQuantityValidValueCheck	A.37
EXT.038	Activity	completionQuantity	ActivityCompletionQuantityValidValueCheck	A.38

#	Element	Attribute/Sub Element	Label	Details
EXT.039	ResourceRole	resourceRef / resourceParameterBinding	ResourceRoleSubelemConstraint	A.39
EXT.040	ResourceRole	resourceParameterBinding	ResourceRoleResourceParameterBindingConstraint	A.40
EXT.041	Task	-	TaskLoopXORMultiInstanceMarker	A.41
EXT.042	ServiceTask	InputSet	ServiceTaskExactlyOneInputSet	A.42
EXT.043	ServiceTask	OutputSet	ServiceTaskAtMostOneOutputSet	A.43
EXT.044	dataInput	itemSubjectRef	ServiceTaskDataInputItemDefCheck	A.44
EXT.045	dataOutput	itemSubjectRef	ServiceTaskDataOutputItemDefCheck	A.45
EXT.046	SendTask	InputSet	SendTaskAtMostOneInputSet	A.46
EXT.047	SendTask	DataInput	SendTaskAtMostOneDataInput	A.47
EXT.048	DataInput (as SubSubelem of SendTask)	itemSubjectRef	SendTaskDataInputItemDefCheck	A.48
EXT.049	ReceiveTask	instantiate	ReceiveTaskInstantiateConstraint	A.49
EXT.050	ReceiveTask	OutputSet	ReceiveTaskAtMostOneOutputSet	A.50
EXT.051	ReceiveTask	DataOutput	ReceiveTaskAtMostOneDataOutput	A.51
EXT.052	DataOutput (as SubSubelem of ReceiveTask)	itemSubjectRef	ReceiveTaskDataOutputItemDefCheck	A.52
EXT.053	ScriptTask	scriptFormat	ScriptTaskScriptFormatRequiredWhenScriptPresent	A.53
EXT.054	SubProcess		SubProcessMaximumOfThreeMarkers	A.54
EXT.055	SubProcess		SubProcessLoopXORMultiInstance	A.55
EXT.056	-	-	SubProcessFlowElementContainerConstraints	A.56
EXT.057	SubProcess		EventSubProcessNoSequenceFlows	A.57
EXT.058	SubProcess	StartEvent	EventSubProcessExactlyOneStartEvent	A.58
EXT.059	SubProcess	EventDefinition	EventSubProcessRequiresEventDefinition	A.59
EXT.060	Transaction	method	TransactionMethodExecConstraint	A.60
EXT.061	AdHocSubProcess		AdHocSubProcessAtLeastOneActivity	A.61
EXT.062	AdHocSubProcess		AdHocSubProcessElementsNotAllowed	A.62
EXT.063	CallActivity	InputOutputSpecification (inherited from Activity)	CallActivitityInputOutputSpecificationConstraints	A.63
EXT.064	CallableElement	ioBinding	CallableElementAtLeastOneIOBinding	A.64
EXT.065	CallableElement	ioBinding	CallableElementInputOutputBindingConstraint	A.65
EXT.066	GlobalTask	-	GlobalTasksCheckCorrespondingRules	A.66
EXT.067	MultiInstanceLoopCharacteristics	-	MultiInstanceLoopCharacteristicsRequiredLoopCardinalityOrLoopDataInput	A.67
EXT.068	MultiInstanceLoopCharacteristics	loopDataInputRef	MultiInstanceLoopCharacteristicsLoopDataInputForTasks	A.68
EXT.069	MultiInstanceLoopCharacteristics	inputDataItem	MultiInstanceLoopCharacteristicsOutputDataItemType	A.69

#	Element	Attribute/Sub Element	Label	Details
EXT.070	MultiInstanceLoopCharacteristics	outputDataItem	MultiInstanceLoopCharacteristicsInputDataItemType	A.70
EXT.071	MultiInstanceLoopCharacteristics	CompletionCondition	MultiInstanceLoopCharacteristicsCompletionConditionBooleanExpression	A.71
EXT.072	ComplexBehaviorDefinition	condition	ComplexBehaviorDefinitionElementCondition	A.72
EXT.073	DataObject	-	DataObjectsAllowedInProcessesOnly	A.73
EXT.074	DataObject	dataState	DataObjectsCannotSpecifyStates	A.74
EXT.075	DataObjectReference	itemSubjectRef	DataObjectReferencesCannotSpecifyItemDefinitions	A.75
EXT.076	DataObjectReference	name	DataObjectReferenceNamingConvention	A.76
EXT.077	DataObject	isCollection	DataObjectIsCollectionSameValueAsInItemDef	A.77
EXT.078	Property	-	PropertyUsageRestrictions	A.78
EXT.079	InputOutputSpecification	-	InputOutputSpecificationUsageRestrictions	A.79
EXT.080	DataInput	-	DataInputAssociationConstraints	A.80
EXT.081	DataInput	isCollection	DataInputIsCollectionDefaultValue	A.81
EXT.082	DataOutput	-	DataOutputAssociationConstraints	A.82
EXT.083	DataOutput	isCollection	DataOutputIsCollectionDefaultValue	A.83
EXT.084	DataInput	-	DataInputMustBeReferencedByAnInputSet	A.84
EXT.085	InputSet	optionalInputRefs	InputSetOptionalInputRefDefinedAsDataInput	A.85
EXT.086	InputSet	whileExecutingInputRefs	InputSetWhileExecutingInputRefsDefinedAsDataInput	A.86
EXT.087	InputSet / OutputSet	outputSetRefs / inputSetRefs	InputOutputRuleDefinition	A.87
EXT.088	DataOutput	-	DataOutputMustBeReferencedByAnOutputSet	A.88
EXT.089	OutputSet	optionalOutputRefs	OutputSetOptionalOutputRefDefinedAsDataOutput	A.89
EXT.090	OutputSet	whileExecutingOutputRefs	OutputSetWhileExecutingOutputRefsDefinedAsDataOutput	A.90
EXT.091	DataAssociation	-	DataAssociationValidityCheck	A.91
EXT.092	DataAssociation	sourceRef	DataAssociationExactlyOneSourceRefIfNoTransformationIsPresent	A.92
EXT.093	Event		EventDataInputOutputConstraints	A.93
EXT.094	Event		EventDataInputOutputItemDefinitionConstraints	A.94
EXT.095	ThrowEvent	eventDefinitions	ThrowEventEventDefinitions	A.95
EXT.096	StartEvent	incoming	StartEventNoIncomingSequenceFlow	A.96
EXT.097	StartEvent	-	StartEventMandatoryWhenEndEventUsed	A.97
EXT.098	StartEvent		StartEventTopLevelAllowedEventDefinitions	A.98
EXT.099	CallActivity	calledElement	CallActivityCalledProcessMandatoryNoneStartEvent	A.99
EXT.100	StartEvent		StartEventSubProcessAllowedEventDefinitions	A.100
EXT.101	StartEvent	outgoing	StartEventMandatoryOutgoingSequenceFlow	A.101
EXT.102	MessageFlow	sourceRef	MessageFlowStartEventInvalidSource	A.102

#	Element	Attribute/Sub Element	Label	Details
EXT.103	StartEvent	messageEventDefinition	StartEventMessageEventDefinintionFlowConstraint	A.103
EXT.104	EndEvent	outgoing	EndEventNoOutgoingSequenceFlow	A.104
EXT.105	EndEvent	-	EndEventMandatoryWhenStartEventUsed	A.105
EXT.106	EndEvent	cancelEventDefinition	EndEventCancelOnlyAllowedInTransaction	A.106
EXT.107	EndEvent	incoming	EndEventMandatoryIncomingSequenceFlow	A.107
EXT.108	MessageFlow	targetRef	MessageFlowEndEventInvalidTarget	A.108
EXT.109	EndEvent	messageEventDefinition	EndEventMessageEventDefinintionFlowConstraint	A.109
EXT.110	BoundaryEvent	cancelActivity	BoundaryEventCancelActivityValueRestrictions	A.110
EXT.111	BoundaryEvent	-	BoundaryEventSubProcessCancelRequiresTransaction	A.111
EXT.112	BoundaryEvent	incoming	BoundaryEventNoIncomingSequenceFlow	A.112
EXT.113	BoundaryEvent	outgoing	BoundaryEventMandatoryOutgoingSequenceFlow	A.113
EXT.114	BoundaryEvent	outgoing	BoundaryEventCompensateNoOutgoingSequenceFlow	A.114
EXT.115	IntermediateEvent	incoming	IntermediateEventMandatoryIncomingSequenceFlow	A.115
EXT.116	IntermediateEvent	outgoing	IntermediateEventMandatoryOutgoingSequenceFlow	A.116
EXT.117	IntermediateEvent	incoming / outgoing	LinkIntermediateEventSequenceFlowConstraint	A.117
EXT.118	IntermediateEvent	-	LinkIntermediateEventSourceTargetMatching	A.118
EXT.119	IntermediateEvent	-	MessageIntermediateEventMessageFlowConstraint	A.119
EXT.120	BoundaryEvent	attachedToRef	CancelIntermediateEventUsageConstraint	A.120
EXT.121	StartEvent		CompensateStartEventUsageConstraint	A.121
EXT.122	BoundaryEvent	attachedToRef	CompensateIntermediateEventCatchUsageConstraint	A.122
EXT.123	conditionalEventDefinition	condition	ConditionalEventDefinitionFormalConditionMandatoryForExecutability	A.123
EXT.124	LinkEventDefinition	source	LinkEventDefinitionSourceRequiredForCatchEvents	A.124
EXT.125	LinkEventDefinition	target	LinkEventDefinitionTargetRequiredForThrowEvents	A.125
EXT.126	LinkEventDefinition	source/target	LinkEventDefinitionLinkValidityConstraint	A.126
EXT.127	MessageEventDefinition	messageRef	MessageEventDefinitionMessageRefMandatoryForExecutability	A.127
EXT.128	MessageEventDefinition	operationRef	MessageEventDefinitionOperationRefMandatoryForExecutability	A.128
EXT.129	StartEvent	-	NoneStartEventNotAllowedForEventSubProcesses	A.129
EXT.130	BoundaryEvent	-	NoneCatchBoundaryEventNotAllowed	A.130
EXT.131	TimerEventDefinition	-	TimerEventDefinitionOnlyOneAttributeAllowed	A.131
EXT.132	TimerEventDefinition	timeDate	TimerEventTimeDateFormatCheck	A.132

#	Element	Attribute/Sub Element	Label	Details
EXT.133	TimerEventDefinition	timeCycle	TimerEventTimeCycleFormatCheck	A.133
EXT.134	TimerEventDefinition	timeDuration	TimerEventTimeDurationFormatCheck	A.134
EXT.135	Gateway	incoming / outgoing	GatewayGeneralSequenceFlowConstraint	A.135
EXT.136	EventBasedGateway	outgoing	EventBasedGatewayOutgoingSequenceFlowCardinalityConstraint	A.136
EXT.137	EventBasedGateway	outgoing	EventBasedGatewayNoConditionForSequenceFlows	A.137
EXT.138	EventBasedGateway	outgoing	EventBasedGatewayComplexSequenceFlowConstraint	A.138
EXT.139	EventBasedGateway	-	EventBasedGatewayNoMixtureOfReceiveTasksAndMessageEvents	A.139
EXT.140	EventBasedGateway	-	EventBasedGatewayNoBoundaryEventsForReceiveTasks	A.140
EXT.141	EventBasedGateway	-	EventBasedGatewayNoAdditionalSequenceFlowsForTargets	A.141
EXT.142	EventBasedGateway	incoming	EventBasedGatewayInstantiationNoIncomingSequenceFlow	A.142
EXT.143	BoundaryEvent	-	CompensateBoundaryEventAssociationRequired	A.143
EXT.144	BoundaryEvent	-	CompensateBoundaryEventAssociationToCompensationActivity	A.144
EXT.145	ScriptTask	scriptFormat	ScriptTaskScriptFormatValidValueCheck	A.145
EXT.146	EndEvent	-	EndEventAllowedEventDefinitions	A.146
EXT.147	BoundaryEvent	-	BoundaryEventAllowedEventDefinitions	A.147
EXT.148	IntermediateEvent	-	IntermediateCatchEventAllowedEventDefinitions	A.148
EXT.149	IntermediateEvent	-	IntermediateThrowEventAllowedEventDefinitions	A.149
EXT.150	StartEvent	-	StartEventUsageSequenceFlowImplications	A.150
EXT.151	EndEvent	-	EndEventUsageSequenceFlowImplications	A.151
EXT.152	SequenceFlow	-	SequenceFlowNoCrossingOfSubProcessBorder	A.152

Table 5: Overview: Extended and complex constraints

4 Analysis of the XSD-based Serialization

The main contribution of our work is, as mentioned before, the extensive set of serialization constraints presented in previous sections which is not limited to the XSD-based serialization of BPMN. However, as a lot of BPMN modeling tools and engines are using this serialization format it is interesting to see which rules are covered and implemented correctly in the normative XSDs.

The model structure, value and reference constraints and even some extended constraints can be expressed using XML schema restrictions. A major advantage of using the XSD-based serialization is the ability to validate the generated models using a XML schema validation. Schema validation is supported by various XML tools and also for most programming languages with XML support.

The following analysis of the constraint coverage is in fact an analysis which rule violations can be detected by performing a XML schema validation against the normative XSD files.

4.1 Basic Attribute/Sub Elements Cardinality

Inheritance of attributes as defined by the class diagrams and textual specifications can directly be realized in XSDs. Using the *minOccurrence*, *maxOccurrence* and *use* mechanisms for XSD element and attribute definitions cardinality constraints can be implemented directly in XSD files, as well.

Therefore the normative XSD files cover most of the basic cardinality rules CARD.001-311. All deviations and constraint violations are listed in detail below.

First, all minor deviations depending on the usage of XSD and on modeling decisions are specified. It is important to note, that these deviations are no rule violations.

- **Multiple inheritance:** For some BPMN elements the standard defines multiple inheritance which cannot directly implemented in XSDs. Instead for following elements attributes for super classes have been manually added to the sub class definition:
  - **CARD.070/071 (*flowElementsContainer*):** *flowElementsContainer* is an abstract class that defines the attributes *flowElements* and *laneSets* which will be inherited to the sub classes *Process*, *SubProcess*, *Choreography* and *SubChoreography* [10, Sec. 8.3.8]. The attribute *flowElements* is directly added in the afflicted XSD elements for *Process*, *SubProcess*, *Choreography* and *SubChoreography* whereas according to the textual constraint in Table 8.45 (*"LaneSets are not used for Choreographies or Sub-Choreographies"* [10, p.89]) the sub element LaneSet is only added to the process and sub process definition (also see constraint EXT.015).
  - **CARD.217/218 (*itemAwareElement*):** The attributes *itemSubjectRef* and *dataState* are realized in the sub classes *DataObject*, *DataObjectReference*, *DataStore*, *DataStoreReference*, *Property*, *DataInput* and *DataOutput*

- **Mandatory attributes with default value:** When an attribute is declared as mandatory and additionally a default value is required it is sufficient to define the default value in the XSD only, as this value is automatically present if no other value is defined on instance level.
- **Omitting bi-directional references:** In BPMN element relations are often defined bi-directional, i.e., element A references element B and vice versa. XML as a markup language is organized hierarchical and thus, the relation between element A and B can be realized as defining element B as an child element of A. If this method is used the back reference from B to A can be omitted as element A can be identified uniquely as parent from B. Moreover, it is often sufficient
  - **CARD.042 (*categoryValue/category*):** A *categoryValue* is implemented as an XSD sub element of *Category*. Therefore an explicit reference can be omitted as the *Category* can be identified as parent element.
  - **CARD.043 (*categoryValue/categorizedFlowElements*):** *FlowElements* use the attribute *categoryValueRef* to bind itself to a *CategoryValue*. The reference collection *categorizedFlowElements* is omitted in the *categoryValue* definition
  - **CARD.092 (*interface/callableElements*):** CallableElements (e.g., *process*) use the attribute *supportedInterfaceRef* to indicate which interface they implement. The reference collection of all callableElements which support an interface is omitted in the *interface* definition.
  - **CARD.113/114 (*participant/partnerRoleRef* and *partnerEntityRef*):** *PartnerRoles* and *PartnerEntities* can reference a participant element. Using this references for each *participant* the performing *PartnerRoles* resp. *PartnerEntities* can be determined [10, p.116] and therefore the attributes *partnerRoleRef* and *partnerEntityRef* are omitted.
  - **CARD.161 (*activity/boundaryEventRefs*):** To an activity boundary events might be attached. The attribute *boundaryEventRefs* is omitted as each boundary event definition already references the activity to which it is attached by the attribute *attachedToRef*.
  - **CARD.232-234 (*DataInput*):** The atttributes *inputSetRefs*, *inputSetwithOptional* and*inputSetWithWhileExecuting* are omitted as the relation can be determined by the corresponding attributes *dataInputRefs*, *optionalInputRefs* and *whileExecutingInputRefs* of the *InputSet* in the surrounding *ioSpecification*.
  - **CARD.237-239 (*DataOutput*):** With the same explanation as for *DataInput* the attributes *outputSetRefs*, *outputSetwithOptional* and *outputSetWithWhileExecuting* can be omitted for *DataOutputs*.
- **CARD.023-030 (Extension mechanism):** In [10, Sec. 8.2.3] it is already mentioned that the BPMN extension mechanism is implemented differently in the XSD serialization format. The elements *extensionDefinition*, *extensionAttributeDefinition* and *extensionAttributeValue* are not needed as XSDs provide the possibility to add arbitrary XML attributes and elements using a `xsd:anyAttribute` and `xsd:any` definition.

- ***definitions* - additional attribute *id***: *BPMN20.xsd* adds an additional optional attribute "*id*" (see l.14)
- ***documentation* - additional attribute *id***: *Semantic.xsd* adds an additional optional attribute "*id*" (see l.509)
- ***extension* - additional sub element *documentation***: *Semantic.xsd* adds an additional sub element "*documentation*" (see l.608)
- ***resourceRole* - additional attribute *name***: *Semantic.xsd* adds an additional optional attribute "*name*" (see l.1200)
- **CARD.064 (*formalExpression/body*):** The attribute body is not implemented in the XSD serialization format, as the expression can be directly defined between the formalExpression tags. This is already indicated in the standard [10, p.86, Table 8.43]: *"Instead, the FormalExpression complex type supports mixed content. The body of the Expression would be specified as element content."*
- **CARD.075 (*itemDefinition/import*):** The direct reference to an import element is omitted in the XSD implementation. Imported definitions which are referenced by the *structureRef* attribute of an itemDefinition can be identified using the QName reference containing a unique namespace/id combination.
- **CARD.227-230 (*inputOutputSpecification*):** Element is renamed to *ioSpecification* in XSD.
- **CARD.251 (*dataAssociation/transformation*):** Due to an inconsistency between a class diagram and the tabular description of *DataAssociations* it is not clear whether the attribute *transformation* has the type *Expression* or *FormalExpression*[6] In the normative XSD the stricter type *FormalExpression* has been chosen, which ensures that all models comply to [10] but this definition might be to strict.
- **CARD.278/279 (*LinkEventDefinition*):** It is intended that each *IntermediateThrowEvent* containing a *LinkEventDefinition* has exactly one target (cardinality: `[1]`) which clearly identifies the target *IntermediateCatchEvent*. An *IntermediateCatchEvent* might be the target for various links but must be at least the target in one *LinkEventDefinition* and therefore there must be at least one source element (cardinality: `[1..*]`).
  In [10, p.270] it is not correctly distinguished between LinkEventDefinitions used in *IntermediateThrowEvent* and used in *IntermediateCatchEvents*. I.e., *LinkEventDefinitions* in an *IntermediateThrowEvent* must not contain a source definition and *LinkEventDefinition* in an *IntermediateCatchEvent* must not contain any target definition[7]. Therefore technically the cardinality of source should be `[0..*]` and the cardinality for target should be `[0..1]`. And these cardinalities are used in the normative XSD files.
  Nevertheless the intended restrictions depending on the usage in a catch or a throw event must be respected, which is described in the extended constraint EXT.123-124 (see Sec. 3.4 and appendices A.123 and A.124).

[6] also see official issue: `http://www.omg.org/issues/bpmn2-rtf.open.html#Issue15539`
[7] see official issue: `http://www.omg.org/issues/bpmn2-rtf.open.html#Issue15739`

- **CARD.283/284: Missing Element *Signal*:** In [10] exists no tabular description of attributes and model associations for *Signals*. In Fig. 10.93 the attributes *name* and *structureRef* can be derived from the class diagram. These attributes are implemented in the XSD definition as optional XML attributes.

More crucial are the following items which are in fact violations of the BPMN constraints CARD.001-311. But in some cases a discussion and clarification by the OMG is needed in order to dertermine which proposal is correct :

- **Required attributes implemented optional:** The most common inconsistency is that attributes which are defined as mandatory are implemented as optional in the corresponding XSD definition. A reason for this rather simple mistakes might be that an attribute definition in XSD does not require the usage of the *use* attribute. If the attribute is not present, the default value "`optional`" applies (see [17, sec. 3.2.2]).
  Violated constraints: CARD.001, CARD.032, CARD.044, CARD.058, CARD.061, CARD.065, CARD.077, CARD.081, CARD.082, CARD.083, CARD.099, CARD.118, CARD.120, CARD.126, CARD.216, CARD.220, CARD.221, CARD.222, CARD.225, CARD.226, CARD.271 (see also Sec. 4.2), CARD.273 (see also Sec. 4.2), CARD.298 and CARD.312

- **CARD.142/143 (*conversationAssociation*)**: Table 9.14 in [10, p.136] claims that the attributes *innerConversationNodeRef* and *outerConversationNodeRef* are optional (`[0..1]` resp. `[0..*]`). In the class diagram on the same page (Fig.9.31) and in the normative XSD file the attributes are marked as mandatory (`[1]` resp. `[1..*]`)[8]. As an empty association between elements is not meaningful the solution using mandatory attributes should be prefered.

- **CARD.189 (*adHocSubProcess/completionCondition*):** cardinality claimed in [10]: `[1]`; realized in XSD as a sub element with `minOccurence=0` and `maxOccurence=unbounded` (i.e., cardinality `[0..*]`)

- **CARD.307 (*ChoreographyTask/MessageRef*):** It is unclear how much messages are allowed for a single ChoreographyTask. [10, p.323] states that *"[a] Choreography Task is an atomic Activity in a Choreography Process. It represents an Interaction, which is one or two Message exchanges between two Participants."*. Therefore the cardinality of messageRef should be `[1..2]` (which is the case in the normative XSD). On the other hand for ChoreographyActivities (parent class of ChoreographyTask) more than 2 participants and therefore more than two message exchanges might be allowed.[9]

4.2 Basic Value Restrictions and Default Values

Most value restrictions and default value constraints revealed in Section 3.2 are correctly implemented in the normative BPMN XSD files. But there still exist some inconsistencies between

[8] also see official issue: `http://www.omg.org/issues/bpmn2-rtf.open.html#Issue15522`
[9] also see official issue: `http://www.omg.org/issues/bpmn2-rtf.open.html#Issue15639`

the standard document and the XSDs. The XSD either do not implement a given constraint, e.g., a default value is not set, or the XSD is stricter than the standard document, e.g., a default value is set where the standard does not require it.

First the violations of the value restriction rules (VAL.001-VAL.041) are listed:

- **VAL.005 (*extensionAttributeDefinition/isReference*): boolean attribute must have the default value "`false`"**
  This rule does not apply to the XSD serialization format as the extension mechanism of BPMN is implemented by leveraging *xsd:any* and *xsd:anyAttribute* constructs.
- **VAL.027 (*adHocSubProcess/ordering*): attribute must have the default value "Parallel"**
  Semantic.xsd does not define any default value (see *Semantic.xsd; l.30*).
- **VAL.033 (*DataStore/isUnlimited*): boolean attribute must have the default value "`false`"**
  Semantic.xsd defines a default value "`true`" (see *Semantic.xsd; l.487*).
- **VAL.038 (*CompensationEventDefinition/waitForCompletion*): boolean attribute must have the default value "`true`"**
  Semantic.xsd does not define any default value (see *Semantic.xsd; l.264*).

And these further inconsistencies exist:

- ***transaction*: The attribute *method* of the *transaction* element is defined as mandatory in [10, p.180, Table 10.21] but no default value is fixed.**
  The XSD attribute is optional but a default value "`##Compensate`" is defined (see *Semantic.xsd; l.1389*)
- ***boundaryEvent*: The boolean attribute *cancelActivity* is mandatory but no default value is fixed (see [10, p.258, Table 10.91]).**
  In *Semantic.xsd* the attribute is optional but the default value "`true`" is defined (see line 102).

4.3 Basic Reference Constraints

The XSD based BPMN serialization uses two different techniques to deal with references: Either a reference is implemented using *xsd:IDREF* [18, Sec. 3.4.9] or *xsd:QName* [18, Sec. 3.3.18] attributes.

Section 15.3.2 in [10] explains the intention behind this: Unique IDs are achieved in the normative XSD files using the datatype *xsd:ID*. These IDs are typically referenced using the datatype *xsd:IDREF*. If an *xsd:IDREF* attributes references a non-existent *xsd:ID*, this error can be detected using an XML schema validation. A drawback of this is that only references to IDs in the same file are possible, because when referencing IDs in other files a schema error would

be detected. In order to provide a reference mechanism that is not limited to the same file, *QNames* are used.

Basically *xsd:QNames* consist of two parts: A *local part* which is used to indicate the ID of the referenced object and a so-called *prefix* which is used to determine which file contains the ID.

The choice of *xsd:IDREF* and *xsd:QName* is summarized in [10] on p. 477 as follows: *"The BPMN XSD utilizes IDREFs wherever possible and resorts to QName only when references can span files."*

As stated before an XML schema validation helps to discover *xsd:IDREF* problems, but a schema validation does not reveal all possible reference issues. On the one hand non-existent QName reference can not be detected regardless of whether only the ID or the whole referenced file cannot be resolved. On the other hand it is not possible to check the type constraints stated in [10]. For example the attribute *sourceRef* of *SequenceFlows* must refer to *FlowNodes* (see REF.019) - but a schema validation does not report an error if, e.g., a *Message* is referenced.

Therefore, in order to detect all possible reference rule validations in serialized BPMN models an external validation mechanism which exceeds an XML schema validation is needed.

Special attention is needed for the checks of the reference constraints REF.036-37 and REF.044-045 which refer to the type *InteractionNode*. The class *InteractionNode* does not add additional attributes to its sub classes, therefore and to avoid multiple inheritance problems the definition of InteractionNode is left out in the XSD. Instead of this type checks for rules REF.036-37 and REF.044-045 must be performed directly for the sub classes *Pools/Participants*, *Activities* and *Events* (see [10, p.123]).

4.4 Extended Constraints

As the extended constraints EXT.001-145 refer to more complex semantic constraints for BPMN models, most of the constraints are not directly realizable using schema definitions and validations. All extended rules covered by the normative XSDs and their implementation are described in following bullet list:

- **EXT.002 (BaseElementUniqueId):** IDs are implemented using the XSD datatype *xsd:ID*. In valid XML document all *xsd:ID*s must be unique [17, Sec. 3.3.4.5]. Duplicated IDs can be detected by a standard schema validation.
- **EXT.015 (FlowElementContainerLaneSetConstraint):** *FlowElementContainer* is not implemented in the XSD but its attributes are added to all concrete sub classes. The stated is constraint in EXT.015 (*"Choreographies and SubChoreographies must not contain LaneSets."*) is implemented by omitting the *laneSets* attribute for (sub) choreographies.
- **EXT.016 (GatewayGatewayDirectionUnspecifiedConstraint):** The constraint states that a *Gateway* whose attribute is set to `unspecified` might have arbitrary incoming and outgoing *SequenceFlows*. This is correct in the XSD as generally the multiplicity of those two attributes is `[0..*]`.

- **EXT.039 (ResourceRoleSubelemConstraint):** The XOR choice between resource references (attribute *resourceRef*) and resource parameter bindings (attribute *resourceParameterBinding*) is implemented using an *xsd:choice* [17] construct.
- **EXT.041 (TaskLoopXORMultiInstanceMarker):** The constraint that a Task must not both defined as a loop and a multi instance task is implemented at *Activity* level where at most a single (`[0..1]`) *loopCharacteristics* can be defined.
- **EXT.055 (SubProcessLoopXORMultiInstance):** The constraint that a Task must not both defined as a loop and a multi instance task is implemented at *Activity* level where at most a single (`[0..1]`) *loopCharacteristics* can be defined.
- **EXT.131 (TimerEventDefinitionOnlyOneAttributeAllowed):** The mutual exclusion of the three different time attributes for *TimerEventDefinitions* is implemented using an *xsd:choice*.

5 Discussion and Comparison to Another Rule Collection

Every BPMN editor and engine developer tries to be compliant to the BPMN standard. BPMN constraints are checked while modeling, when models are imported or saved, or when executable models should be deployed on an BPMN engine.

As there exists no official, normative list of all constraints, it is clear that each developer has to perform a standard analysis to identify constraints regarding BPMN models. As there already exist various tools which perform consistency checks to some degree, there should also exist lots of lists of rules stated in [10]. But hardly any vendors and authors published their findings.

An exception are Bruce Silver and itp-commerce: Bruce Silver published a list of the *"most important official rules for Level 2 (non-executable) process modeling"* [16, p.136] which is used by itp-commerce in their BPMN modeling tool "*Process Modeler for Microsoft Visio*"[10].

The list [16, p.135-141] of Bruce Silver comprises 39 rules which he classifies as mandatory process modeling rules directly derived from the standard [10] and 28 additional rules which are classified as "Style Rules". Silver introduces style rules in order to improve the comprehensibility of BPMN process models. They are *"consistent with the official rules, intended to make the process logic clear from the diagram alone"* [16, p.139]. In contrast to that the rule lists at hand only consist of rules stated in [10] leaving out style conventions and guidelines.

When comparing the Silver's list of process modeling rules to our constraint set it is obvious that our list is far more extensive. This is mainly due to fact that Silver leaves out all structural and cardinality aspects which are already covered by the XSDs. Moreover, Silver divided the modeling using BPMN in "levels" which refer more or less to the definition of *Process Modeling Conformance Subclasses* in [10, p.2-9]. His rules cover "level 2" which is equal to the *"Analytic Process Modeling Conformance subclass"*. These aspects are also covered in this work, but we also consider all serialization aspects regarding the definition of executable BPMN models (*"Common Executable subclass"* [10, p.2-9]).

However, the comparison of our rule set with the constraints stated by Silver revealed that some aspects have not been covered by our rule list. Especially rules 2 and 3 are more concise in [16] and have been adapted in rules EXT.151 and EXT.152. But also the constraints EXT.147-149 and EXT.152 have been added subsequently after cross-checking the different rule sets. For Silver's rules 24, 26, 37 and 39 no source can be found in the standard and in our opinion these rules are not mandatory but are style guidelines. Rule 38 refers to the nesting of pools (resp. participants) which is covered by the structural constraints of the BPMN element *participant*.

The following table 6 relates all rules from *BPMN Method & Style* to the corresponding rules numbers in this technical reports.

[10] `http://www.itp-commerce.com/`

# in [16]	pg. in [16]	#/Comment
1	p. 136	CARD.084-085, REF.019-020, EXT.021-022
2	p. 136	EXT.151
3	p. 137	EXT.152
4	p. 137	EXT.028
5	p. 137	EXT.153
6	p. 137	EXT.025
7	p. 137	EXT.137
8	p. 137	CARD.158, CARD.288-289, CARD.290
9	p. 137	EXT.031
10	p. 137	REF.036, EXT.102, EXT.126
11	p. 137	REF.037, EXT.108, EXT.126
12	p. 137	CARD.127-128, REF.036-037
13	p. 137	EXT.096
14	p. 137	EXT.102
15	p. 137	EXT.103
16	p. 137	EXT.098
17	p. 137	EXT.100
18	p. 137	EXT.104
19	p. 137	EXT.108
20	p. 137	EXT.109
21	p. 138	EXT.113
22	p. 138	EXT.147
23	p. 138	EXT.112
24	p. 138	(no source found in [10])
25	p. 138	EXT.110
26	p. 138	(no source found in [10])
27	p. 138	REF.037, EXT.108, EXT.126
28	p. 138	REF.036, EXT.102, EXT.126
29	p. 138	EXT.149
30	p. 138	EXT.148
31	p. 138	EXT.116-117
32	p. 138	EXT.117
33	p. 138	REF.037, EXT.108, EXT.126
34	p. 138	REF.036, EXT.102, EXT.126
35	p. 138	EXT.018-019
36	p. 138	EXT.139
37	p. 138	(no source found in [10])
38	p. 138	(realized when respecting the structural constraints of participant and process)
39	p. 139	(no source found in [10])

Table 6: Mapping of Silver's rules to the corresponding Constraint Numbers in this work

6 Summary

The work presented in this technical report is a first step in a project to evaluate and assess the quality of BPMN process models. Independent from domain-specific rules, style guidelines and best-practices a sound foundation is needed. This foundation is partly missing as it is unclear which generic constraints are relevant for all BPMN process models. The set of rules we presented in the work is a required substep for farther analyses and research.

But our set of rules is also helpful for tool and engine developers as well as for modelers who want to check their models.

Modeling tools and engines depend on correct models especially when execution is planned. Each tool has an internal model and data structure to implement the structural and semantic constraints for BPMN models. The basis for such internal models is often the normative XSD or XMI format. We have shown that the XSDs are not well aligned with the standard in all aspects.

These flaws and especially all aspects which are not covered in the structural XSD and XMI definitions should be corrected and checked in BPMN compliant editors and engines. A lot of tools already perform some internal checks but a first tentative evaluation of some major editors showed that not all of our detected rules are covered. This may lead to models which violate BPMN constraints. Future work will perform an in-depth analysis of the checking mechanisms of BPMN modeling tools.

Due to this shortcomings it is not guaranteed that models created with a modeling are standard compliant. Therefore also modelers as end users of modeling tools can use the presented lists of constraints to evaluate the compliance of their concrete models. Performing a manual check of all 611 extracted rules is hardly possible. As shown in section 4 a lot of rules can be checked by performing a schema validation, if the BPMN process model is serialized using the XSD serialization format. But more than 200 constraints remain unchecked.

A reference checking tool is currently under development which will be able to detect violations of the reference constraints (REF.001-104; see Section 3.3) automatically. Automatic checks for the extended rules (Sec. 3.4) and all value restrictions and cardinality aspects which are not covered by the XSDs are planned to be implemented in future work.

As mentioned before most BPMN tool vendors must have an internal list of BPMN constraints. Due to the missing publication of this internal data we could only compare our result to the rules of Bruce Silver [16]. In comparison to Silver [16] our work is more extensive. We do not leave out all aspects which are already covered by the normative XSDs, namely all basic structural and value restrictions. And also rules regarding the *"common executable conformance subclass"* [10, p.2] which are left out in [16] are listed in our constraint list.

But the analysis of Silver's rules also made it evident that we cannot claim that our list is already complete and flawless. It is very likely that complex constraints which are not explicitly mentioned in [10] are still missing. Another source for errors may be a misinterpretation of the running text. Therefore, ongoing work is the continual improvement and completion of our rule

set.

The scientific community as well as interested practicians are invited to contribute new constraints, report errors and to give general feedback to our work.

The current status, including a list of all deviations from the constraints published here, a searchable and sortable database of all constraints and tools for checking constraints are available on our project web site:

`http://www.uni-bamberg.de/pi/bpmn-constraints`

References

[1] Business Process Management Initiative (BPMI). *Business Process Modeling Notation (BPMN) Version 1.0*, May 2004.

[2] M. Chinosi and A. Trombetta. Modeling and validating BPMN diagrams. In *Commerce and Enterprise Computing, 2009. CEC'09. IEEE Conference on*, pages 353–360. IEEE, 2009.

[3] R. M. Dijkman, M. Dumas, and C. Ouyang. Semantics and analysis of business process models in BPMN. *Information and Software Technology*, 50(12):1281 – 1294, 2008.

[4] D. Falcioni, A. Polini, A. Polzonetti, and B. Re. Direct verification of bpmn processes through an optimized unfolding technique. In *Quality Software (QSIC), 2012 12th International Conference on*, pages 179–188. IEEE, 2012.

[5] P. V. Gorp and R. Dijkman. A visual token-based formalization of BPMN 2.0 based on in-place transformations. *Information and Software Technology*, 2012.

[6] O. Kopp, D. Martin, D. Wutke, and F. Leymann. The Difference Between Graph-Based and Block-Structured Business Process Modelling Languages. *Enterprise Modelling and Information Systems*, 4(1):3–13, 2009.

[7] OASIS. *Web Services Business Process Execution Language*, April 2007. v2.0.

[8] OMG. *Business Process Modeling Notation (BPMN) Version 1.1*, January 2008.

[9] OMG. *Business Process Modeling Notation (BPMN) Version 1.2*, January 2009.

[10] OMG. *Business Process Model and Notation (BPMN) Version 2.0*, January 2011.

[11] OMG. *OMG MOF 2 XMI Mapping Specification Version 2.4.1*, August 2011.

[12] C. Ouyang, M. Dumas, A. HM, and W. M. Van Der Aalst. Pattern-based translation of bpmn process models to bpel web services. *International Journal of Web Services Research (IJWSR)*, 5(1):42–62, 2008.

[13] C. Ouyang, M. Dumas, A. H. M. ter Hofstede, and W. M. P. van der Aalst. From BPMN Process Models to BPEL Web Services. In *International Conference on Web Services (ICWS)*, pages 285–292, 2006.

[14] D. Prandi, P. Quaglia, and N. Zannone. Formal analysis of BPMN via a translation into COWS. In *Coordination Models and Languages*, pages 249–263. Springer, 2008.

[15] J. C. Recker and J. Mendling. On the translation between BPMN and BPEL: Conceptual mismatch between process modeling languages. In *The 18th International Conference on Advanced Information Systems Engineering. Proceedings of Workshops and Doctoral Consortium*, pages 521–532. Namur University Press, 2006.

[16] B. Silver. *BPMN method and style: With BPMN implementer's guide*. Cody-Cassidy Press, Aptos and California and USA, 2 edition, 2011.

[17] W3C. *XML Schema Part 1: Structures Second Edition*, October 2004.

[18] W3C. *XML Schema Part 2: Datatypes Second Edition*, October 2004.

[19] M. Weidlich, G. Decker, A. Grosskopf, and M. Weske. BPEL to BPMN: The Myth of a Straight-Forward Mapping. In R. Meersman and Z. Tari, editors, *On the Move to Meaningful Internet Systems: OTM 2008*, volume 5331 of *Lecture Notes in Computer Science*, pages 265–282. Springer Berlin Heidelberg, 2008.

[20] P. Wong and J. Gibbons. A Process Semantics for BPMN. In *Formal Methods and Software Engineering*, LNCS, pages 355–374. Springer Berlin Heidelberg, 2008.

[21] Worklow Management Coalition (WfMC). *XML Process Definition Language (XPDL) Version 2.1*, April 2008.

[22] Worklow Management Coalition (WfMC). *XML Process Definition Language (XPDL) Version 2.2*, 2012.

A Detailed Description of All Extended Constraints

A.1 Rule EXT.001: ImportProcessing

Rule #		Conf.Level
EXT.001		all
Label:	**ImportProcessing**	
	Affected Element:	`import`
	Attribute/Sub Element:	`-`

Constraint:		The referenced file must be present in the given location and must be compatible to the defined type in order to be processed.	
	(Pre-) Condition:	-	
	Source:	*implicit*	
		Chapter	pg.
		-	-

A.2 Rule EXT.002: BaseElementUniqueId

Rule #		Conf.Level
EXT.002		all
Label:	**BaseElementUniqueId**	
	Affected Element:	`baseElement`
	Attribute/Sub Element:	`id`

Constraint:		The ID must be unique.	
	(Pre-) Condition:	-	
	Source:	*implicit*	
		Chapter	pg.
		8.2.1	56

A.3 Rule EXT.003: BaseElementId

Rule #		Conf.Level
EXT.003		all
Label:	**BaseElementId**	
	Affected Element:	`baseElement`
	Attribute/Sub Element:	`id`

Constraint:		An ID must be present if the element may be referenced.	
	(Pre-) Condition:	-	
	Source:	"The id is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the id MAY be omitted."	
		Chapter	pg.
		8.2.1	56

A.4 Rule EXT.004: DocumentationTextFormatMimeType

Rule #		Conf.Level
EXT.004		all
Label:	**DocumentationTextFormatMimeType**	
	Affected Element:	`documentation`
	Attribute/Sub Element:	`textFormat`

Constraint:		The Value of the attribute must be a valid mime-type.		
	(Pre-) Condition:	-		
	Source:	"It MUST follow the mime-type format. The default is 'text/plain.'"		
			Chapter	pg.
			8.2.2	56

A.5 Rule EXT.005: ArtifactExtensibility

Rule #		Conf.Level
EXT.005		all
Label:	**ArtifactExtensibility**	
	Affected Element:	`artifact`
	Attribute/Sub Element:	-

Constraint:		Any non-standard artifact must respect the standard flow connection rules.		
	(Pre-) Condition:	-		
	Source:	"A modeler or modeling tool MAY extend a BPMN diagram and add new types of Artifacts to a Diagram. Any new Artifact MUST follow the Sequence Flow and Message Flow connection rules [...]. Associations can be used to link Artifacts to Flow Objects."		
			Chapter	pg.
			8.3.1	66

A.6 Rule EXT.006: ArtifactSeqFlowNoTarget

Rule #		Conf.Level
EXT.006		all
Label:	**ArtifactSeqFlowNoTarget**	
	Affected Element:	`artifact/SequenceFlow`
	Attribute/Sub Element:	-

Constraint:		An Artifact MUST NOT be a target for a Sequence Flow.		
	(Pre-) Condition:	-		
	Source:	"An Artifact MUST NOT be a target for a Sequence Flow."		
			Chapter	pg.
			8.3.1	67

A.7 Rule EXT.007: ArtifactSeqFlowNoSource

Rule #		Conf.Level
EXT.007		all
Label:	**ArtifactSeqFlowNoSource**	
	Affected Element:	`artifact/SequenceFlow`
	Attribute/Sub Element:	-

Constraint:		An Artifact MUST NOT be a source for a Sequence Flow.		
	(Pre-) Condition:	-		
	Source:	"An Artifact MUST NOT be a source for a Sequence Flow."		
			Chapter	pg.
			8.3.1	67

A.8 Rule EXT.008: ArtifactMsgFlowNoTarget

Rule #		Conf.Level
EXT.008		all
Label:	**ArtifactMsgFlowNoTarget**	
	Affected Element:	`artifact/MessageFlow`
	Attribute/Sub Element:	-

Constraint:		An Artifact MUST NOT be a target for a Message Flow.		
	(Pre-) Condition:	-		
	Source:	"An Artifact MUST NOT be a target for a Message Flow."		
			Chapter	pg.
			8.3.1	67

A.9 Rule EXT.009: ArtifactMsgFlowNoSource

Rule #		Conf.Level
EXT.009		all
Label:	**ArtifactMsgFlowNoSource**	
	Affected Element:	`artifact/MessageFlow`
	Attribute/Sub Element:	-

Constraint:		An Artifact MUST NOT be a source for a Message Flow.		
	(Pre-) Condition:	-		
	Source:	"An Artifact MUST NOT be a source for a Message Flow."		
			Chapter	pg.
			8.3.1	67

A.10 Rule EXT.010: TextAnnotationTextFormatMimeType

Rule #		Conf.Level
EXT.010		all
Label:	**TextAnnotationTextFormatMimeType**	
	Affected Element:	`textAnnotation`
	Attribute/Sub Element:	`textFormat`

Constraint:		The Value of the attribute must be a valid mime-type.		
	(Pre-) Condition:	-		
	Source:	"It MUST follow the mimetype format. The default is 'text/plain.'"		
			Chapter	pg.
			8.3.1	72

A.11 Rule EXT.011: EscalationEscalationCodeRules

Rule #		Conf.Level
EXT.011		exec
Label:	**EscalationEscalationCodeRules**	
	Affected Element:	`Escalation`
	Attribute/Sub Element:	`escalationCode`

Constraint:		An escalationCode must be present if the escalation is used in an EndEvent or in an intermediate Event if the trigger is an Escalation.		
	(Pre-) Condition:	process is defined as executable		
	Source:	"For an End Event: If the Result is an Escalation, then the escalationCode MUST be supplied [...] For an Intermediate Event within normal flow: If the trigger is an Escalation, then the escalationCode MUST be entered [...] For an Intermediate Event attached to the boundary of an Activity: If the trigger is an Escalation, then the escalationCode MAY be entered."		
			Chapter	pg.
			8.3.4	83

A.12 Rule EXT.012: ExpressionNaturalLanguageNotExecutable

Rule #		Conf.Level
EXT.012		exec
Label:	**ExpressionNaturalLanguageNotExecutable**	
	Affected Element:	`Expression`
	Attribute/Sub Element:	-

Constraint:		If natural-language expressions are used the process is not executable.	
	(Pre-) Condition:	-	
	Source:	"The Expression class is used to specify an Expression using natural-language text. These Expressions are not executable." (p. 84) "The Expression class is used to specify an Expression using natural-language text. These Expressions are not executable and are considered underspecified." (p.85)	
		Chapter	pg.
		8.3.6	84/ 85

A.13 Rule EXT.013: FormalExpressionBodyMustBePresent

Rule #		Conf.Level
EXT.013		all
Label:	**FormalExpressionBodyMustBePresent**	
	Affected Element:	`FormalExpression`
	Attribute/Sub Element:	`body`

Constraint:		According to constraint CARD.064 (see Sec. 3.1) *body* is a mandatory attribute of a FormalExpression. Therefore the element must contain any Expression as a string content.	
	(Pre-) Condition:	-	
	Source:	-	
		Chapter	pg.
		8.3.6	86

A.14 Rule EXT.014: FormalExpressionLanguageUriFormat

Rule #		Conf.Level
EXT.014		all
Label:	**FormalExpressionLanguageUriFormat**	
	Affected Element:	`FormalExpression`
	Attribute/Sub Element:	`language`

Constraint:		Value must conform to URI format specification.	
	(Pre-) Condition:	-	
	Source:	"The language MUST be specified in a URI format."	
		Chapter	pg.
		8.3.6	86

A.15 Rule EXT.015: FlowElementContainerLaneSetConstraint

Rule #			Conf.Level
EXT.015			chor
Label:	**FlowElementContainerLaneSetConstraint**		
	Affected Element:	`FlowElementContainer`	
	Attribute/Sub Element:	-	

Constraint:		Choreographies and SubChoreographies must not contain LaneSets		
	(Pre-) Condition:	-		
	Source:	"LaneSets are not used for Choreographies or Sub-Choreographies."		
			Chapter	pg.
			8.3.8	89

A.16 Rule EXT.016: GatewayGatewayDirectionUnspecifiedConstraint

Rule #			Conf.Level
EXT.016			all
Label:	**GatewayGatewayDirectionUnspecifiedConstraint**		
	Affected Element:	`Gateway`	
	Attribute/Sub Element:	`gatewayDirection`	

Constraint:		A unspecified Gateway may have any number of incoming and outgoing sequence flows.		
	(Pre-) Condition:	Value of attribute is "Unspecified"		
	Source:	"Unspecified: There are no constraints. The Gateway MAY have any number of incoming and outgoing Sequence Flows." (Table 8.46) "A Gateway with a gatewayDirection of unspecified MAY have both multiple incoming and outgoing Sequence Flows." (p.290)		
			Chapter	pg.
			8.3.9/ 10.5.1	91/ 290

A.17 Rule EXT.017: GatewayGatewayDirectionConvergingConstraint

Rule #			Conf.Level
EXT.017			all
Label:	**GatewayGatewayDirectionConvergingConstraint**		
	Affected Element:	`Gateway`	
	Attribute/Sub Element:	`gatewayDirection`	

Constraint:		A converging Gateway must not have more than one outgoing Sequence Flow.		
	(Pre-) Condition:	Value of attribute is “Converging”		
	Source:	“Converging: This Gateway MAY have multiple incoming Sequence Flows but MUST have no more than one (1) outgoing Sequence Flow.” (Table 8.46) “A Gateway with a gatewayDirection of converging MUST have multiple incoming Sequence Flows, but MUST NOT have multiple outgoing Sequence Flows.” (p. 290)		
			Chapter	pg.
			8.3.9/ 10.5.1	91/ 290

A.18 Rule EXT.018: GatewayGatewayDirectionDivergingConstraint

Rule #			Conf.Level
EXT.018			all
Label:	**GatewayGatewayDirectionDivergingConstraint**		
	Affected Element:	`Gateway`	
	Attribute/Sub Element:	`gatewayDirection`	

Constraint:		A diverging Gateway must not have more than one incoming Sequence Flow.		
	(Pre-) Condition:	Value of attribute is “Diverging”		
	Source:	“Diverging: This Gateway MAY have multiple outgoing Sequence Flows but MUST have no more than one (1) incoming Sequence Flow.” (Table 8.46) “A Gateway with a gatewayDirection of diverging MUST have multiple outgoing Sequence Flows, but MUST NOT have multiple incoming Sequence Flows.” (p. 290)		
			Chapter	pg.
			8.3.9/ 10.5.1	91/ 290

A.19 Rule EXT.019: GatewayGatewayDirectionMixedConstraint

Rule #			Conf.Level
EXT.019			all
Label:	**GatewayGatewayDirectionMixedConstraint**		
	Affected Element:	`Gateway`	
	Attribute/Sub Element:	`gatewayDirection`	

Constraint:		A mixed Gateway must have more than one incoming and outgoing Sequence Flow.		
	(Pre-) Condition:	Value of attribute is “Mixed”		
	Source:	“Mixed: This Gateway contains multiple outgoing and multiple incoming Sequence Flows.” (Table 8.46) “A Gateway with a gatewayDirection of mixed MUST have both multiple incoming and outgoing Sequence Flows.” (p.290)		
			Chapter	pg.
			8.3.9/ 10.5.1	91/ 290

A.20 Rule EXT.020: ItemDefinitionIsCollectionCheck

Rule #			Conf.Level
EXT.020			all
Label:	**ItemDefinitionIsCollectionCheck**		
	Affected Element:	`ItemDefinition`	
	Attribute/Sub Element:	`isCollection`	

Constraint:		A CollectionItem must be used if the ItemDefinition is declared as a Collection		
	(Pre-) Condition:	Value of attribute is “true”		
	Source:	“In cases where the data structure represents a collection, the multiplicity can be projected into the attribute isCollection. If this attribute is set to ’true’, but the actual type is not a collection type, the model is considered as invalid. BPMN compliant tools might support an automatic check for these inconsistencies and report this as an error. The default value for this element is ’false’.”		
			Chapter	pg.
			8.3.10	92

A.21 Rule EXT.021: SequenceFlowSourceRefConstraints

Rule #		Conf.Level
EXT.021		proc/chor
Label:	**SequenceFlowSourceRefConstraints**	
	Affected Element:	`SequenceFlow`
	Attribute/Sub Element:	`sourceRef`

Constraint:		Only FlowNodes are allowed as source of a Sequence Flow. (REF.019; additional Restrictions see quotation)		
	(Pre-) Condition:	-		
	Source:	"For a Process: Of the types of FlowNode, only Activities, Gateways, and Events can be the source. However, Activities that are Event Sub-Processes are not allowed to be a source. For a Choreography: Of the types of FlowNode, only Choreography Activities, Gateways, and Events can be the source." (Table 8.51)		
			Chapter	pg.
			8.3.13	99

A.22 Rule EXT.022: SequenceFlowTargetRefConstraints

Rule #		Conf.Level
EXT.022		proc/chor
Label:	**SequenceFlowTargetRefConstraints**	
	Affected Element:	`SequenceFlow`
	Attribute/Sub Element:	`targetRef`

Constraint:		Only FlowNodes are allowed as target of a Sequence Flow. (REF.020; additional Restrictions see quotation)		
	(Pre-) Condition:	-		
	Source:	"For a Process: Of the types of FlowNode, only Activities, Gateways, and Events can be the target. However, Activities that are Event Sub-Processes are not allowed to be a target. For a Choreography: Of the types of FlowNode, only Choreography Activities, Gateways, and Events can be the target." (Table 8.51)		
			Chapter	pg.
			8.3.13	99

A.23 Rule EXT.023: SequenceFlowUsageConsistencyCheck

Rule #		Conf.Level
EXT.023		proc/chor
Label:	**SequenceFlowUsageConsistencyCheck**	
	Affected Element:	`SequenceFlow`
	Attribute/Sub Element:	-

Constraint:		The source and target element of the sequence flow must reference the SequenceFlow definition using their incoming/outcoming attributes.	
	(Pre-) Condition:	-	
	Source:	*implicit*	
		Chapter	pg.
		8.3.13	99

A.24 Rule EXT.024: SequenceFlowIsImmediateExecutableConstraint

Rule #		Conf.Level
EXT.024		exec
Label:	**SequenceFlowIsImmediateExecutableConstraint**	
	Affected Element:	`SequenceFlow`
	Attribute/Sub Element:	`isImmediate`

Constraint:		The optional attribute must not be "false" for executable processes.	
	(Pre-) Condition:	process is defined as executable	
	Source:	"An optional boolean value specifying whether Activities or Choreography Activities not in the model containing the Sequence Flow can occur between the elements connected by the Sequence Flow. If the value is true, they MAY NOT occur. If the value is false, they MAY occur. Also see the isClosed attribute on Process, Choreography, and Collaboration. When the attribute has no value, the default semantics depends on the kind of model containing Sequence Flows: [...] - For an executable Processes no value has the same semantics as if the value were true. - For executable Processes, the attribute MUST NOT be false." (Table 8.51)	
		Chapter	pg.
		8.3.13	99

A.25 Rule EXT.025: SequenceFlowConditionalActivityConstraint

Rule #		Conf.Level
EXT.025		all
Label:	**SequenceFlowConditionalActivityConstraint**	
	Affected Element:	`SequenceFlow`
	Attribute/Sub Element:	-

Constraint:		An Activity must not have only one outgoing conditional sequence flow.		
	(Pre-) Condition:	conditionExpression is present.		
	Source:	"If a conditional Sequence Flow is used from a source Activity, then there MUST be at least one other outgoing Sequence Flow from that Activity."		
			Chapter	pg.
			8.3.13	97

A.26 Rule EXT.026: SequenceFlowDefaultUsageConsistencyCheck

Rule #		Conf.Level
EXT.026		all
Label:	**SequenceFlowDefaultUsageConsistencyCheck**	
	Affected Element:	`SequenceFlow`
	Attribute/Sub Element:	`(sourceRef)`

Constraint:		If an activity or gateway references a sequenceFlow as default flow - the referenced sequence flow must reference the activity/the gateway as sourceRef		
	(Pre-) Condition:	Source Activity or Gateway is using the default attribute.		
	Source:	*implicit* & "A Sequence Flow that has an Exclusive, Inclusive, or Complex Gateway or an Activity as its source can also be defined with as default."		
			Chapter	pg.
			8.3.13	98

A.27 Rule EXT.027: CollaborationChoreographyRefConstraints

Rule #		Conf.Level
EXT.027		proc/chor
Label:	**CollaborationChoreographyRefConstraints**	
	Affected Element:	`Collaboration`
	Attribute/Sub Element:	`choreographyRef`

Constraint:		A choreography or a GlobalConversation must not reference a choreography.		
	(Pre-) Condition:	-		
	Source:	"Note that this attribute is not applicable for Choreography or GlobalConversation which are a subtypes of Collaboration. Thus, a Choreography cannot reference another Choreography."		
			Chapter	pg.
			9	110

A.28 Rule EXT.028: PoolNoSequenceFlowBetweenPools

Rule #		Conf.Level
EXT.028		proc/chor
Label:	**PoolNoSequenceFlowBetweenPools**	
	Affected Element:	`SequenceFlow`
	Attribute/Sub Element:	`sourceRef/targetRef`

Constraint:	A Sequence Flow must not cross the border of a Pool (i.e., a Sequence flow must link to elements of a single process)	
(Pre-) Condition:	-	
Source:	"The Sequence Flows can cross the boundaries between Lanes of a Pool [...], but cannot cross the boundaries of a Pool. That is, a Process is fully contained within the Pool. The interaction between Pools is shown through Message Flows."	
	Chapter	pg.
	9.2	112

A.29 Rule EXT.029: ParticipantMultiplicityMinimumValidValueCheck

Rule #		Conf.Level
EXT.029		proc/chor
Label:	**ParticipantMultiplicityMinimumValidValueCheck**	
	Affected Element:	`ParticipantMultiplicity`
	Attribute/Sub Element:	`minimum`

Constraint:	int value >= 0	
(Pre-) Condition:	-	
Source:	*implicit*	
	Chapter	pg.
	9.2.1	118

A.30 Rule EXT.030: ParticipantMultiplicityMaximumValidValueCheck

Rule #		Conf.Level
EXT.030		proc/chor
Label:	**ParticipantMultiplicityMaximumValidValueCheck**	
	Affected Element:	`ParticipantMultiplicity`
	Attribute/Sub Element:	`maximum`

Constraint:	int value >=1 AND >= `minimum` value	
(Pre-) Condition:	-	
Source:	"The value of maximum MUST be one or greater, AND MUST be equal or greater than the minimum value."	
	Chapter	pg.
	9.2.1	118

A.31 Rule EXT.031: MessageFlowConnectionOfPools

Rule #		Conf.Level
EXT.031		proc/chor
Label:	**MessageFlowConnectionOfPools**	
	Affected Element:	`MessageFlow`
	Attribute/Sub Element:	`sourceRef/targetRef`

Constraint:		A message flow must connect 'InteractionNodes' from different Pools.		
	(Pre-) Condition:	-		
	Source:	"A Message Flow MUST connect two separate Pools. They connect either to the Pool boundary or to Flow Objects within the Pool boundary. They MUST NOT connect two objects within the same Pool."		
			Chapter	pg.
			9.3	120

A.32 Rule EXT.032: GlobalConversationConstraints

Rule #		Conf.Level
EXT.032		proc/chor
Label:	**GlobalConversationConstraints**	
	Affected Element:	`GlobalConversation`
	Attribute/Sub Element:	-

Constraint:		*see quotation*		
	(Pre-) Condition:	-		
	Source:	"A GlobalConversation is a restricted type of Collaboration, it is an 'empty Collaboration'. A GlobalConversation MUST NOT contain any ConversationNodes. Since a GlobalConversation does not have any Flow Elements, it does not require MessageFlowAssociations, ParticipantAssociations, or ConversationAssociations or Artifacts. It is basically a set of Participants, Message Flows, and CorrelationKeys intended for reuse. Also, the Collaboration attribute choreographyRef is not applicable to GlobalConversation."		
			Chapter	pg.
			9.4.5	132

A.33 Rule EXT.033: CollaborationIsClosedAdditionalConstraint

Rule #		Conf.Level
EXT.033		proc/chor
Label:	**CollaborationIsClosedAdditionalConstraint**	
	Affected Element:	`Collaboration`
	Attribute/Sub Element:	`isClosed`

Constraint:		The value of isClosed must be equal in the collaboration and its contained choreography.	
	(Pre-) Condition:	Collaboration contains a Choreography.	
	Source:	"If a Collaboration contains a Choreography, then the value of the isClosed attribute MUST be the same in both."	
		Chapter	pg.
		9.4.8	137

A.34 Rule EXT.034: LaneConsistentToConversation

Rule #		Conf.Level
EXT.034		proc/chor
Label:	**LaneConsistentToConversation**	
	Affected Element:	-
	Attribute/Sub Element:	-

Constraint:		Flow Elements contained in a Lane which represents a Conversation must be consistent to the Conversation definition regarding the sent messages.	
	(Pre-) Condition:	A Lane is used to represent a Conversation.	
	Source:	"When a Lane (in a Process) represents a Conversation, the Flow Elements in the Lane (or elements nested or called in them) that send or receive Messages MUST do so as part of the Conversation represented by the Lane."	
		Chapter	pg.
		9.5	137

A.35 Rule EXT.035: ProcessIsExecutableConstraint

Rule #		Conf.Level
EXT.035		proc
Label:	**ProcessIsExecutableConstraint**	
	Affected Element:	`Process`
	Attribute/Sub Element:	`isExecutable`

Constraint:		A Public process may not be marked as executable.	
	(Pre-) Condition:	-	
	Source:	"For public Processes, no value has the same semantics as if the value were false. The value MAY not be true for public Processes."	
		Chapter	pg.
		10	148

A.36 Rule EXT.036: ProcessFlowElementConstraints

Rule #			Conf.Level	
EXT.036			89	
Label:	**ProcessFlowElementConstraints**			
	Affected Element:	`Process`		
	Attribute/Sub Element:	`flowElement`		
Constraint:		A Process must not contain Choreography Activities (derived rule from FlowElementsContainer)		
	(Pre-) Condition:			
	Source:	"Choreography Activities MUST NOT be included as a flowElement for a Process."		
			Chapter	pg.
			8.3.8	89

A.37 Rule EXT.037: ActivityStartQuantityValidValueCheck

Rule #			Conf.Level	
EXT.037			proc	
Label:	**ActivityStartQuantityValidValueCheck**			
	Affected Element:	`Activity`		
	Attribute/Sub Element:	`startQuantity`		
Constraint:		int value >= 1		
	(Pre-) Condition:	-		
	Source:	"The default value is 1. The value MUST NOT be less than 1."		
			Chapter	pg.
			10.2	152

A.38 Rule EXT.038: ActivityCompletionQuantityValidValueCheck

Rule #			Conf.Level	
EXT.038			proc	
Label:	**ActivityCompletionQuantityValidValueCheck**			
	Affected Element:	`Activity`		
	Attribute/Sub Element:	`completionQuantity`		
Constraint:		int value >= 1		
	(Pre-) Condition:	-		
	Source:	"The default value is 1. The value MUST NOT be less than 1."		
			Chapter	pg.
			10.2	153

A.39 Rule EXT.039: ResourceRoleSubelemConstraint

Rule #			Conf.Level
EXT.039			proc
Label:	**ResourceRoleSubelemConstraint**		
	Affected Element:	`ResourceRole`	
	Attribute/Sub Element:	`resourceRef / resourceAssignmentExpression`	

Constraint:		Either a resourceRef XOR a resourceAssignmentExpression should be used.		
	(Pre-) Condition:	-		
	Source:	"resourceRef: Should not be specified when resourceAssignmentExpression is provided. [. . .] resourceAssignmentExpression: Should not be specified when a resourceRef is provided."		
			Chapter	pg.
			10.2.1	155

A.40 Rule EXT.040: ResourceRoleResourceParameterBindingConstraint

Rule #			Conf.Level
EXT.040			proc
Label:	**ResourceRoleResourceParameterBindingConstraint**		
	Affected Element:	`ResourceRole`	
	Attribute/Sub Element:	`resourceParameterBinding`	

Constraint:		resourceParameterBindings are only allowed if a resourceRef is specified.		
	(Pre-) Condition:	-		
	Source:	"Is only applicable if a resourceRef is specified."		
			Chapter	pg.
			10.2.1	155

A.41 Rule EXT.041: TaskLoopXORMultiInstanceMarker

Rule #		Conf.Level
EXT.041		proc
Label:	**TaskLoopXORMultiInstanceMarker**	
	Affected Element:	`Task`
	Attribute/Sub Element:	-

Constraint:	A Task must not have a Loop and a Multi-Instance marker.	
(Pre-) Condition:	-	
Source:	"BPMN specifies three types of markers for Task: a Loop marker or a Multi-Instance marker and a Compensation marker. A Task MAY have one or two of these markers. [...] The loop Marker MAY be used in combination with the compensation marker. [...] The multi-instance marker MAY be used in combination with the compensation marker. [...] The Compensation Marker MAY be used in combination with the loop marker or the multi-instance marker."	
	Chapter	pg.
	10.2.3	156-157

A.42 Rule EXT.042: ServiceTaskExactlyOneInputSet

Rule #		Conf.Level
EXT.042		proc
Label:	**ServiceTaskExactlyOneInputSet**	
	Affected Element:	`ServiceTask`
	Attribute/Sub Element:	`InputSet`

Constraint:	Exactly one SubElement <ioSpecification><InputSet> must be present	
(Pre-) Condition:	operationRef is present	
Source:	"The Service Task has exactly one inputSet [...]"	
	Chapter	pg.
	10.2.3.1	158

A.43 Rule EXT.043: ServiceTaskAtMostOneOutputSet

Rule #		Conf.Level
EXT.043		proc
Label:	**ServiceTaskAtMostOneOutputSet**	
	Affected Element:	`ServiceTask`
	Attribute/Sub Element:	`OutputSet`

Constraint:		At most one Subelement <ioSpecification><OutputSet>	
	(Pre-) Condition:	operationRef is present	
	Source:	"The Service Task has [...] at most one outputSet."	
		Chapter	pg.
		10.2.3.1	158

A.44 Rule EXT.044: ServiceTaskDataInputItemDefCheck

Rule #		Conf.Level
EXT.044		proc
Label:	**ServiceTaskDataInputItemDefCheck**	
	Affected Element:	`dataInput`
	Attribute/Sub Element:	`itemSubjectRef`

Constraint:		Referenced item must be declared as InputMessage item	
	(Pre-) Condition:	operationRef is present	
	Source:	"It has a single Data Input with an ItemDefinition equivalent to the one defined by the Message referenced by the inMessageRef attribute of the associated Operation."	
		Chapter	pg.
		10.2.3.1	158

A.45 Rule EXT.045: ServiceTaskDataOutputItemDefCheck

Rule #		Conf.Level
EXT.045		158
Label:	**ServiceTaskDataOutputItemDefCheck**	
	Affected Element:	`dataOutput`
	Attribute/Sub Element:	`itemSubjectRef`

Constraint:		Referenced item must be declared as OutputMessage item.	
	(Pre-) Condition:	operationRef is present; operation has output message	
	Source:	"If the Operation defines output Messages, the Service Task has a single Data Output that has an ItemDefinition equivalent to the one defined by the Message referenced by the outMessageRef attribute of the associated Operation."	
		Chapter	pg.
		10.2.3.1	158

A.46 Rule EXT.046: SendTaskAtMostOneInputSet

Rule #		Conf.Level
EXT.046		proc
Label:	**SendTaskAtMostOneInputSet**	
	Affected Element:	`SendTask`
	Attribute/Sub Element:	`InputSet`

Constraint:		At most one SubElement <ioSpecification><InputSet> must be present	
	(Pre-) Condition:	messageRef is present	
	Source:	"[...] constraints apply when the Send Task references a Message: The Send Task has at most one inputSet and one Data Input." (p.160) "If the Send Task is associated with a Message, there MUST be at most [one] inputSet set and at most one Data Input on the Send Task." (p.217)	
		Chapter	pg.
		10.2.3.1	160/ 217

A.47 Rule EXT.047: SendTaskAtMostOneDataInput

Rule #		Conf.Level
EXT.047		proc
Label:	**SendTaskAtMostOneDataInput**	
	Affected Element:	`SendTask`
	Attribute/Sub Element:	`DataInput`

Constraint:		At most one SubElement <ioSpecification><DataInput> must be present	
	(Pre-) Condition:	messageRef is present	
	Source:	"[...] constraints apply when the Send Task references a Message: The Send Task has at most one inputSet and one Data Input." (p.160) "If the Send Task is associated with a Message, there MUST be at most [one] inputSet set and at most one Data Input on the Send Task." (p.217)	
		Chapter	pg.
		10.2.3.1	160/ 217

A.48 Rule EXT.048: SendTaskDataInputItemDefCheck

Rule #			Conf.Level	
EXT.048			proc	
Label:	**SendTaskDataInputItemDefCheck**			
	Affected Element:	`DataInput` (as SubSubelem of SendTask)		
	Attribute/Sub Element:	`itemSubjectRef`		
Constraint:		An Item must be referenced which must be declared in referenced Message definition		
	(Pre-) Condition:	messageRef (of ReceiveTask) and a dataInput is present		
	Source:	"If the Data Input is present, it MUST have an ItemDefinition equivalent to the one defined by the associated Message."		
			Chapter	pg.
			10.2.3.1	160

A.49 Rule EXT.049: ReceiveTaskInstantiateConstraint

Rule #			Conf.Level	
EXT.049			proc	
Label:	**ReceiveTaskInstantiateConstraint**			
	Affected Element:	`ReceiveTask`		
	Attribute/Sub Element:	`instantiate`		
Constraint:		A ReceiveTask with attribute instantiate set to true must not have any incoming sequence flow.		
	(Pre-) Condition:	-		
	Source:	"This attribute MAY be set to true if the Task is the first Activity (i.e., there are no incoming Sequence Flows)." (p.162) "In order for the Receive Task to instantiate the Process its instantiate attribute MUST be set to true and it MUST NOT have any incoming Sequence Flow." (p.161)		
			Chapter	pg.
			10.2.3.1	162/ 161

A.50 Rule EXT.050: ReceiveTaskAtMostOneOutputSet

Rule #			Conf.Level	
EXT.050			proc	
Label:	**ReceiveTaskAtMostOneOutputSet**			
	Affected Element:	`ReceiveTask`		
	Attribute/Sub Element:	`OutputSet`		
Constraint:		At most one SubElement <ioSpecification><OuputSet> must be present		
	(Pre-) Condition:	messageRef is present		
	Source:	"The Receive Task has at most one outputSet"		
			Chapter	pg.
			10.2.3.1	162

A.51 Rule EXT.051: ReceiveTaskAtMostOneDataOutput

Rule #			Conf.Level	
EXT.051			proc	
Label:	**ReceiveTaskAtMostOneDataOutput**			
	Affected Element:	`ReceiveTask`		
	Attribute/Sub Element:	`DataOutput`		
Constraint:		At most one SubElement <ioSpecification><DataOutput> must be present		
	(Pre-) Condition:	messageRef is present		
	Source:	"The Receive Task has [...] at most one Data output."		
			Chapter	pg.
			10.2.3.1	162

A.52 Rule EXT.052: ReceiveTaskDataOutputItemDefCheck

Rule #			Conf.Level	
EXT.052			proc	
Label:	**ReceiveTaskDataOutputItemDefCheck**			
	Affected Element:	`DataOutput` (as SubSubelem of ReceiveTask)		
	Attribute/Sub Element:	`itemSubjectRef`		
Constraint:		An Item must be referenced which must be declared in referenced Message definition		
	(Pre-) Condition:	messageRef (of ReceiveTask) and a dataOutput is present		
	Source:	"If the Data output is present, it MUST have an ItemDefinition equivalent to the one defined by the associated Message."		
			Chapter	pg.
			10.2.3.1	162

A.53 Rule EXT.053: ScriptTaskScriptFormatRequiredWhenScriptPresent

Rule #			Conf.Level	
EXT.053			proc	
Label:	**ScriptTaskScriptFormatRequiredWhenScriptPresent**			
	Affected Element:	`ScriptTask`		
	Attribute/Sub Element:	`scriptFormat`		
Constraint:		If a script is present the script type must be defined.		
	(Pre-) Condition:	The scriptTask contains a Script subelement.		
	Source:	"[...] And it MUST be specified if a script is provided."		
			Chapter	pg.
			10.2.3.1	165

A.54 Rule EXT.054: SubProcessMaximumOfThreeMarkers

Rule #			Conf.Level	
EXT.054			proc	
Label:	**SubProcessMaximumOfThreeMarkers**			
	Affected Element:	`SubProcess`		
	Attribute/Sub Element:	-		
Constraint:		At most three Markers (SubProcess Marker, Loop, Multi-Instance, Compensation, Ad-Hoc) may be used.		
	(Pre-) Condition:	-		
	Source:	"A collapsed Sub-Process MAY have one to three of these other markers, in all combinations except that loop and multi-instance cannot be shown at the same time."		
			Chapter	pg.
			10.2.5	175

A.55 Rule EXT.055: SubProcessLoopXORMultiInstance

Rule #			Conf.Level	
EXT.055			proc	
Label:	**SubProcessLoopXORMultiInstance**			
	Affected Element:	`SubProcess`		
	Attribute/Sub Element:	-		
Constraint:		Loop and MultiInstance markers must not be used in the same SubProcess.		
	(Pre-) Condition:	-		
	Source:	"A collapsed Sub-Process MAY have one to three of these other markers, in all combinations except that loop and multi-instance cannot be shown at the same time."		
			Chapter	pg.
			10.2.5	175

A.56 Rule EXT.056: SubProcessFlowElementContainerConstraints

Rule #			Conf.Level	
EXT.056			proc	
Label:	**SubProcessFlowElementContainerConstraints**			
	Affected Element:	`SubProcess`		
	Attribute/Sub Element:	`flowElement`		
Constraint:		A Process must not contain Choreography Activities (derived rule from FlowElementsContainer)		
	(Pre-) Condition:	-		
	Source:	"Choreography Activities MUST NOT be included as a flowElement for a Process."		
			Chapter	pg.
			8.3.8	89

A.57 Rule EXT.057: EventSubProcessNoSequenceFlows

Rule #		Conf.Level
EXT.057		proc
Label:	**EventSubProcessNoSequenceFlows**	
	Affected Element:	`SubProcess`
	Attribute/Sub Element:	-

Constraint:		An Event Sub-Process MUST NOT have any incoming or outgoing Sequence Flows.	
	(Pre-) Condition:	the process is an EventSubProcess, e.g., triggeredByEvent=true	
	Source:	"An Event Sub-Process MUST NOT have any incoming or outgoing Sequence Flows."	
		Chapter	pg.
		10.2.5	177

A.58 Rule EXT.058: EventSubProcessExactlyOneStartEvent

Rule #		Conf.Level
EXT.058		proc
Label:	**EventSubProcessExactlyOneStartEvent**	
	Affected Element:	`SubProcess`
	Attribute/Sub Element:	`StartEvent`

Constraint:		An Event Sub-Process MUST have exactly one Start Event.	
	(Pre-) Condition:	the process is an EventSubProcess, e.g., triggeredByEvent=true	
	Source:	"An Event Sub-Process MUST have one and only one Start Event."	
		Chapter	pg.
		10.2.5	177

A.59 Rule EXT.059: EventSubProcessRequiresEventDefinition

Rule #		Conf.Level
EXT.059		proc
Label:	**EventSubProcessRequiresEventDefinition**	
	Affected Element:	`SubProcess`
	Attribute/Sub Element:	`EventDefinition`

Constraint:		An Event Sub-Process MUST define at least of the following EventDefinitions: messageEventDefinition, errorEventDefinition, escalationEventDefinition, compensationEventDefinition, conditionalEventDefinition, signalEventDefinition	
	(Pre-) Condition:	the process is an EventSubProcess, e.g., triggeredByEvent=true	
	Source:	"The Start Event of an Event Sub-Process MUST have a defined trigger. The Start Event trigger (EventDefinition) MUST be from the following types: Message, Error,Escalation, Compensation, Conditional, Signal, and Multiple" **problematic:** parallel multiple also allowed on pg. 260	
		Chapter	pg.
		10.2.5	177

A.60 Rule EXT.060: TransactionMethodExecConstraint

Rule #			Conf.Level
EXT.060			proc/exec
Label:	**TransactionMethodExecConstraint**		
	Affected Element:	`Transaction`	
	Attribute/Sub Element:	`method`	

Constraint:				
	The Transaction method SHOULD refer to a concrete technology using a specific URI.			
	(Pre-) Condition:	process is defined as executable		
	Source:	"For executable Processes, it SHOULD be set to a technology specific URI"		
			Chapter	pg.
			10.2.5	180

A.61 Rule EXT.061: AdHocSubProcessAtLeastOneActivity

Rule #			Conf.Level
EXT.061			proc
Label:	**AdHocSubProcessAtLeastOneActivity**		
	Affected Element:	`AdHocSubProcess`	
	Attribute/Sub Element:	-	

Constraint:				
	At least one Activity must be contained in an AdHocSubProcess			
	(Pre-) Condition:	-		
	Source:	"The list of BPMN elements that MUST be used in an Ad-Hoc Sub-Process: Activity."		
			Chapter	pg.
			10.2.5	182

A.62 Rule EXT.062: AdHocSubProcessElementsNotAllowed

Rule #			Conf.Level
EXT.062			proc
Label:	**AdHocSubProcessElementsNotAllowed**		
	Affected Element:	`AdHocSubProcess`	
	Attribute/Sub Element:	-	

Constraint:				
	Start Event, End Event, Conversations, Conversation Links and Choreography Activities MUST NOT be used in an AdHocSubProcess			
	(Pre-) Condition:	-		
	Source:	"The list of BPMN elements that MUST NOT be used in an Ad-Hoc Sub-Process: Start Event, End Event, Conversations (graphically), Conversation Links (graphically), and Choreography Activities."		
			Chapter	pg.
			10.2.5	182

A.63 Rule EXT.063: CallActivitityInputOutputSpecificationConstraints

Rule #			Conf.Level
EXT.063			proc
Label:	**CallActivitityInputOutputSpecificationConstraints**		
	Affected Element:	`CallActivity`	
	Attribute/Sub Element:	`InputOutputSpecification`	

Constraint:		*see quotation*		
	(Pre-) Condition:	callableElement is referenced		
	Source:	"A Call Activity MUST fulfill the data requirements, as well as return the data produced by the CallableElement being invoked (see Figure 10.41). This means that the elements contained in the Call Activity's InputOutputSpecification MUST exactly match the elements contained in the referenced CallableElement. This includes DataInputs, DataOutputs, InputSets, and OutputSets.		
			Chapter	pg.
			10.2.6	185

A.64 Rule EXT.064: CallableElementAtLeastOneIOBinding

Rule #			Conf.Level
EXT.064			proc
Label:	**CallableElementAtLeastOneIOBinding**		
	Affected Element:	`CallableElement`	
	Attribute/Sub Element:	`ioBinding`	

Constraint:		At least one InputOutputBinding must be defined as the Callable Element is exposed as a Service.		
	(Pre-) Condition:	element is exposed as a Service; i.e. an Interface is referenced		
	Source:	"When a CallableElement is exposed as a Service, it has to define one or more InputOutputBinding elements."		
			Chapter	pg.
			10.2.6	187

A.65 Rule EXT.065: CallableElementInputOutputBindingConstraint

Rule #		Conf.Level
EXT.065		proc
Label:	**CallableElementInputOutputBindingConstraint**	
	Affected Element:	`CallableElement`
	Attribute/Sub Element:	`ioBinding`

Constraint:		*see quotation*		
	(Pre-) Condition:	-		
	Source:	"An InputOutputBinding element binds one Input and one Output of the InputOutputSpecification to an Operation of a Service Interface."		
			Chapter	pg.
			10.2.6	187

A.66 Rule EXT.066: GlobalTasksCheckCorrespondingRules

Rule #		Conf.Level
EXT.066		proc
Label:	**GlobalTasksCheckCorrespondingRules**	
	Affected Element:	`GlobalTask`
	Attribute/Sub Element:	`-`

Constraint:		Constraints defined for Tasks must also be fulfilled by the corresponding global variants. • *GlobalUserTask: CARD.184, CARD.185, EXT.041* • *GlobalManualTask: EXT.041* • *GlobalScriptTask: CARD.182, CARD.183, EXT.041, EXT.053, EXT.145* • *GlobalBusinessRuleTask: CARD.181, EXT.041*		
	(Pre-) Condition:	-		
	Source:	"Only GlobalUserTask,GlobalManualTask, GlobalScriptTask, and GlobalBusinessRuleTask are defined in BPMN. For the sake of efficiency in this specification, the list of Task types is presented once on page 156. The behavior, attributes, and model associations defined in that section also apply to the corresponding types of Global Tasks."		
			Chapter	pg.
			10.2.7	188

A.67 Rule EXT.067: MultiInstanceLoopCharacteristicsRequiredLoopCardinalityOrLoopDataInput

Rule #			Conf.Level	
EXT.067			proc/exec	
Label:	**MultiInstanceLoopCharacteristicsRequiredLoopCardinalityOrLoopDataInput**			
	Affected Element:	`MultiInstanceLoopCharacteristics`		
	Attribute/Sub Element:	-		
Constraint:		Either a loopCardinality or a loopDataInputRef must be present.		
	(Pre-) Condition:	process is defined as executable (interpretation of 'in order to initialize a valid [...]')		
	Source:	"In order to initialize a valid multi-instance, either the loopCardinality Expression or the loopDataInput MUST be specified."		
			Chapter	pg.
			10.2.8	192

A.68 Rule EXT.068: MultiInstanceLoopCharacteristicsLoopDataInputForTasks

Rule #			Conf.Level	
EXT.068			proc/exec	
Label:	**MultiInstanceLoopCharacteristicsLoopDataInputForTasks**			
	Affected Element:	`MultiInstanceLoopCharacteristics`		
	Attribute/Sub Element:	`loopDataInputRef`		
Constraint:		Reference must be resolvable to a DataInput defined in the InputOutputSpecification of the Task.		
	(Pre-) Condition:	MultiInstanceLoopCharacteristics element is used within a Task.		
	Source:	"For Tasks it is a reference to a Data Input which is part of the Activity's InputOutputSpecification."		
			Chapter	pg.
			10.2.8	192

A.69 Rule EXT.069: MultiInstanceLoopCharacteristicsOutputDataItemType

Rule #			Conf.Level
EXT.069			proc/exec
Label:	**MultiInstanceLoopCharacteristicsOutputDataItemType**		
	Affected Element:	`MultiInstanceLoopCharacteristics`	
	Attribute/Sub Element:	`inputDataItem`	

Constraint:		Type of DataInput must be the scalar of the loopDataInput type.		
	(Pre-) Condition:	-		
	Source:	"The type of this Data Input MUST the scalar of the type defined for the loopDataInput."		
			Chapter	pg.
			10.2.8	192

A.70 Rule EXT.070: MultiInstanceLoopCharacteristicsInputDataItemType

Rule #			Conf.Level
EXT.070			proc/exec
Label:	**MultiInstanceLoopCharacteristicsInputDataItemType**		
	Affected Element:	`MultiInstanceLoopCharacteristics`	
	Attribute/Sub Element:	`outputDataItem`	

Constraint:		Type of DataOutput must be the scalar of the loopDataOutput type.		
	(Pre-) Condition:	-		
	Source:	"The type of this Data Output MUST the scalar of the type defined for the loopDataOutput."		
			Chapter	pg.
			10.2.8	192

A.71 Rule EXT.071: MultiInstanceLoopCharacteristicsCompletionConditionBooleanExpression

Rule #			Conf.Level
EXT.071			proc
Label:	**MultiInstanceLoopCharacteristicsCompletionConditionBooleanExpression**		
	Affected Element:	`MultiInstanceLoopCharacteristics`	
	Attribute/Sub Element:	`CompletionCondition`	

Constraint:		Expression must evaluate to a boolean.		
	(Pre-) Condition:	-		
	Source:	"This attribute defines a boolean Expression [...]"		
			Chapter	pg.
			10.2.8	193

A.72 Rule EXT.072: ComplexBehaviorDefinitionElementCondition

Rule #			Conf.Level
EXT.072			proc
Label:	**ComplexBehaviorDefinitionElementCondition**		
	Affected Element:	`ComplexBehaviorDefinition`	
	Attribute/Sub Element:	`condition`	

Constraint:		Expression must evaluate to a boolean.		
	(Pre-) Condition:	-		
	Source:	"This attribute defines a boolean Expression [...]"		
			Chapter	pg.
			10.2.8	195

A.73 Rule EXT.073: DataObjectsAllowedInProcessesOnly

Rule #			Conf.Level
EXT.073			proc
Label:	**DataObjectsAllowedInProcessesOnly**		
	Affected Element:	`DataObject`	
	Attribute/Sub Element:	-	

Constraint:		DataObjects may not be used outside of a (Sub-)Process.		
	(Pre-) Condition:	-		
	Source:	"Data Object elements MUST be contained within Process or Sub-Process elements."		
			Chapter	pg.
			10.3.1	205

A.74 Rule EXT.074: DataObjectsCannotSpecifyStates

Rule #			Conf.Level
EXT.074			proc
Label:	**DataObjectsCannotSpecifyStates**		
	Affected Element:	`DataObject`	
	Attribute/Sub Element:	`dataState`	

Constraint:		DataObjects cannot specify states. ***Issues:*** *XSD allows that explicitly, sections States on p.206 also mentions the ability of defining States for DataObjects*[11]		
	(Pre-) Condition:	-		
	Source:	"Data Objects cannot specify states."		
			Chapter	pg.
			10.3.1	205

[11] see official issue: `http://www.omg.org/issues/bpmn2-rtf.open.html#Issue15388`

A.75 Rule EXT.075: DataObjectReferencesCannotSpecifyItemDefinitions

Rule #			Conf.Level	
EXT.075			proc	
Label:	**DataObjectReferencesCannotSpecifyItemDefinitions**			
	Affected Element:	`DataObjectReference`		
	Attribute/Sub Element:	`itemSubjectRef`		
Constraint:		Data Object Reference cannot specify item definitions ***Issues:*** *XSD allows that explicitly, sections States on p.206 also mentions the ability of defining States for DataObjects*[12]		
	(Pre-) Condition:			
	Source:	"Data Object Reference cannot specify item definitions"		
			Chapter	pg.
			10.3.1	205

A.76 Rule EXT.076: DataObjectReferenceNamingConvention

Rule #			Conf.Level	
EXT.076			proc	
Label:	**DataObjectReferenceNamingConvention**			
	Affected Element:	`DataObjectReference`		
	Attribute/Sub Element:	`name`		
Constraint:		Naming Convention: name = <Data Object Name> [<Data Object Reference State>]		
	(Pre-) Condition:	-		
	Source:	"The names of Data Object References are derived by concatenating the name of the referenced Data Data Object the state of the Data Object Reference in square brackets as follows: <Data Object Name> [<Data Object Reference State>]."		
			Chapter	pg.
			10.3.1	205

[12] see official issue: `http://www.omg.org/issues/bpmn2-rtf.open.html#Issue15388`

A.77 Rule EXT.077: DataObjectIsCollectionSameValueAsInItemDef

Rule #		Conf.Level
EXT.077		proc
Label:	**DataObjectIsCollectionSameValueAsInItemDef**	
	Affected Element:	`DataObject`
	Attribute/Sub Element:	`isCollection`

Constraint:		The value of isCollection must be equal to the value in the referenced ItemDefinition.		
	(Pre-) Condition:	an ItemDefinition is referenced (e.g. attribute itemSubjectRef is present)		
	Source:	"If an itemDefinition is referenced, then this attribute MUST have the same value as the isCollection attribute of the referenced itemDefinition."		
			Chapter	pg.
			10.3.1	206

A.78 Rule EXT.078: PropertyUsageRestrictions

Rule #		Conf.Level
EXT.078		proc
Label:	**PropertyUsageRestrictions**	
	Affected Element:	`Property`
	Attribute/Sub Element:	-

Constraint:		A Property is only allowed within a Process, Activity or Event.		
	(Pre-) Condition:	-		
	Source:	"Certain flow elements MAY contain properties, in particular only Processes, Activities, and Events MAY contain Properties. [...] Property elements MUST be contained within a FlowElement."		
			Chapter	pg.
			10.3.1	210

A.79 Rule EXT.079: InputOutputSpecificationUsageRestrictions

Rule #		Conf.Level
EXT.079		proc
Label:	**InputOutputSpecificationUsageRestrictions**	
	Affected Element:	`InputOutputSpecification`
	Attribute/Sub Element:	-

Constraint:		InputOutputSpecifications are not allowed in SubProcesses.		
	(Pre-) Condition:	-		
	Source:	"Certain Activities and CallableElements contain a InputOutputSpecification element to describe their data requirements. [...] Not every Activity type defines inputs and outputs, only Tasks, CallableElements (Global Tasks and Processes) MAY define their data requirements. Embedded Sub-Processes MUST NOT define Data Inputs and Data Outputs directly, however they MAY define them indirectly via MultiInstanceLoopCharacteristics."		
			Chapter	pg.
			10.3.1	211

A.80 Rule EXT.080: DataInputAssociationConstraints

Rule #		Conf.Level
EXT.080		proc
Label:	**DataInputAssociationConstraints**	
	Affected Element:	`DataInput`
	Attribute/Sub Element:	-

Constraint:		DataInputs of a top-level process must not be target of a DataAssociation.		
	(Pre-) Condition:	-		
	Source:	"Data Inputs MAY have incoming Data Associations: If the Data Input is directly contained by the top-level Process, it MUST not be the target of Data Associations within the underlying model. Only Data Inputs that are contained by Activities or Events MAY be the target of Data Associations in the model."		
			Chapter	pg.
			10.3.1	213

A.81 Rule EXT.081: DataInputIsCollectionDefaultValue

Rule #		Conf.Level
EXT.081		proc
Label:	**DataInputIsCollectionDefaultValue**	
	Affected Element:	`DataInput`
	Attribute/Sub Element:	`isCollection`

Constraint:		The value of isCollection must be equal to the value in the referenced ItemDefinition.		
	(Pre-) Condition:	an ItemDefinition is referenced (e.g. attribute itemSubjectRef is present)		
	Source:	"If an itemDefinition is referenced, then this attribute MUST have the same value as the isCollection attribute of the referenced itemDefinition."		
			Chapter	pg.
			10.3.1	215

A.82 Rule EXT.082: DataOutputAssociationConstraints

Rule #		Conf.Level
EXT.082		proc
Label:	**DataOutputAssociationConstraints**	
	Affected Element:	`DataOutput`
	Attribute/Sub Element:	-

Constraint:		DataOutputs of a top-level process must not be source of a DataAssociation.		
	(Pre-) Condition:	-		
	Source:	"Data Outputs MAY have outgoing DataAssociations. If the Data Output is directly contained by the top-level Process, it MUST not be the source of Data Associations within the underlying model. Only Data Outputs that are contained by Activities or Events MAY be the target of Data Associations in the model."		
			Chapter	pg.
			10.3.1	215

A.83 Rule EXT.083: DataOutputIsCollectionDefaultValue

Rule #		Conf.Level
EXT.083		proc
Label:	**DataOutputIsCollectionDefaultValue**	
	Affected Element:	`DataOutput`
	Attribute/Sub Element:	`isCollection`

Constraint:		The value of isCollection must be equal to the value in the referenced ItemDefinition.		
	(Pre-) Condition:	an ItemDefinition is referenced (e.g. attribute itemSubjectRef is present)		
	Source:	"If an itemDefinition is referenced, then this attribute MUST have the same value as the isCollection attribute of the referenced itemDefinition."		
			Chapter	pg.
			10.3.1	217

A.84 Rule EXT.084: DataInputMustBeReferencedByAnInputSet

Rule #		Conf.Level
EXT.084		proc
Label:	**DataInputMustBeReferencedByAnInputSet**	
	Affected Element:	`DataInput`
	Attribute/Sub Element:	-

Constraint:		A DataInput must be referenced by at least one InputSet.		
	(Pre-) Condition:	-		
	Source:	"A single DataInput MAY be associated with multiple InputSet elements, but it MUST always be referenced by at least one InputSet."		
			Chapter	pg.
			10.3.1	218

A.85 Rule EXT.085: InputSetOptionalInputRefDefinedAsDataInput

Rule #		Conf.Level
EXT.085		proc
Label:	**InputSetOptionalInputRefDefinedAsDataInput**	
	Affected Element:	`InputSet`
	Attribute/Sub Element:	`optionalInputRefs`

Constraint:		An optionalInputRef must be listed as dataInputRef.		
	(Pre-) Condition:	-		
	Source:	"This association MUST NOT reference a DataInput that is not listed in the dataInputRefs."		
			Chapter	pg.
			10.3.1	219

A.86 Rule EXT.086: InputSetWhileExecutingInputRefsDefinedAsDataInput

Rule #		Conf.Level
EXT.086		proc
Label:	**InputSetWhileExecutingInputRefsDefinedAsDataInput**	
	Affected Element:	`InputSet`
	Attribute/Sub Element:	`whileExecutingInputRefs`

Constraint:		A whileExecutingInputRef must be listed as dataInputRef.		
	(Pre-) Condition:	-		
	Source:	"This association MUST NOT reference a DataInput that is not listed in the dataInputRefs."		
			Chapter	pg.
			10.3.1	219

A.87 Rule EXT.087: InputOutputRuleDefinition

Rule #		Conf.Level
EXT.087		proc
Label:	**InputOutputRuleDefinition**	
	Affected Element:	`InputSet/OutputSet`
	Attribute/Sub Element:	`outputSetRefs/inputSetRefs`

Constraint:		Definition of an Input/Output rule: cross referencing between InputSet and OutputSet		
	(Pre-) Condition:	-		
	Source:	"Specifies an Input/Output rule that defines which OutputSet is expected to be created by the Activity when this InputSet became valid. This attribute is paired with the inputSetRefs attribute of OutputSets."		
			Chapter	pg.
			10.3.1	219

A.88 Rule EXT.088: DataOutputMustBeReferencedByAnOutputSet

Rule #		Conf.Level
EXT.088		proc
Label:	**DataOutputMustBeReferencedByAnOutputSet**	
	Affected Element:	`DataOutput`
	Attribute/Sub Element:	-

Constraint:		A DataOutput must be referenced by at least one OutputSet.		
	(Pre-) Condition:	-		
	Source:	"A single DataOutput MAY be associated with multiple OutputSet elements, but it MUST always be referenced by at least one OutputSet."		
			Chapter	pg.
			10.3.1	219

A.89 Rule EXT.089: OutputSetOptionalOutputRefDefinedAsDataOutput

Rule #			Conf.Level	
EXT.089			proc	
Label:	**OutputSetOptionalOutputRefDefinedAsDataOutput**			
	Affected Element:	`OutputSet`		
	Attribute/Sub Element:	`optionalOutputRefs`		
Constraint:	An optionalOutputRef must be listed as dataOutputRef.			
	(Pre-) Condition:	-		
	Source:	"This association MUST NOT reference a DataOutput that is not listed in the dataOutputRefs."		
			Chapter	pg.
			10.3.1	221

A.90 Rule EXT.090: OutputSetWhileExecutingOutputRefsDefinedAsDataOutput

Rule #			Conf.Level	
EXT.090			proc	
Label:	**OutputSetWhileExecutingOutputRefsDefinedAsDataOutput**			
	Affected Element:	`OutputSet`		
	Attribute/Sub Element:	`whileExecutingOutputRefs`		
Constraint:	A whileExecutingOutputRef must be listed as dataOutputRef.			
	(Pre-) Condition:	-		
	Source:	"This association MUST NOT reference a DataOutput that is not listed in the dataOutputRefs."		
			Chapter	pg.
			10.3.1	221

A.91 Rule EXT.091: DataAssociationValidityCheck

Rule #		Conf.Level
EXT.091		proc
Label:	**DataAssociationValidityCheck**	
	Affected Element:	`DataAssociation`
	Attribute/Sub Element:	-

Constraint:		sourceRef and targetRef must have the same ItemDefinition or a transformation must be present.		
	(Pre-) Condition:	-		
	Source:	"The ItemDefinition from the souceRef and targetRef MUST have the same ItemDefinition or the DataAssociation MUST have a transformation Expression that transforms the source ItemDefinition into the target ItemDefinition."		
			Chapter	pg.
			10.3.1	221

A.92 Rule EXT.092: DataAssociationExactlyOneSourceRefIfNoTransformationIsPresent

Rule #		Conf.Level
EXT.092		proc
Label:	**DataAssociationExactlyOneSourceRefIfNoTransformationIsPresent**	
	Affected Element:	`DataAssociation`
	Attribute/Sub Element:	`sourceRef`

Constraint:		A single sourceRef must be present.		
	(Pre-) Condition:	no Transformation defined		
	Source:	"If there is no transformation defined or referenced, then only one source MUST be defined [...]"		
			Chapter	pg.
			10.3.1	222

A.93 Rule EXT.093: EventDataInputOutputConstraints

Rule #		Conf.Level
EXT.093		proc
Label:	**EventDataInputOutputConstraints**	
	Affected Element:	`Event`
	Attribute/Sub Element:	-

Constraint:		For each eventDefinition a DataInput or DataOutput must be defined (depending on the type of the event)	
	(Pre-) Condition:	-	
	Source:	"If the Event is associated with multiple EventDefinitions, there MUST be one Data Input (in the case of throw Events) or one Data Output (in the case of catch Events) for each EventDefinition. The order of the EventDefinitions and the order of the Data Inputs/Outputs determine which Data Input/Output corresponds with which EventDefinition."	
		Chapter	pg.
		10.4.1	235

A.94 Rule EXT.094: EventDataInputOutputItemDefinitionConstraints

Rule #		Conf.Level
EXT.094		proc
Label:	**EventDataInputOutputItemDefinitionConstraints**	
	Affected Element:	`Event`
	Attribute/Sub Element:	-

Constraint:		An itemDefinition must be present for each eventDefintion with its corresponding Data Input/Output.	
	(Pre-) Condition:	-	
	Source:	"For each EventDefinition and Data Input/Output pair, if the Data Input/Output is present, it MUST have an ItemDefinition equivalent to the one defined by the Message, Escalation, Error, or Signal on the associated EventDefinition. In the case of a throw Event, if the Data Input is not present, the Message, Escalation, Error, or Signal will not be populated with data. In the case of a catch Event, if the Data Output is not present, the payload within the Message, Escalation, Error, or Signal will not flow out of the Event and into the Process."	
		Chapter	pg.
		10.4.1	235

A.95 Rule EXT.095: ThrowEventEventDefinitions

Rule #		Conf.Level
EXT.095		proc
Label:	**ThrowEventEventDefinitions**	
	Affected Element:	`ThrowEvent`
	Attribute/Sub Element:	`eventDefinitions`

Constraint:	EventDefinitions defined in a throw event are only valid within these element.		
	(Pre-) Condition:	-	
	Source:	"These EventDefinitions are only valid inside the current Event."	
		Chapter	pg.
		10.4.1	237

A.96 Rule EXT.096: StartEventNoIncomingSequenceFlow

Rule #		Conf.Level
EXT.096		proc
Label:	**StartEventNoIncomingSequenceFlow**	
	Affected Element:	`StartEvent`
	Attribute/Sub Element:	`incoming`

Constraint:	A Start Event must not have an incoming sequence flow.		
	(Pre-) Condition:	-	
	Source:	"[...] the Start Event [...] will not have any incoming Sequence Flows" (p.238) & "A Start Event MUST NOT be a target for Sequence Flows; it MUST NOT have incoming Sequence Flows. An exception to this is when a Start Event is used in an Expanded Sub-Process and is attached to the boundary of that Sub-Process. In this case, a Sequence Flow from the higher-level Process MAY connect to that Start Event in lieu of connecting to the actual boundary of the Sub-Process." (p.245)	
		Chapter	pg.
		10.4.2	238/ 245

A.97 Rule EXT.097: StartEventMandatoryWhenEndEventUsed

Rule #		Conf.Level
EXT.097		proc
Label:	**StartEventMandatoryWhenEndEventUsed**	
	Affected Element:	`StartEvent`
	Attribute/Sub Element:	-

Constraint:	A Start event must be present when an End event is used in the same process level.		
	(Pre-) Condition:	EndEvent used in process.	
	Source:	"If there is an End Event, then there MUST be at least one Start Event."	
		Chapter	pg.
		10.4.2	239

A.98 Rule EXT.098: StartEventTopLevelAllowedEventDefinitions

Rule #			Conf.Level	
EXT.098			proc	
Label:	**StartEventTopLevelAllowedEventDefinitions**			
	Affected Element:	`StartEvent`		
	Attribute/Sub Element:	-		
Constraint:		Only messageEventDefininitions, timerEventDefinitions, conditionalEventDefinitions and signalEventDefinition are allowed for top-level process start events.		
	(Pre-) Condition:	StartEvent is used in a top-level process definition.		
	Source:	"There are seven (7) types of Start Events for top-level Processes in BPMN (see Table 10.84): None, Message, Timer, Conditional, Signal, Multiple, and Parallel." *+ Table 10.84*		
			Chapter	pg.
			10.4.2	240

A.99 Rule EXT.099: CallActivityCalledProcessMandatoryNoneStartEvent

Rule #			Conf.Level	
EXT.099			proc	
Label:	**CallActivityCalledProcessMandatoryNoneStartEvent**			
	Affected Element:	`CallActivity`		
	Attribute/Sub Element:	`calledElement`		
Constraint:		Referenced process must have at least one None Start Event.		
	(Pre-) Condition:	-		
	Source:	"A top-level Process that has at least one None Start Event MAY be called by a Call Activity in another Process. The None Start Event is used for invoking the Process from the Call Activity. All other types of Start Events are only applicable when the Process is used as a top-level Process."		
			Chapter	pg.
			10.4.2	240

A.100 Rule EXT.100: StartEventSubProcessAllowedEventDefinitions

Rule #			Conf.Level	
EXT.100			proc	
Label:	**StartEventSubProcessAllowedEventDefinitions**			
	Affected Element:	`StartEvent`		
	Attribute/Sub Element:	-		
Constraint:		No EventDefinition is allowed for Start Events in Sub-Process definitions.		
	(Pre-) Condition:	StartEvent is used in a sub-process definition.		
	Source:	"There is only one type of Start Event for Sub-Processes in BPMN (see Figure 10.82): None."		
			Chapter	pg.
			10.4.2	241

A.101 Rule EXT.101: StartEventMandatoryOutgoingSequenceFlow

Rule #			Conf.Level	
EXT.101			proc	
Label:	**StartEventMandatoryOutgoingSequenceFlow**			
	Affected Element:	`StartEvent`		
	Attribute/Sub Element:	`outgoing`		
Constraint:		A Start Event MUST be a source for a Sequence Flow.		
	(Pre-) Condition:	-		
	Source:	"A Start Event MUST be a source for a Sequence Flow."		
			Chapter	pg.
			10.4.2	245

A.102 Rule EXT.102: MessageFlowStartEventInvalidSource

Rule #			Conf.Level	
EXT.102			245	
Label:	**MessageFlowStartEventInvalidSource**			
	Affected Element:	`MessageFlow`		
	Attribute/Sub Element:	`sourceRef`		
Constraint:		A Start Event MUST NOT be a source for a message flow.		
	(Pre-) Condition:	-		
	Source:	"A Start Event MUST NOT be a source for a Message Flow; it MUST NOT have outgoing Message Flows."		
			Chapter	pg.
			10.4.2	245

A.103 Rule EXT.103: StartEventMessageEventDefinintionFlow-Constraint

Rule #		Conf.Level
EXT.103		proc
Label:	**StartEventMessageEventDefinintionFlowConstraint**	
	Affected Element:	`StartEvent`
	Attribute/Sub Element:	`messageEventDefinition`

Constraint:		If a Start Event is target of a MessageFlow definition, at least one messageEvent-Definition must be present.		
	(Pre-) Condition:	The start event is target of a message flow definition.		
	Source:	*implicit*		
			Chapter	pg.
			10.4.2	

A.104 Rule EXT.104: EndEventNoOutgoingSequenceFlow

Rule #		Conf.Level
EXT.104		proc
Label:	**EndEventNoOutgoingSequenceFlow**	
	Affected Element:	`EndEvent`
	Attribute/Sub Element:	`outgoing`

Constraint:		An End Event must not have an outgoing sequence flow.		
	(Pre-) Condition:	-		
	Source:	"[...] the End Event [...] will not have any outgoing Sequence Flows" (p.246) & "An End Event MUST NOT be a source for Sequence Flows; that is, there MUST NOT be outgoing Sequence Flows. An exception to this is when an End Event is used in an Expanded Sub-Process and is attached to the boundary of that Sub-Process. In this case, a Sequence Flow from the higher-level Process MAY connect from that End Event in lieu of connecting from the actual boundary of the Sub-Process." (p.249)		
			Chapter	pg.
			10.4.3	246/ 249

A.105 Rule EXT.105: EndEventMandatoryWhenStartEventUsed

Rule #		Conf.Level
EXT.105		proc
Label:	**EndEventMandatoryWhenStartEventUsed**	
	Affected Element:	`EndEvent`
	Attribute/Sub Element:	-

Constraint:		An end event must be present when a start event is used in the same process level.		
	(Pre-) Condition:	StartEvent used in process.		
	Source:	"If there is a Start Event, then there MUST be at least one End Event."		
			Chapter	pg.
			10.4.3	247

A.106 Rule EXT.106: EndEventCancelOnlyAllowedInTransaction

Rule #		Conf.Level
EXT.106		proc
Label:	**EndEventCancelOnlyAllowedInTransaction**	
	Affected Element:	`EndEvent`
	Attribute/Sub Element:	`cancelEventDefinition`

Constraint:		A cancel EndEvent is only allowed in a transaction sub-process.		
	(Pre-) Condition:	-		
	Source:	"This type of End is used within a Transaction Sub-Process." (p.248) & "The Cancel End Event MUST only be used within a Transaction Sub-Process and, thus, MAY NOT be used in any other type of Sub-Process or Process." (p.263)		
			Chapter	pg.
			10.4.3	248/ 263

A.107 Rule EXT.107: EndEventMandatoryIncomingSequenceFlow

Rule #		Conf.Level
EXT.107		proc
Label:	**EndEventMandatoryIncomingSequenceFlow**	
	Affected Element:	`EndEvent`
	Attribute/Sub Element:	`incoming`

Constraint:		An End Event MUST have at least one incoming Sequence Flow.		
	(Pre-) Condition:	-		
	Source:	"An End Event MUST be a target for a Sequence Flow. An End Event MAY have multiple incoming Sequence Flows."		
			Chapter	pg.
			10.4.3	249

A.108 Rule EXT.108: MessageFlowEndEventInvalidTarget

Rule #		Conf.Level
EXT.108		proc
Label:	**MessageFlowEndEventInvalidTarget**	
	Affected Element:	`MessageFlow`
	Attribute/Sub Element:	`targetRef`

Constraint:		An End Event MUST NOT be a target for a message flow.		
	(Pre-) Condition:	-		
	Source:	"An End Event MUST NOT be the target of a Message Flow; it can have no incoming Message Flows."		
			Chapter	pg.
			10.4.3	249

A.109 Rule EXT.109: EndEventMessageEventDefinintionFlowConstraint

Rule #		Conf.Level
EXT.109		proc
Label:	**EndEventMessageEventDefinintionFlowConstraint**	
	Affected Element:	`EndEvent`
	Attribute/Sub Element:	`messageEventDefinition`

Constraint:		If an end event is source of a MessageFlow definition, at least one messageEventDefinition must be present.		
	(Pre-) Condition:	The end event is source of a message flow definition.		
	Source:	"The Result attribute of the End Event MUST be set to Message or Multiple if there are any outgoing Message Flows. The Result attribute of the End Event MUST be set to Multiple if there is more than one outgoing Message Flows."		
			Chapter	pg.
			10.4.3	249

A.110 Rule EXT.110: BoundaryEventCancelActivityValueRestrictions

Rule #		Conf.Level
EXT.110		proc
Label:	**BoundaryEventCancelActivityValueRestrictions**	
	Affected Element:	`BoundaryEvent`
	Attribute/Sub Element:	`cancelActivity`

Constraint:		The allowed values for the attribute depend on the used EventDefinitions. (see Table 10.92 in [10, p.258] for details.)		
	(Pre-) Condition:	-		
	Source:	*Table 10.92*		
			Chapter	pg.
			10.4.4	258

A.111 Rule EXT.111: BoundaryEventSubProcessCancelRequires-Transaction

Rule #		Conf.Level
EXT.111		proc
Label:	**BoundaryEventSubProcessCancelRequiresTransaction**	
	Affected Element:	`BoundaryEvent`
	Attribute/Sub Element:	`-`

Constraint:		The Transaction attribute of a Sub-Process with an attached CancelBoundaryEvent must be true.		
	(Pre-) Condition:	BoundaryEvent type is cancel and is attached to a Sub-Process.		
	Source:	"An Intermediate Event with a Cancel trigger MAY be attached to a Sub-Process boundary only if the Transaction attribute of the Sub-Process is set to true."		
			Chapter	pg.
			10.4.4	259

A.112 Rule EXT.112: BoundaryEventNoIncomingSequenceFlow

Rule #		Conf.Level
EXT.112		proc
Label:	**BoundaryEventNoIncomingSequenceFlow**	
	Affected Element:	`BoundaryEvent`
	Attribute/Sub Element:	`incoming`

Constraint:		A boundary event must not be target of a Sequence Flow.		
	(Pre-) Condition:	-		
	Source:	"If the Intermediate Event is attached to the boundary of an Activity: The Intermediate Event MUST NOT be a target for a Sequence Flow; it cannot have an incoming Sequence Flows."		
			Chapter	pg.
			10.4.4	259

A.113 Rule EXT.113: BoundaryEventMandatoryOutgoingSequenceFlow

Rule #			Conf.Level
EXT.113			proc
Label:	**BoundaryEventMandatoryOutgoingSequenceFlow**		
	Affected Element:	`BoundaryEvent`	
	Attribute/Sub Element:	`outgoing`	

Constraint:		A boundary event must be a source of at least a SequenceFlow.	
	(Pre-) Condition:	BoundaryEvent type does not contain a compensateEventDefinition.	
	Source:	"[If the Intermediate Event is attached to the boundary of an Activity:] The Intermediate Event MUST be a source for a Sequence Flow. Multiple Sequence Flows MAY originate from an Intermediate Event. [...] An exception to this: an Intermediate Event with a Compensation trigger MUST NOT have an outgoing Sequence Flow (it MAY have an outgoing Association)."	
		Chapter	pg.
		10.4.4	259

A.114 Rule EXT.114: BoundaryEventCompensateNoOutgoingSequenceFlow

Rule #			Conf.Level
EXT.114			proc
Label:	**BoundaryEventCompensateNoOutgoingSequenceFlow**		
	Affected Element:	`BoundaryEvent`	
	Attribute/Sub Element:	`outgoing`	

Constraint:		A compensation boundary event MUST NOT have an outgoing Sequence Flow (it MAY have an outgoing Association).	
	(Pre-) Condition:	BoundaryEvent contains a compensateEventDefinition.	
	Source:	"[If the Intermediate Event is attached to the boundary of an Activity:] The Intermediate Event MUST be a source for a Sequence Flow. [...] An exception to this: an Intermediate Event with a Compensation trigger MUST NOT have an outgoing Sequence Flow (it MAY have an outgoing Association)."	
		Chapter	pg.
		10.4.4	259

A.115 Rule EXT.115: IntermediateEventMandatoryIncomingSequenceFlow

Rule #			Conf.Level
EXT.115			proc
Label:	**IntermediateEventMandatoryIncomingSequenceFlow**		
	Affected Element:	`IntermediateEvent`	
	Attribute/Sub Element:	`incoming`	

Constraint:		Intermediate Events MUST be a target of at least a Sequence Flow.		
	(Pre-) Condition:	IntermediateEvent does not contain a link.		
	Source:	"If the Intermediate Event is used within normal flow: Intermediate Events MUST be a target of a Sequence Flow. [...] An Intermediate Event MAY have multiple incoming Sequence Flows."		
			Chapter	pg.
			10.4.4	259

A.116 Rule EXT.116: IntermediateEventMandatoryOutgoingSequenceFlow

Rule #			Conf.Level
EXT.116			proc
Label:	**IntermediateEventMandatoryOutgoingSequenceFlow**		
	Affected Element:	`IntermediateEvent`	
	Attribute/Sub Element:	`outgoing`	

Constraint:		Intermediate Events MUST be a source of at least a Sequence Flow.		
	(Pre-) Condition:	IntermediateEvent does not contain a link.		
	Source:	"An Intermediate Event MUST be a source for a Sequence Flow. Multiple Sequence Flows MAY originate from an Intermediate Event. For each sequence Flow that has the Intermediate Event as a source, a new parallel path SHALL be generated. An exception to this: a source Link Intermediate Event (as defined below), it is NOT REQUIRED to have an outgoing Sequence Flow."		
			Chapter	pg.
			10.4.4	259

A.117 Rule EXT.117: LinkIntermediateEventSequenceFlowConstraint

Rule #		Conf.Level
EXT.117		proc
Label:	**LinkIntermediateEventSequenceFlowConstraint**	
	Affected Element:	`IntermediateEvent`
	Attribute/Sub Element:	`incoming/outgoing`

Constraint:		A Link Intermediate Event MUST NOT be both a target and a source of a Sequence Flow.	
	(Pre-) Condition:	IntermediateEvent contains a link.	
	Source:	"A Link Intermediate Event MUST NOT be both a target and a source of a Sequence Flow." "A Link Intermediate Event MAY be the target (target Link) or a source (source Link) of a Sequence Flow, but MUST NOT be both a target and a source."	
		Chapter	pg.
		10.4.4	259

A.118 Rule EXT.118: LinkIntermediateEventSourceTargetMatching

Rule #		Conf.Level
EXT.118		proc
Label:	**LinkIntermediateEventSourceTargetMatching**	
	Affected Element:	`IntermediateEvent`
	Attribute/Sub Element:	-

Constraint:		For each source Link there must exist a correspondig target. There may be multiple sources for one target.	
	(Pre-) Condition:	IntermediateEvent contains a "link".	
	Source:	"If there is a source Link, there MUST be a matching target Link (they have the same name). There MAY be multiple source Links for a single target Link. There MUST NOT be multiple target Links for a single source Link."	
		Chapter	pg.
		10.4.4	259

A.119 Rule EXT.119: MessageIntermediateEventMessageFlowConstraint

Rule #		Conf.Level
EXT.119		proc
Label:	**MessageIntermediateEventMessageFlowConstraint**	
	Affected Element:	`IntermediateEvent`
	Attribute/Sub Element:	-

Constraint:		A Message Intermediate Event MAY have an incoming Message Flow or an outgoing Message Flow, but not both.	
	(Pre-) Condition:	Intermediate Event contains a message	
	Source:	"- A Message Intermediate Event MAY be the target for a Message Flow; it can have one incoming Message Flow. - A Message Intermediate Event MAY be a source for a Message Flow; it can have one outgoing Message Flow. - A Message Intermediate Event MAY have an incoming Message Flow or an outgoing Message Flow, but not both."	
		Chapter	pg.
		10.4.4	260

A.120 Rule EXT.120: CancelIntermediateEventUsageConstraint

Rule #		Conf.Level
EXT.120		proc
Label:	**CancelIntermediateEventUsageConstraint**	
	Affected Element:	`BoundaryEvent`
	Attribute/Sub Element:	`attachedToRef`

Constraint:		A cancel Intermediate Event must be attached to a Transaction Sub-Process. (i.e. the attachedToRef of the BoundaryEvent must point to a Transaction-Element)	
	(Pre-) Condition:	BoundaryEvent contains a cancelEventDefinition	
	Source:	"The catch Cancel Intermediate Event MUST only be attached to the boundary of a Transaction Sub-Process and, thus, MAY NOT be used in normal flow."	
		Chapter	pg.
		10.4.5	263

A.121 Rule EXT.121: CompensateStartEventUsageConstraint

Rule #		Conf.Level
EXT.121		proc
Label:	**CompensateStartEventUsageConstraint**	
	Affected Element:	`StartEvent`
	Attribute/Sub Element:	

Constraint:		A compensation start event is only allowed for Event Sub-Processes.		
	(Pre-) Condition:	StartEvent contains a compensateEventDefinition		
	Source:	"The Compensation Start Event MAY NOT be used for a top-level Process. The Compensation Start Event MAY be used for an Event Sub-Process."		
			Chapter	pg.
			10.4.5	263

A.122 Rule EXT.122: CompensateIntermediateEventCatchUsage-Constraint

Rule #		Conf.Level
EXT.122		proc
Label:	**CompensateIntermediateEventCatchUsageConstraint**	
	Affected Element:	`BoundaryEvent`
	Attribute/Sub Element:	`attachedToRef`

Constraint:		The catch Compensation Intermediate Event MUST only be attached to the boundary of an Activity and, thus, MAY NOT be used in normal flow. The throw Compensation Intermediate Event MAY be used in normal flow.		
	(Pre-) Condition:	BoundaryEvent contains a compensateEventDefinition		
	Source:	"The catch Compensation Intermediate Event MUST only be attached to the boundary of an Activity and, thus, MAY NOT be used in normal flow. The throw Compensation Intermediate Event MAY be used in normal flow."		
			Chapter	pg.
			10.4.5	263

A.123 Rule EXT.123: ConditionalEventDefinitionFormalConditionMandatoryForExecutability

Rule #			Conf.Level	
EXT.123			proc	
Label:	**ConditionalEventDefinitionFormalConditionMandatoryForExecutability**			
	Affected Element:	`conditionalEventDefinition`		
	Attribute/Sub Element:	`condition`		
Constraint:		For executable Processes (isExecutable = true), if the trigger is Conditional, then a FormalExpression MUST be entered.		
	(Pre-) Condition:	process is defined as executable		
	Source:	"For executable Processes (isExecutable = true), if the trigger is Conditional, then a FormalExpression MUST be entered."		
			Chapter	pg.
			10.4.5	265

A.124 Rule EXT.124: LinkEventDefinitionSourceRequiredForCatchEvents

Rule #			Conf.Level	
EXT.124			proc	
Label:	**LinkEventDefinitionSourceRequiredForCatchEvents**			
	Affected Element:	`LinkEventDefinition`		
	Attribute/Sub Element:	`source`		
Constraint:		A LinkEventDefinition in a Catch Event must have at least one source Element.		
	(Pre-) Condition:	LinkEventDefinition is used in an IntermediateCatchEvent		
	Source:	***Issue:*** *Wrong Text and multiplicity in Standard*[13]		
			Chapter	pg.
			10.4.5	270

[13]see official issue: `http://www.omg.org/issues/bpmn2-rtf.open.html#Issue15739)`

A.125 Rule EXT.125: LinkEventDefinitionTargetRequiredForThrowEvents

Rule #			Conf.Level
EXT.125			proc
Label:	**LinkEventDefinitionTargetRequiredForThrowEvents**		
	Affected Element:	`LinkEventDefinition`	
	Attribute/Sub Element:	`target`	

Constraint:		A LinkEventDefinition in a Throw Event must have exactly one target Element.	
	(Pre-) Condition:	LinkEventDefinition is used in an IntermediateThrowEvent	
	Source:	***Issue:*** *Wrong Text and multiplicity in Standard*[14]	
		Chapter	pg.
		10.4.5	270

A.126 Rule EXT.126: LinkEventDefinitionLinkValidityConstraint

Rule #			Conf.Level
EXT.126			proc
Label:	**LinkEventDefinitionLinkValidityConstraint**		
	Affected Element:	`LinkEventDefinition`	
	Attribute/Sub Element:	`source/target`	

Constraint:		Links are only allowed to a target in the same process and process level.	
	(Pre-) Condition:	-	
	Source:	"The use of Link Events is limited to a single Process level (i.e., they cannot link a parent Process with a Sub-Process)."	
		Chapter	pg.
		10.4.5	267

A.127 Rule EXT.127: MessageEventDefinitionMessageRefMandatoryForExecutability

Rule #			Conf.Level
EXT.127			proc
Label:	**MessageEventDefinitionMessageRefMandatoryForExecutability**		
	Affected Element:	`MessageEventDefinition`	
	Attribute/Sub Element:	`messageRef`	

Constraint:		A messageRef must be present if the process should be executable.	
	(Pre-) Condition:	process is defined asexecutable	
	Source:	"The Message MUST be supplied (if the isExecutable attribute of the Process is set to true)."	
		Chapter	pg.
		10.4.5	271

A.128 Rule EXT.128: MessageEventDefinitionOperationRefMandatoryForExecutability

Rule #			Conf.Level	
EXT.128			proc/exec	
Label:	**MessageEventDefinitionOperationRefMandatoryForExecutability**			
	Affected Element:	`MessageEventDefinition`		
	Attribute/Sub Element:	`operationRef`		
Constraint:		An operationRef must be present if the process should be executable.		
	(Pre-) Condition:	process is defined as executable		
	Source:	"It MUST be specified for executable Processes."		
			Chapter	pg.
			10.4.5	271

A.129 Rule EXT.129: NoneStartEventNotAllowedForEventSubProcesses

Rule #			Conf.Level	
EXT.129			proc	
Label:	**NoneStartEventNotAllowedForEventSubProcesses**			
	Affected Element:	`StartEvent`		
	Attribute/Sub Element:	-		
Constraint:		The None Start Event MAY NOT be used for an Event Sub-Process. (i.e. there must exist at least one EventDefinition)		
	(Pre-) Condition:	StartEvent is used in an Event Sub-Process.		
	Source:	"The None Start Event MAY NOT be used for an Event Sub-Process."		
			Chapter	pg.
			10.4.5	272

A.130 Rule EXT.130: NoneCatchBoundaryEventNotAllowed

Rule #			Conf.Level	
EXT.130			proc	
Label:	**NoneCatchBoundaryEventNotAllowed**			
	Affected Element:	`BoundaryEvent`		
	Attribute/Sub Element:	-		
Constraint:		A catch None BoundaryEvent is not allowed.		
	(Pre-) Condition:	BoundaryEvent is of type catch		
	Source:	"The catch None Intermediate Event MUST only be used in normal flow and, thus, MAY NOT be attached to the boundary of an Activity."		
			Chapter	pg.
			10.4.5	272

A.131 Rule EXT.131: TimerEventDefinitionOnlyOneAttributeAllowed

Rule #		Conf.Level
EXT.131		proc/exec
Label:	**TimerEventDefinitionOnlyOneAttributeAllowed**	
	Affected Element:	`TimerEventDefinition`
	Attribute/Sub Element:	-

Constraint:		Timer attributes are mutually exclusive, i.e., only one of the Attributes timeDate, timeCycle and timeDuration might be set for executable processes.	
	(Pre-) Condition:	process is defined as executable	
	Source:	"Timer attributes are mutually exclusive [...] (if the isExecutable attribute of the Process is set to true)"	
		Chapter	pg.
		10.4.5	273

A.132 Rule EXT.132: TimerEventTimeDateFormatCheck

Rule #		Conf.Level
EXT.132		proc
Label:	**TimerEventTimeDateFormatCheck**	
	Affected Element:	`TimerEventDefinition`
	Attribute/Sub Element:	`timeDate`

Constraint:		The return type of the attribute timeDate MUST conform to the ISO-8601 format for date and time representations.	
	(Pre-) Condition:	-	
	Source:	"The return type of the attribute timeDate MUST conform to the ISO-8601 format for date and time representations."	
		Chapter	pg.
		10.4.5	273

A.133 Rule EXT.133: TimerEventTimeCycleFormatCheck

Rule #		Conf.Level
EXT.133		proc
Label:	**TimerEventTimeCycleFormatCheck**	
	Affected Element:	`TimerEventDefinition`
	Attribute/Sub Element:	`timeCycle`

Constraint:		The return type of the attribute timeCycle MUST conform to the ISO-8601 format for date and time representations.	
	(Pre-) Condition:	-	
	Source:	"The return type of the attribute timeCycle MUST conform to the ISO-8601 format for date and time representations."	
		Chapter	pg.
		10.4.5	273

A.134 Rule EXT.134: TimerEventTimeDurationFormatCheck

Rule #			Conf.Level	
EXT.134			proc	
Label:	**TimerEventTimeDurationFormatCheck**			
	Affected Element:	`TimerEventDefinition`		
	Attribute/Sub Element:	`timeDuration`		
Constraint:		The return type of the attribute timeDuration MUST conform to the ISO-8601 format for date and time representations.		
	(Pre-) Condition:	-		
	Source:	"The return type of the attribute timeDuration MUST conform to the ISO-8601 format for date and time representations."		
			Chapter	pg.
			10.4.5	273

A.135 Rule EXT.135: GatewayGeneralSequenceFlowConstraint

Rule #			Conf.Level	
EXT.135			proc	
Label:	**GatewayGeneralSequenceFlowConstraint**			
	Affected Element:	`Gateway`		
	Attribute/Sub Element:	`incoming/outgoing`		
Constraint:		A Gateway MUST have either multiple incoming Sequence Flows or multiple outgoing Sequence Flows.		
	(Pre-) Condition:	-		
	Source:	"A Gateway MUST have either multiple incoming Sequence Flows or multiple outgoing Sequence Flows (i.e., it MUST merge or split the flow)."		
			Chapter	pg.
			10.5.1	290

A.136 Rule EXT.136: EventBasedGatewayOutgoingSequenceFlowCardinalityConstraint

Rule #			Conf.Level	
EXT.136			proc	
Label:	**EventBasedGatewayOutgoingSequenceFlowCardinalityConstraint**			
	Affected Element:	`EventBasedGateway`		
	Attribute/Sub Element:	`outgoing`		
Constraint:		An Event Gateway MUST have two or more outgoing Sequence Flows.		
	(Pre-) Condition:	-		
	Source:	"An Event Gateway MUST have two or more outgoing Sequence Flows."		
			Chapter	pg.
			10.5.6	297

A.137 Rule EXT.137: EventBasedGatewayNoConditionForSequenceFlows

Rule #			Conf.Level
EXT.137			proc
Label:	**EventBasedGatewayNoConditionForSequenceFlows**		
	Affected Element:	`EventBasedGateway`	
	Attribute/Sub Element:	`outgoing`	

Constraint:		The outgoing Sequence Flows of the Event Gateway MUST NOT have a conditionExpression.	
	(Pre-) Condition:	-	
	Source:	"The outgoing Sequence Flows of the Event Gateway MUST NOT have a conditionExpression."	
		Chapter	pg.
		10.5.6	297

A.138 Rule EXT.138: EventBasedGatewayComplexSequenceFlowConstraint

Rule #			Conf.Level
EXT.138			proc
Label:	**EventBasedGatewayComplexSequenceFlowConstraint**		
	Affected Element:	`EventBasedGateway`	
	Attribute/Sub Element:	`outgoing`	

Constraint:		An eventBasedGateway may only be connected to an ReceiveTask or one of the following intermediate Events: Message, Signal, Timer, Conditional, and Multiple (which can only include the previous triggers)	
	(Pre-) Condition:	-	
	Source:	"Event-Based Gateways are configured by having outgoing Sequence Flows target an Intermediate Event or a Receive Task in any combination [...] Only the following Intermediate Event triggers are valid: Message, Signal, Timer, Conditional, and Multiple (which can only include the previous triggers). Thus, the following Intermediate Event triggers are not valid: Error, Cancel, Compensation, and Link."	
		Chapter	pg.
		10.5.6	298

A.139 Rule EXT.139: EventBasedGatewayNoMixtureOfReceiveTasksAndMessageEvents

Rule #		Conf.Level
EXT.139		proc
Label:	**EventBasedGatewayNoMixtureOfReceiveTasksAndMessageEvents**	
	Affected Element:	`EventBasedGateway`
	Attribute/Sub Element:	-

Constraint:		If Message Intermediate Events are used in the configuration, then Receive Tasks MUST NOT be used in that configuration and vice versa.	
	(Pre-) Condition:	-	
	Source:	"If Message Intermediate Events are used in the configuration, then Receive Tasks MUST NOT be used in that configuration and vice versa."	
		Chapter	pg.
		10.5.6	298

A.140 Rule EXT.140: EventBasedGatewayNoBoundaryEventsForReceiveTasks

Rule #		Conf.Level
EXT.140		proc
Label:	**EventBasedGatewayNoBoundaryEventsForReceiveTasks**	
	Affected Element:	`EventBasedGateway`
	Attribute/Sub Element:	-

Constraint:		Receive Tasks used in an Event Gateway configuration MUST NOT have any attached Intermediate Events.	
	(Pre-) Condition:	-	
	Source:	"Receive Tasks used in an Event Gateway configuration MUST NOT have any attached Intermediate Events."	
		Chapter	pg.
		10.5.6	298

A.141 Rule EXT.141: EventBasedGatewayNoAdditionalSequenceFlowsForTargets

Rule #		Conf.Level
EXT.141		proc
Label:	**EventBasedGatewayNoAdditionalSequenceFlowsForTargets**	
	Affected Element:	`EventBasedGateway`
	Attribute/Sub Element:	-

Constraint:		Targets of an EventBasedGateway must not have any other incoming SequenceFlow.	
	(Pre-) Condition:	-	
	Source:	"Target elements in an Event Gateway configuration MUST NOT have any additional incoming Sequence Flows (other than that from the Event Gateway)."	
		Chapter	pg.
		10.5.6	298

A.142 Rule EXT.142: EventBasedGatewayInstantiationNoIncomingSequenceFlow

Rule #		Conf.Level
EXT.142		proc
Label:	**EventBasedGatewayInstantiationNoIncomingSequenceFlow**	
	Affected Element:	`EventBasedGateway`
	Attribute/Sub Element:	`incoming`

Constraint:		When an EventBasedGateway is used to instantiate a process instance no Incoming Sequence Flow is allowed.	
	(Pre-) Condition:	Attribute instantiate is true	
	Source:	"In order for an Event Gateway to instantiate a Process, it MUST not have any incoming Sequence Flows."	
		Chapter	pg.
		10.5.6	299

A.143 Rule EXT.143: CompensateBoundaryEventAssociationRequired

Rule #		Conf.Level
EXT.143		proc
Label:	**CompensateBoundaryEventAssociationRequired**	
	Affected Element:	`BoundaryEvent`
	Attribute/Sub Element:	-

Constraint:		A compensateBoundaryEvent must be connected with an Association to a Compensation Activity.	
	(Pre-) Condition:	BoundaryEvent has a compensateEventDefinition	
	Source:	"This compensation is modeled with a specialized Compensation Activity, which is connected to the boundary Event through an Association"	
		Chapter	pg.
		10.6.1	303

A.144 Rule EXT.144: CompensateBoundaryEventAssociationToCompensationActivity

Rule #		Conf.Level
EXT.144		proc
Label:	**CompensateBoundaryEventAssociationToCompensationActivity**	
	Affected Element:	`BoundaryEvent`
	Attribute/Sub Element:	-

Constraint:		The associated Activity must be a Task or a Sub-Process which is marked for compensation (i.e., isForCompensation=true)	
	(Pre-) Condition:	BoundaryEvent has a compensateEventDefinition	
	Source:	"The Compensation Activity, which can be either a Task or a Sub-Process, has a marker to show that it is used for compensation only and is outside the normal flow of the Process."	
		Chapter	pg.
		10.6.1	303

A.145 Rule EXT.145: ScriptTaskScriptFormatValidValueCheck

Rule #		Conf.Level
EXT.145		proc
Label:	**ScriptTaskScriptFormatValidValueCheck**	
	Affected Element:	`ScriptTask`
	Attribute/Sub Element:	`scriptFormat`

Constraint:		The value must be in mime-type format.		
	(Pre-) Condition:	-		
	Source:	"This attribute value MUST be specified with a mime-type format."		
			Chapter	pg.
			10.2.3.1	165

A.146 Rule EXT.146: EndEventAllowedEventDefinitions

Rule #		Conf.Level
EXT.146		proc
Label:	**EndEventAllowedEventDefinitions**	
	Affected Element:	`EndEvent`
	Attribute/Sub Element:	-

Constraint:		Only messageEventDefininitions, escalationEventDefinitions, errorEventDefinitions, cancelEventDefinitions, compensationEventDefinitions, signalEventDefinitions and terminateEventDefinitions are allowed for end events.		
	(Pre-) Condition:	-		
	Source:	"There are nine types of End Events in BPMN: None, Message, Escalation, Error, Cancel, Compensation, Signal, Terminate, and Multiple." & Table 10.88		
			Chapter	pg.
			10.4.3	247-249

A.147 Rule EXT.147: BoundaryEventAllowedEventDefinitions

Rule #		Conf.Level
EXT.147		proc
Label:	**BoundaryEventAllowedEventDefinitions**	
	Affected Element:	`BoundaryEvent`
	Attribute/Sub Element:	-

Constraint:		Only messageEventDefininitions, timerEventDefinitions, escalationEventDefinitions, errorEventDefinitions, cancelEventDefinitions, compensationEventDefinitions, conditionalEventDefinitions and signalEventDefinitions are allowed for boundary events.		
	(Pre-) Condition:	-		
	Source:	(Table 10.90)		
			Chapter	pg.
			10.4.4	254-257

A.148 Rule EXT.148: IntermediateCatchEventAllowedEventDefinitions

Rule #			Conf.Level	
EXT.148			proc	
Label:	**IntermediateCatchEventAllowedEventDefinitions**			
	Affected Element:	`IntermediateEvent`		
	Attribute/Sub Element:	-		
Constraint:		Only messageEventDefininitions, timerEventDefinitions, conditionalEventDefinitions, linkEventDefinitions and signalEventDefinitions are allowed for intermediate catch events.		
	(Pre-) Condition:	-		
	Source:	(Table 10.89)		
			Chapter	pg.
			10.4.4	251-254

A.149 Rule EXT.149: IntermediateThrowEventAllowedEventDefinitions

Rule #			Conf.Level	
EXT.149			proc	
Label:	**IntermediateThrowEventAllowedEventDefinitions**			
	Affected Element:	`IntermediateEvent`		
	Attribute/Sub Element:	-		
Constraint:		Only messageEventDefininitions, escalationEventDefinitions, compensationEventDefinitions, linkEventDefinitions and signalEventDefinition are allowed for intermediate throw events.		
	(Pre-) Condition:	-		
	Source:	(Table 10.89)		
			Chapter	pg.
			10.4.4	251-254

A.150 Rule EXT.150: StartEventUsageSequenceFlowImplications

Rule #		Conf.Level
EXT.150		proc
Label:	**StartEventUsageSequenceFlowImplications**	
	Affected Element:	`StartEvent`
	Attribute/Sub Element:	-

Constraint:		If a start event is used to initiate a process, all flow nodes (besides start events, boundary events and catching Link events, compensation activies and event subprocesses) must have an incoming sequence flow.	
	(Pre-) Condition:	StartEvent is used in a process level.	
	Source:	"A Start Event is OPTIONAL: a Process level [...] MAY (is NOT REQUIRED to) have a Start Event." (p.238) "All Flow Objects that do not have an incoming Sequence Flow (i.e., are not a target of a Sequence Flow) SHALL be instantiated when the Process is instantiated. Exceptions to this are Activities that are defined as being Compensation Activities [...] catching Link Intermediate Event[s] [...] Event Sub-Process[es]" (p.239)	
		Chapter	pg.
		10.4.2	238-239

A.151 Rule EXT.151: EndEventUsageSequenceFlowImplications

Rule #		Conf.Level
EXT.151		proc
Label:	**EndEventUsageSequenceFlowImplications**	
	Affected Element:	`EndEvent`
	Attribute/Sub Element:	-

Constraint:		If end events are used, all flow nodes other (besides end events, throwing Link events, compensating activities and event subprocesses) must have an outgoing sequence flow.	
	(Pre-) Condition:	EndEvent is used in a process level.	
	Source:	"An End Event is OPTIONAL: a given Process level [...] MAY (is NOT REQUIRED to) have this shape [...] If the End Event is not used, then all Flow Objects that do not have any outgoing Sequence Flow (i.e., are not a source of a Sequence Flow) mark the end of a path in the Process. However, the Process MUST NOT end until all parallel paths have completed." (p.246-247)	
		Chapter	pg.
		10.4.3	246-247

A.152 Rule EXT.152: SequenceFlowNoCrossingOfSubProcessBorder

Rule #			Conf.Level
EXT.152			proc
Label:	**SequenceFlowNoCrossingOfSubProcessBorder**		
	Affected Element:	`SequenceFlow`	
	Attribute/Sub Element:	-	

Constraint:		A Sequence Flow must not cross the border of a SubProcess, but must link to the SubProcess itself (i.e., a Sequence flow must not to elements within a subprocess)		
	(Pre-) Condition:	-		
	Source:	*implicit (further source: [16, p.137; rule #5])*		
			Chapter	pg.
			-	-

B List of previous University of Bamberg reports

Bamberger Beiträge zur Wirtschaftsinformatik

Nr. 1 (1989) Augsburger W., Bartmann D., Sinz E.J.: Das Bamberger Modell: Der Diplom-Studiengang Wirtschaftsinformatik an der Universität Bamberg (Nachdruck Dez. 1990)

Nr. 2 (1990) Esswein W.: Definition, Implementierung und Einsatz einer kompatiblen Datenbankschnittstelle für PROLOG

Nr. 3 (1990) Augsburger W., Rieder H., Schwab J.: Endbenutzerorientierte Informationsgewinnung aus numerischen Daten am Beispiel von Unternehmenskennzahlen

Nr. 4 (1990) Ferstl O.K., Sinz E.J.: Objektmodellierung betrieblicher Informationsmodelle im Semantischen Objektmodell (SOM) (Nachdruck Nov. 1990)

Nr. 5 (1990) Ferstl O.K., Sinz E.J.: Ein Vorgehensmodell zur Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell (SOM)

Nr. 6 (1991) Augsburger W., Rieder H., Schwab J.: Systemtheoretische Repräsentation von Strukturen und Bewertungsfunktionen über zeitabhängigen betrieblichen numerischen Daten

Nr. 7 (1991) Augsburger W., Rieder H., Schwab J.: Wissensbasiertes, inhaltsorientiertes Retrieval statistischer Daten mit EISREVU / Ein Verarbeitungsmodell für eine modulare Bewertung von Kennzahlenwerten für den Endanwender

Nr. 8 (1991) Schwab J.: Ein computergestütztes Modellierungssystem zur Kennzahlenbewertung

Nr. 9 (1992) Gross H.-P.: Eine semantiktreue Transformation vom Entity-Relationship-Modell in das Strukturierte Entity-Relationship-Modell

Nr. 10 (1992) Sinz E.J.: Datenmodellierung im Strukturierten Entity-Relationship-Modell (SERM)

Nr. 11 (1992) Ferstl O.K., Sinz E. J.: Glossar zum Begriffsystem des Semantischen Objektmodells

Nr. 12 (1992) Sinz E. J., Popp K.M.: Zur Ableitung der Grobstruktur des konzeptuellen Schemas aus dem Modell der betrieblichen Diskurswelt

Nr. 13 (1992) Esswein W., Locarek H.: Objektorientierte Programmierung mit dem Objekt-Rollenmodell

Nr. 14 (1992) Esswein W.: Das Rollenmodell der Organsiation: Die Berücksichtigung aufbauorganisatorische Regelungen in Unternehmensmodellen

Nr. 15 (1992) Schwab H. J.: EISREVU-Modellierungssystem. Benutzerhandbuch

Nr. 16 (1992) Schwab K.: Die Implementierung eines relationalen DBMS nach dem Client/Server-Prinzip

Nr. 17 (1993) Schwab K.: Konzeption, Entwicklung und Implementierung eines computergestützten Bürovorgangssystems zur Modellierung von Vorgangsklassen und Abwicklung und Überwachung von Vorgängen. Dissertation

Nr. 18 (1993) Ferstl O.K., Sinz E.J.: Der Modellierungsansatz des Semantischen Objektmodells

Nr. 19 (1994) Ferstl O.K., Sinz E.J., Amberg M., Hagemann U., Malischewski C.: Tool-Based Business Process Modeling Using the SOM Approach

Nr. 20 (1994) Ferstl O.K., Sinz E.J.: From Business Process Modeling to the Specification of Distributed Business Application Systems - An Object-Oriented Approach -. 1st edition, June 1994

Ferstl O.K., Sinz E.J. : Multi-Layered Development of Business Process Models and Distributed Business Application Systems - An Object-Oriented Approach -. 2nd edition, November 1994

Nr. 21 (1994) Ferstl O.K., Sinz E.J.: Der Ansatz des Semantischen Objektmodells zur Modellierung von Geschäftsprozessen

Nr. 22 (1994) Augsburger W., Schwab K.: Using Formalism and Semi-Formal Constructs for Modeling Information Systems

Nr. 23 (1994) Ferstl O.K., Hagemann U.: Simulation hierarischer objekt- und transaktionsorientierter Modelle

Nr. 24 (1994) Sinz E.J.: Das Informationssystem der Universität als Instrument zur zielgerichteten Lenkung von Universitätsprozessen

Nr. 25 (1994) Wittke M., Mekinic, G.: Kooperierende Informationsräume. Ein Ansatz für verteilte Führungsinformationssysteme

Nr. 26 (1995) Ferstl O.K., Sinz E.J.: Re-Engineering von Geschäftsprozessen auf der Grundlage des SOM-Ansatzes

Nr. 27 (1995) Ferstl, O.K., Mannmeusel, Th.: Dezentrale Produktionslenkung. Erscheint in CIM-Management 3/1995

Nr. 28 (1995) Ludwig, H., Schwab, K.: Integrating cooperation systems: an event-based approach

Nr. 30 (1995) Augsburger W., Ludwig H., Schwab K.: Koordinationsmethoden und -werkzeuge bei der computergestützten kooperativen Arbeit

Nr. 31 (1995) Ferstl O.K., Mannmeusel T.: Gestaltung industrieller Geschäftsprozesse

Nr. 32 (1995) Gunzenhäuser R., Duske A., Ferstl O.K., Ludwig H., Mekinic G., Rieder H., Schwab H.-J., Schwab K., Sinz E.J., Wittke M: Festschrift zum 60. Geburtstag von Walter Augsburger

Nr. 33 (1995) Sinz, E.J.: Kann das Geschäftsprozeßmodell der Unternehmung das unternehmensweite Datenschema ablösen?

Nr. 34 (1995) Sinz E.J.: Ansätze zur fachlichen Modellierung betrieblicher Informationssysteme - Entwicklung, aktueller Stand und Trends -

Nr. 35 (1995) Sinz E.J.: Serviceorientierung der Hochschulverwaltung und ihre Unterstützung durch workflow-orientierte Anwendungssysteme

Nr. 36 (1996) Ferstl O.K., Sinz, E.J., Amberg M.: Stichwörter zum Fachgebiet Wirtschaftsinformatik. Erscheint in: Broy M., Spaniol O. (Hrsg.): Lexikon Informatik und Kommunikationstechnik, 2. Auflage, VDI-Verlag, Düsseldorf 1996

Nr. 37 (1996) Ferstl O.K., Sinz E.J.: Flexible Organizations Through Object-oriented and Transaction-oriented Information Systems, July 1996

Nr. 38 (1996) Ferstl O.K., Schäfer R.: Eine Lernumgebung für die betriebliche Aus- und Weiterbildung on demand, Juli 1996

Nr. 39 (1996) Hazebrouck J.-P.: Einsatzpotentiale von Fuzzy-Logic im Strategischen Management dargestellt an Fuzzy-System-Konzepten für Portfolio-Ansätze

Nr. 40 (1997) Sinz E.J.: Architektur betrieblicher Informationssysteme. In: Rechenberg P., Pomberger G. (Hrsg.): Handbuch der Informatik, Hanser-Verlag, München 1997

Nr. 41 (1997) Sinz E.J.: Analyse und Gestaltung universitärer Geschäftsprozesse und Anwendungssysteme. Angenommen für: Informatik '97. Informatik als Innovationsmotor. 27. Jahrestagung der Gesellschaft für Informatik, Aachen 24.-26.9.1997

Nr. 42 (1997) Ferstl O.K., Sinz E.J., Hammel C., Schlitt M., Wolf S.: Application Objects – fachliche Bausteine für die Entwicklung komponentenbasierter Anwendungssysteme. Angenommen für: HMD – Theorie und Praxis der Wirtschaftsinformatik. Schwerpunkheft ComponentWare, 1997

Nr. 43 (1997): Ferstl O.K., Sinz E.J.: Modeling of Business Systems Using the Semantic Object Model (SOM) – A Methodological Framework - . Accepted for: P. Bernus, K. Mertins, and G. Schmidt (ed.): Handbook on Architectures of Information Systems. International Handbook on Information Systems, edited by Bernus P., Blazewicz J., Schmidt G., and Shaw M., Volume I, Springer 1997

Ferstl O.K., Sinz E.J.: Modeling of Business Systems Using (SOM), 2nd Edition. Appears in: P. Bernus, K. Mertins, and G. Schmidt (ed.): Handbook on Architectures of Information Systems. International Handbook on Information Systems, edited by Bernus P., Blazewicz J., Schmidt G., and Shaw M., Volume I, Springer 1998

Nr. 44 (1997) Ferstl O.K., Schmitz K.: Zur Nutzung von Hypertextkonzepten in Lernumgebungen. In: Conradi H., Kreutz R., Spitzer K. (Hrsg.): CBT in der Medizin – Methoden, Techniken, Anwendungen -. Proceedings zum Workshop in Aachen 6. – 7. Juni 1997. 1. Auflage Aachen: Verlag der Augustinus Buchhandlung

Nr. 45 (1998) Ferstl O.K.: Datenkommunikation. In. Schulte Ch. (Hrsg.): Lexikon der Logistik, Oldenbourg-Verlag, München 1998

Nr. 46 (1998) Sinz E.J.: Prozeßgestaltung und Prozeßunterstützung im Prüfungswesen. Erschienen in: Proceedings Workshop „Informationssysteme für das Hochschulmanagement". Aachen, September 1997

Nr. 47 (1998) Sinz, E.J.:, Wismans B.: Das „Elektronische Prüfungsamt". Erscheint in: Wirtschaftswissenschaftliches Studium WiSt, 1998

Nr. 48 (1998) Haase, O., Henrich, A.: A Hybrid Respresentation of Vague Collections for Distributed Object Management Systems. Erscheint in: IEEE Transactions on Knowledge and Data Engineering

Nr. 49 (1998) Henrich, A.: Applying Document Retrieval Techniques in Software Engineering Environments. In: Proc. International Conference on Database and Expert Systems

Applications. (DEXA 98), Vienna, Austria, Aug. 98, pp. 240-249, Springer, Lecture Notes in Computer Sciences, No. 1460

Nr. 50 (1999) Henrich, A., Jamin, S.: On the Optimization of Queries containing Regular Path Expressions. Erscheint in: Proceedings of the Fourth Workshop on Next Generation Information Technologies and Systems (NGITS'99), Zikhron-Yaakov, Israel, July, 1999 (Springer, Lecture Notes)

Nr. 51 (1999) Haase O., Henrich, A.: A Closed Approach to Vague Collections in Partly Inaccessible Distributed Databases. Erscheint in: Proceedings of the Third East-European Conference on Advances in Databases and Information Systems – ADBIS'99, Maribor, Slovenia, September 1999 (Springer, Lecture Notes in Computer Science)

Nr. 52 (1999) Sinz E.J., Böhnlein M., Ulbrich-vom Ende A.: Konzeption eines Data Warehouse-Systems für Hochschulen. Angenommen für: Workshop „Unternehmen Hochschule" im Rahmen der 29. Jahrestagung der Gesellschaft für Informatik, Paderborn, 6. Oktober 1999

Nr. 53 (1999) Sinz E.J.: Konstruktion von Informationssystemen. Der Beitrag wurde in geringfügig modifizierter Fassung angenommen für: Rechenberg P., Pomberger G. (Hrsg.): Informatik-Handbuch. 2., aktualisierte und erweiterte Auflage, Hanser, München 1999

Nr. 54 (1999) Herda N., Janson A., Reif M., Schindler T., Augsburger W.: Entwicklung des Intranets SPICE: Erfahrungsbericht einer Praxiskooperation.

Nr. 55 (2000) Böhnlein M., Ulbrich-vom Ende A.: Grundlagen des Data Warehousing. Modellierung und Architektur

Nr. 56 (2000) Freitag B, Sinz E.J., Wismans B.: Die informationstechnische Infrastruktur der Virtuellen Hochschule Bayern (vhb). Angenommen für Workshop "Unternehmen Hochschule 2000" im Rahmen der Jahrestagung der Gesellschaft f. Informatik, Berlin 19. - 22. September 2000

Nr. 57 (2000) Böhnlein M., Ulbrich-vom Ende A.: Developing Data Warehouse Structures from Business Process Models.

Nr. 58 (2000) Knobloch B.: Der Data-Mining-Ansatz zur Analyse betriebswirtschaftlicher Daten.

Nr. 59 (2001) Sinz E.J., Böhnlein M., Plaha M., Ulbrich-vom Ende A.: Architekturkonzept eines verteilten Data-Warehouse-Systems für das Hochschulwesen. Angenommen für: WI-IF 2001, Augsburg, 19.-21. September 2001

Nr. 60 (2001) Sinz E.J., Wismans B.: Anforderungen an die IV-Infrastruktur von Hochschulen. Angenommen für: Workshop „Unternehmen Hochschule 2001" im Rahmen der Jahrestagung der Gesellschaft für Informatik, Wien 25. – 28. September 2001

Änderung des Titels der Schriftenreihe *Bamberger Beiträge zur Wirtschaftsinformatik* in *Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik* ab Nr. 61

Note: The title of our technical report series has been changed from *Bamberger Beiträge zur Wirtschaftsinformatik* to *Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik* starting with TR No. 61

Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik

Nr. 61 (2002) Goré R., Mendler M., de Paiva V. (Hrsg.): Proceedings of the International Workshop on Intuitionistic Modal Logic and Applications (IMLA 2002), Copenhagen, July 2002.

Nr. 62 (2002) Sinz E.J., Plaha M., Ulbrich-vom Ende A.: Datenschutz und Datensicherheit in einem landesweiten Data-Warehouse-System für das Hochschulwesen. Erscheint in: Beiträge zur Hochschulforschung, Heft 4-2002, Bayerisches Staatsinstitut für Hochschulforschung und Hochschulplanung, München 2002

Nr. 63 (2005) Aguado, J., Mendler, M.: Constructive Semantics for Instantaneous Reactions

Nr. 64 (2005) Ferstl, O.K.: Lebenslanges Lernen und virtuelle Lehre: globale und lokale Verbesserungspotenziale. Erschienen in: Kerres, Michael; Keil-Slawik, Reinhard (Hrsg.); Hochschulen im digitalen Zeitalter: Innovationspotenziale und Strukturwandel, S. 247 – 263; Reihe education quality forum, herausgegeben durch das Centrum für eCompetence in Hochschulen NRW, Band 2, Münster/New York/München/Berlin: Waxmann 2005

Nr. 65 (2006) Schönberger, Andreas: Modelling and Validating Business Collaborations: A Case Study on RosettaNet

Nr. 66 (2006) Markus Dorsch, Martin Grote, Knut Hildebrandt, Maximilian Röglinger, Matthias Sehr, Christian Wilms, Karsten Loesing, and Guido Wirtz: Concealing Presence Information in Instant Messaging Systems, April 2006

Nr. 67 (2006) Marco Fischer, Andreas Grünert, Sebastian Hudert, Stefan König, Kira Lenskaya, Gregor Scheithauer, Sven Kaffille, and Guido Wirtz: Decentralized Reputation Management for Cooperating Software Agents in Open Multi-Agent Systems, April 2006

Nr. 68 (2006) Michael Mendler, Thomas R. Shiple, Gérard Berry: Constructive Circuits and the Exactness of Ternary Simulation

Nr. 69 (2007) Sebastian Hudert: A Proposal for a Web Services Agreement Negotiation Protocol Framework . February 2007

Nr. 70 (2007) Thomas Meins: Integration eines allgemeinen Service-Centers für PC-und Medientechnik an der Universität Bamberg – Analyse und Realisierungs-Szenarien. February 2007 (out of print)

Nr. 71 (2007) Andreas Grünert: Life-cycle assistance capabilities of cooperating Software Agents for Virtual Enterprises. März 2007

Nr. 72 (2007) Michael Mendler, Gerald Lüttgen: Is Observational Congruence on μ-Expressions Axiomatisable in Equational Horn Logic?

Nr. 73 (2007) Martin Schissler: out of print

Nr. 74 (2007) Sven Kaffille, Karsten Loesing: Open chord version 1.0.4 User's Manual. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 74, Bamberg University, October 2007. ISSN 0937-3349.

Nr. 75 (2008) Karsten Loesing (Hrsg.): Extended Abstracts of the Second *Privacy Enhancing Technologies Convention* (PET-CON 2008.1). Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 75, Bamberg University, April 2008. ISSN 0937-3349.

Nr. 76 (2008) Gregor Scheithauer, Guido Wirtz: Applying Business Process Management Systems – A Case Study. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 76, Bamberg University, May 2008. ISSN 0937-3349.

Nr. 77 (2008) Michael Mendler, Stephan Scheele: Towards Constructive Description Logics for Abstraction and Refinement. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 77, Bamberg University, September 2008. ISSN 0937-3349.

Nr. 78 (2008) Gregor Scheithauer, Matthias Winkler: A Service Description Framework for Service Ecosystems. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 78, Bamberg University, October 2008. ISSN 0937-3349.

Nr. 79 (2008) Christian Wilms: Improving the Tor Hidden Service Protocol Aiming at Better Performances. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 79, Bamberg University, November 2008. ISSN 0937-3349.

Nr. 80 (2009) Thomas Benker, Stefan Fritzemeier, Matthias Geiger, Simon Harrer, Tristan Kessner, Johannes Schwalb, Andreas Schönberger, Guido Wirtz: QoS Enabled B2B Integration. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 80, Bamberg University, May 2009. ISSN 0937-3349.

Nr. 81 (2009) Ute Schmid, Emanuel Kitzelmann, Rinus Plasmeijer (Eds.): Proceedings of the ACM SIGPLAN Workshop on *Approaches and Applications of Inductive Programming* (AAIP'09), affiliated with ICFP 2009, Edinburgh, Scotland, September 2009. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 81, Bamberg University, September 2009. ISSN 0937-3349.

Nr. 82 (2009) Ute Schmid, Marco Ragni, Markus Knauff (Eds.): Proceedings of the KI 2009 Workshop *Complex Cognition*, Paderborn, Germany, September 15, 2009. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 82, Bamberg University, October 2009. ISSN 0937-3349.

Nr. 83 (2009) Andreas Schönberger, Christian Wilms and Guido Wirtz: A Requirements Analysis of Business-to-Business Integration. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 83, Bamberg University, December 2009. ISSN 0937-3349.

Nr. 84 (2010) Werner Zirkel, Guido Wirtz: A Process for Identifying Predictive Correlation Patterns in Service Management Systems. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 84, Bamberg University, February 2010. ISSN 0937-3349.

Nr. 85 (2010) Jan Tobias Mühlberg und Gerald Lüttgen: Symbolic Object Code Analysis. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 85, Bamberg University, February 2010. ISSN 0937-3349.

Nr. 86 (2010) Werner Zirkel, Guido Wirtz: Proaktives Problem Management durch Eventkorrelation – ein *Best Practice* Ansatz. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 86, Bamberg University, August 2010. ISSN 0937-3349.

Nr. 87 (2010) Johannes Schwalb, Andreas Schönberger: Analyzing the Interoperability of WS-Security and WS-ReliableMessaging Implementations. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 87, Bamberg University, September 2010. ISSN 0937-3349.

Nr. 88 (2011) Jörg Lenhard: A Pattern-based Analysis of WS-BPEL and Windows Workflow. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 88, Bamberg University, March 2011. ISSN 0937-3349.

Nr. 89 (2011) Andreas Henrich, Christoph Schlieder, Ute Schmid [eds.]: Visibility in Information Spaces and in Geographic Environments – Post-Proceedings of the KI'11 Workshop. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 89, Bamberg University, December 2011. ISSN 0937-3349.

Nr. 90 (2012) Simon Harrer, Jörg Lenhard: Betsy - A BPEL Engine Test System. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 90, Bamberg University, July 2012. ISSN 0937-3349.

Nr. 91 (2013) Michael Mendler, Stephan Scheele: On the Computational Interpretation of CKn for Contextual Information Processing - Ancillary Material. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 91, Bamberg University, May 2013. ISSN 0937-3349.

Nr. 92 (2013) Matthias Geiger: BPMN 2.0 Process Model Serialization Constraints. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 92, Bamberg University, May 2013. ISSN 0937-3349.