

# Pattern Catalog Description

This document is generated from the pattern catalog DTD (found at <http://www.wagz.informatik.uni-kl.de/projects/psigene/dtd>).

## Table of Contents

Pattern Catalog Description	1
Table of Contents	1
1. Entities	3
1.1 AbstractionType	3
1.2 Access	3
1.3 ActionType	3
1.4 Aggregation	3
1.5 ArtefactType	4
1.6 BindingType	4
1.7 Boolean	5
1.8 CDRelType	5
1.9 CellStyle	5
1.10 ChoiceType	5
1.11 CreateType	6
1.12 LanguageType	6
1.13 Location	6
1.14 Mutability	6
1.15 RelType	6
2. Elements	7
2.1 CDAttribute	7
2.2 CDClassDiagram	7
2.3 CDMethod	7
2.4 CDOBJECTType	8
2.5 CDParameter	8
2.6 CDParameters	8
2.7 CDRelation	8
2.8 SVG	9
2.9 Action	9
2.10 Activity	9

2.11 Aka	9
2.12 Arc	10
2.13 Artefact	10
2.14 Attribute	10
2.15 Author	10
2.16 Catalog	11
2.17 CatalogRef	12
2.18 Cell	13
2.19 Choice	13
2.20 Choices	13
2.21 Consequence	14
2.22 Constraint	14
2.23 Context	14
2.24 Copyright	14
2.25 Description	14
2.26 Force	15
2.27 Formula	15
2.28 Generator	16
2.29 Implementation	16
2.30 Keyword	18
2.31 License	18
2.32 Parameter	18
2.33 Pattern	18
2.34 PatternRef	20
2.35 Problem	20
2.36 Reference	20
2.37 Relationship	21
2.38 ReleaseNotes	21
2.39 Solution	21
2.40 Structure	22
2.41 Summary	22
2.42 Table	22
2.43 Text	22
2.44 Url	23
2.45 Version	23

# 1. Entities

## 1.1 AbstractionType

Different abstraction levels for a Pattern.

<i>Value</i>	<i>Description</i>
Analysis	Analysis of the requirements, create acceptance tests.
Architecture	An architectural pattern describes an abstract view on the architecture of a given problem.
Cots	Describes a "Component Of The Shelf" (COTS).
Definition	Define common vocabulary.
Design	"Classical" design patterns.
Idiom	Idioms (programming patterns).
Maintainance	Maintain a running system.
Requirements	Create and ask for system requirements.
Test	Patterns to create test cases.
Testcase	Reusable test cases.

## 1.2 Access

<i>Value</i>	<i>Description</i>
Private	
Protected	
Public	

## 1.3 ActionType

An ActionType is the type of an action.

<i>Value</i>	<i>Description</i>
Aggregate	
Bind	
Select	

## 1.4 Aggregation

The Aggregation entity is the string representation of an element's level of aggregation over another in a relationship.

<i>Value</i>	<i>Description</i>
Aggregation	Other objects are aggregated by this object.
Composition	The object is composed of other objects (strong aggregation).
None	Normal "uses" association.

## 1.5 ArtefactType

An ArtefactType is the type of an artefact.

<i>Value</i>	<i>Description</i>
Artefact	A generic Artefact
Attribute	
COTS	
CodeBlock	
Configuration	A generic Configuration
DataType	
DesignModelPart	
Document	
Instantiation	
Method	
ObjectType	
Package	
ProblemDescription	
Relation	
RelationEndPoint	
Requirement	
Strategy	
TestCase	
TestStrategy	

## 1.6 BindingType

A BindingType defines the possible Elements of a Design Model for a Parameter binding.

<i>Value</i>	<i>Description</i>
Attribute	An Attribute
Boolean	A boolean value (True/False, Yes/No)
Choice	A choice of values (to be defined as sub-elements)
Method	A Method
ObjectType	An ObjectType
Relation	A Relation
String	A string

## 1.7 Boolean

<i>Value</i>	<i>Description</i>
False	
True	

## 1.8 CDRelType

<i>Value</i>	<i>Description</i>
Aggregate	
Implement	
Inherit	
Strong_aggregate	
Use	

## 1.9 CellStyle

Defines the type of the cell. Cells may be "normal" or "heading"-cells.

<i>Value</i>	<i>Description</i>
Heading	Heading cells are printed in bold. The first heading cells are considered to be the heading of a table
Normal	A normal cell.

## 1.10 ChoiceType

A ChoiceType defines how a choice should be presented to a user.

<i>Value</i>	<i>Description</i>
Combo	Display choice as a combo box
Radio	Display choice as radio buttons

## 1.11 CreateType

A CreateType defines if the target must exist, is implemented or ist implemented as needed

<i>Value</i>	<i>Description</i>
AsNeeded	Implement a new model element if needed
Implement	Implement a new model element
Use	Use an existing model element

## 1.12 LanguageType

The Language defines the language in which the pattern is formulated.

<i>Value</i>	<i>Description</i>
Deutsch	german language
English	english language

## 1.13 Location

A location entity is optionally one of the following: 1) a relative path, delimited by "/" characters, that defines the location of a resource relative to the location of the XML file it is referenced within 2) or a URI path to an external resource.

<i>Value</i>	<i>Description</i>
CDATA	The URI data.

## 1.14 Mutability

The Mutability entity is the string representation of an element's changeability.

<i>Value</i>	<i>Description</i>
Read	the element may be read.
ReadWrite	the element may be read or written.

## 1.15 RelType

The RelType entity represents a type of relationship between two patterns.

<i>Value</i>	<i>Description</i>
Chain	The referenced pattern can be chained to this pattern
Exclude	This pattern excludes the use of the other pattern
Like	One pattern is like another.
Nested	The referenced pattern is nested in this pattern (ie. it is a subpattern).
Reference	Reference to another pattern.

## 2. Elements

### 2.1 CDAttribute

An Attribute of an ObjectType.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>		
Description	0..1			

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Default	CDATA	No	-	An optional default value
Name	CDATA	Yes	-	The attributes name
Static	Boolean	Yes	false	True if this attribute is static
Type	CDATA	Yes	-	The type of the Attribute (as String)
Visibility	Access	Yes	public	The access to this attribute

### 2.2 CDClassDiagram

Describes a Class diagram.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>		
Description	0..1			
CDOBJECTTYPE	0..*	A List of all Objecttypes (classes) in this diagram		
CDRelation	0..*	A List of all Associations in this diagram		

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Title	CDATA	Yes	-	The title of this class diagram

### 2.3 CDMethod

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>		
Description	0..1			
CDParameter	0..*			

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Name	CDATA	Yes	-	
ReturnType	CDATA	No	-	
Static	Boolean	Yes	false	
Visibility	Access	Yes	public	

## 2.4 CDOBJECTType

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Description	0..1	
CDAttribute	0..*	
CDMethod	0..*	

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Abstract	Boolean	Yes	false	
Interface	Boolean	Yes	false	
Name	CDATA	Yes	-	
Visibility	Access	Yes	public	

## 2.5 CDParameter

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Description	0..1	

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Name	CDATA	Yes	-	
Type	CDATA	Yes	-	

## 2.6 CDParameters

An Attribute of an ObjectType.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
CDParameter	0..*	

## 2.7 CDRelation

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Description	0..1	

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Directed	Boolean	Yes	false	
Name	CDATA	Yes	-	
Origin	CDATA	Yes	-	
OriginCard	CDATA	No	-	
Target	CDATA	Yes	-	
TargetCard	CDATA	No	-	
Type	CDRelType	Yes	use	

## 2.8 SVG

Contains a SVG drawing

## 2.9 Action

Actions describe, what should be done in the actual activity. This description is made in a special action-language (to be done!). Actions can be: select, generate, ...

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Type	ActionType	Yes	-	
What	CDATA	Yes	-	

## 2.10 Activity

Activities are the active part of an Activity-Diagram. Each Activity contains an action (something that should be done in order to apply this activity) and may contain a generator (an automatic way to implement the activity).

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Description	1	Describes this activity.
Action	0..*	An Action that should be done to apply this activity.
Generator	0..1	Shows how this Activity should be implemented.

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Name	CDATA	Yes	-	The name of the activity. Each activity name must be unique for a strategy.

## 2.11 Aka

An aka element is an another name that the parent element may be known by (Also known As).

## 2.12 Arc

Arcs connect Activities in an Activity-Diagram. Each Arc can connect several sources with several targets and may contain a constraint. Empty sources are initial arcs end empty targets are ending arcs.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Constraint	0..*	A constraint that must be fulfilled in order to be able to traverse this arc.

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Sources	CDATA	No	-	A list of sources for this arc. This should be a comma separated list of activity names. (Better use ID/IDREFS?) Empty sources describe an arc to an initial activity.
Targets	CDATA	No	-	A list of targets for this arc (see sources). An empty target list describes a possible End-Activity.

## 2.13 Artefact

An artefact is a part of a design model. Artefacts can be generated by generators via the implement command.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Attribute	0..*	

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Name	CDATA	Yes	-	
Type	ArtefactType	Yes	-	

## 2.14 Attribute

An attribute is a name-value pair describing an attribute of an artefact.

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Name	CDATA	Yes	-	
Value	CDATA	Yes	-	

## 2.15 Author

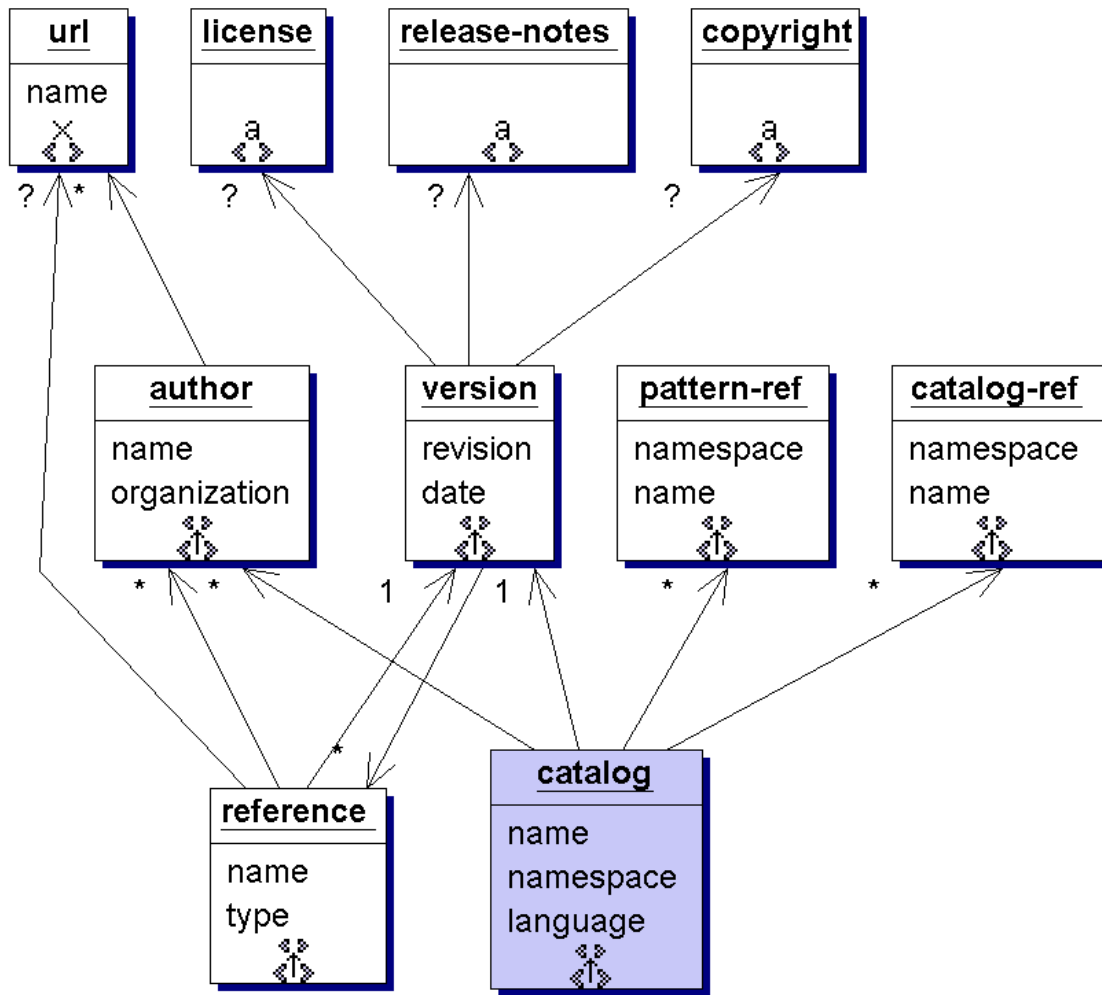
An author element contains information identifying the creator of a resource.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Description	1	Description of the author
Url	0..*	A URL where information pertaining to the author, his organization, or his works may be obtained. This includes e-mail addresses.

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Name	CDATA	Yes	-	Name of the author
Organization	CDATA	No	-	Organization the author represents. If the name is an actual organization, then this attribute may be omitted.

## 2.16 Catalog

A catalog groups a number of related patterns and strategies according to some criteria. There is no restriction on how they are grouped, so it could be by domain, company, abstraction level, etc. A catalog serves as the basis for packaging and exchanging patterns and strategies.



<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Version	1	The catalogs version.
Author	0..*	The author of this catalog (ie. the person that assembled all the patterns for the catalog).
Description	0..1	Description of the catalog
CatalogRef	0..*	Nested catalogs
PatternRef	0..*	Patterns that are included within the catalog

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Language	LanguageType	No	-	The language of this strategy (english or deutsch)
Name	CDATA	Yes	-	Name of the catalog
Namespace	CDATA	Yes	-	A space within which the catalog name must be unique

## 2.17 CatalogRef

A reference to other pattern catalogs.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Description	0..1	Comments about this reference.

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Name	CDATA	Yes	-	Name of the referenced catalog
Namespace	CDATA	Yes	-	Namespace of the referenced catalog.

## 2.18 Cell

A cell displays data in a table.

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Colspan	CDATA	Yes	1	A number of how many columns this cell should span.
Style	CellStyle	Yes	normal	Contains the style of the table cell. Heading cells will be displayed in bold. If the table starts with heading cells, these will be considered as table heading.

## 2.19 Choice

Choices are choosable elements of a choices element

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Description	0..1	The (optional) description of this choice.

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Name	CDATA	Yes	-	The name of the choice.

## 2.20 Choices

A Choices element describes a choice that has to be made in order to apply this strategy. Choices can be used to control the activity diagram of the strategy.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Description	1	A Further description of this choice
Choice	1..*	The values that can be chosen

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Type	ChoiceType	Yes	combo	How should be chosen. Currently supported are combo-box and radio-buttons

## 2.21 Consequence

A consequence represents a pro or con of pattern usage. It describes how a pattern supports its objectives and the trade-offs in doing so.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Description	1	A description of the consequence

## 2.22 Constraint

Constraints are a way to restrict certain activities. They can be informal text or can be formally verified using an automated tool.

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Name	CDATA	No	-	An optional name of the constraint to create meaningful error messages.
Type	(python_condition python_function informal)	Yes	-	Shows the type of this constraint. Currently only informal, python_condition and python_function are supported.

## 2.23 Context

A context represents the environment within which a pattern describes itself and is a general motivation for its existence.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Description	1	A description of the context

## 2.24 Copyright

The copyright element provides a copyright notice.

## 2.25 Description



Contains a textual representation of a formula

## 2.28 Generator

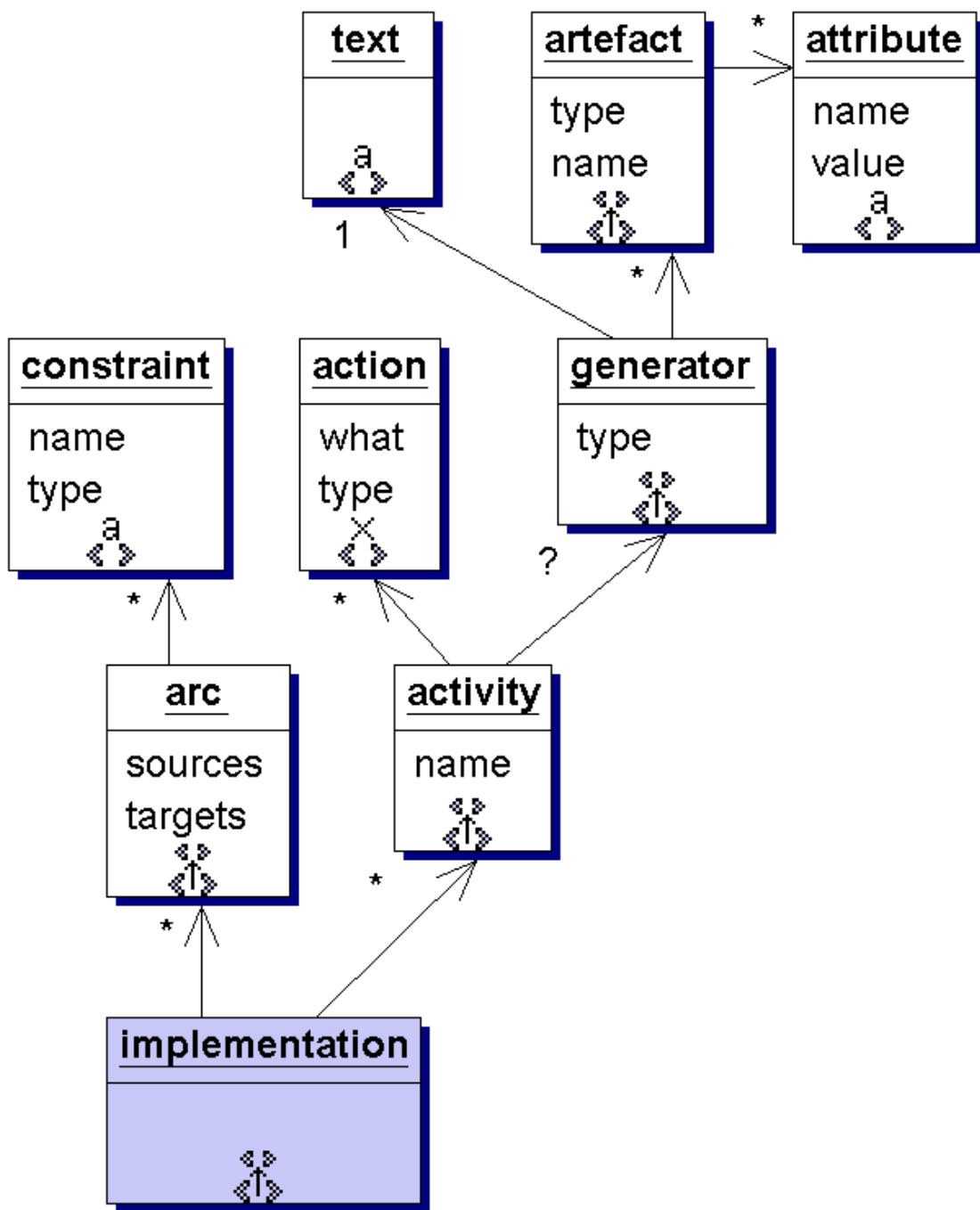
A generator automatically implements certain aspects of this strategy. Currently there are two types of generators: text and python. A python-generator can automatically implement source code or model code. text-generators show a descriptive text of what should be implemented manually.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Artefact	0..*	A number of artefacts that can be generated.
Text	1	

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Type	(implicit python text)	Yes	-	Holds the generator type. May be text or python.

## 2.29 Implementation

The Implementation describes how a strategy is can be applied. It describes the generated SW-model and gives an Activity- Diagram that guides the implementation.



<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Description	0..1	A description of the Model-Components that get implemented by this strategy. Describes the solution of this strategy
Activity	0..*	All activities that can be used for the implementation.
Arc	0..*	Arcs connect the different activities forming an Activity-Diagram.

## 2.30 Keyword

A keyword element is word or phrase that is useful in categorizing the parent element and its characteristics.

## 2.31 License

The license element provides licensing information for a resource that, among other things, defines usage restrictions.

## 2.32 Parameter

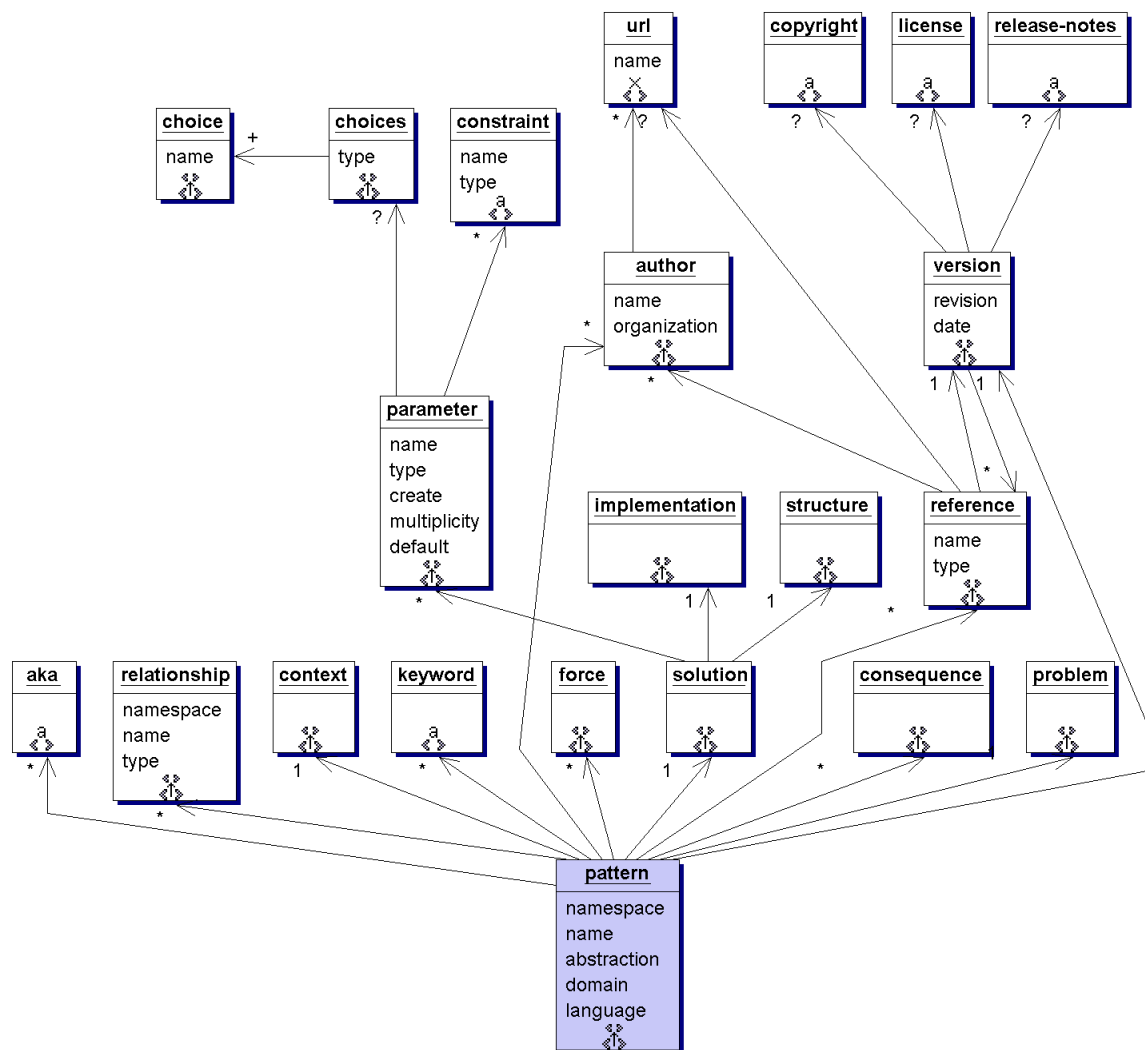
A parameter represents a distinct role played by a component in the pattern solution. Each parameter describes its general characteristics but does not place any constraints on how it may be realized.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Choices	0..1	
Description	1	Description of the parameter and its role in the solution
Constraint	0..*	

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Create	CreateType	Yes	use	Must the target exist, ist it implemented or is it implemented if needed.
Default	CDATA	No	-	
Multiplicity	CDATA	Yes	1	Allows the role to be filled by more than one component. This is useful in patterns like Abstract Factory, where the concrete factory role will be mapped to multiple components. The following values are available: 1 - One 1..* - Any whole number > 1 * - Many or more than one
Name	CDATA	Yes	-	Name of the parameter, which must be unique among the others.
Type	BindingType	Yes	-	The parameter type of this parameter.

## 2.33 Pattern

A pattern is a somewhat generic description of a solution provided to address one or a common set of problems in a certain context. Although a pattern describes a solution, it does not put any constraints on how that solution may be realized. A pattern may; however, describe how it relates to other patterns and even how it may be composed of other patterns. In this way, the abstract nature of patterns is preserved while the realization of solutions and idioms is reserved for strategies.



<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Aka	0..*	Other names for the pattern (also known as)
Keyword	0..*	Categorizations or classifications of the pattern
Version	1	Version information for the pattern
Author	0..*	Authors of the pattern
Context	1	Environment of the pattern
Force	0..*	Motivation of the pattern
Problem	1	The problem solved by the pattern
Solution	1	The solution to the problem provided by the pattern
Consequence	0..*	Consequence of the pattern's use
Relationship	0..*	Other related patterns
Reference	0..*	External resources that further describes the pattern

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Abstraction	AbstractionType	No	-	Abstraction level of the pattern, which may include such descriptions as "Architectural" or "Design"
Domain	CDATA	No	-	Domain the pattern is particularly well suited for or intended for, which may include such descriptions as "Financial," "Telecommunication," "Medical," etc.
Language	LanguageType	No	-	The language in that the pattern is written
Name	CDATA	Yes	-	Name of the pattern
Namespace	CDATA	Yes	-	A space within which the pattern name must be unique

## 2.34 PatternRef

A pattern-ref represents the pattern for which this strategy provides a solution implementation. A strategy may realize only one pattern.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Description	0..1	Additional description of this reference.

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Name	CDATA	Yes	-	Name of the pattern.
Namespace	CDATA	Yes	-	Namespace of the pattern.

## 2.35 Problem

A problem represents a design need that is to be addressed by a pattern. It essentially distinguishes the use of one pattern over another.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Description	1	More detailed explanation of the problem

## 2.36 Reference

A reference element is an external file that may not be appropriately supplied in XML form. When related to patterns, it helps to further describe a pattern or instruct in its use. Examples of an artifact include: UML diagram, graphical image, binary documentation, etc.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Description	0..1	Description of the artifact
Author	0..*	An author of the artifact
Url	0..1	Location of the artifact in the form of a URL, which may be either relative or absolute
Version	1	Version information for the artifact

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Name	CDATA	Yes	-	Name of the artifact
Type	CDATA	Yes	-	File type of the artifact, which should be represented by a common file extension (i.e. html, doc, mdl) or as internal link with catalog-ref, pattern-ref, or strategy-ref.

## 2.37 Relationship

A relationship represents a relationship between two patterns. A pattern relationship is purely descriptive, but it does have an attribute that specifies what type of relationship it is. This element would be used to refer to a like pattern or to describe a pattern nesting.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Description	1	Description of how the two patterns are related

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Name	CDATA	Yes	-	Name of the related pattern
Namespace	CDATA	Yes	-	Namespace of the related pattern
Type	RelType	Yes	reference	Defines the type of relationship

## 2.38 ReleaseNotes

The release-notes element provides a description of a resource revision.

## 2.39 Solution

A solution solves the problem described in a pattern. It is composed of a number of parameters and defines the static structure and dynamic interactions of them

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Description	1	More detailed explanation of the solution
Parameter	0..*	A formal binding parameter of the pattern
Structure	1	Static structure of the solution
Implementation	1	

## 2.40 Structure

A structure represents the static interaction of participants (as in a UML class diagram) in a solution.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Description	1	Description of the collaboration, including how the participants interact

## 2.41 Summary

A summary element is a quick summary of another element.

## 2.42 Table

A table displays cells in a tabular way. Cells may span several columns. The number of columns of the table must be known in advance.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Cell	1..*	All cells that make up this table.

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Caption	CDATA	No	-	An optional caption for this table. If it is given, the table gets annotated with "Table : ".
Cols	CDATA	Yes	-	The number of columns in this table.
Name	CDATA	No	-	The name of the Table. The name can be used to reference the table (eg. for table lookups).

## 2.43 Text

Contains a paragraph of text

## 2.44 Url

An url element is an URL link. An example would be an address of "www.johndoe.com."

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Name	CDATA	Yes	-	the adress of the url

## 2.45 Version

A version element represents versioning information of a resource. It's primary purpose is to distinguish multiple revisions of the same resource.

<i>Element</i>	<i>Cardinality</i>	<i>Description</i>
Description	0..1	Description of the revision
Copyright	0..1	Copyright notice for this revision of the resource
ReleaseNotes	0..1	Notes that describe important aspects of this revision
License	0..1	Licensing information for this revision of the resource
Reference	0..*	

<i>Value</i>	<i>Type</i>	<i>Required</i>	<i>Default</i>	<i>Description</i>
Date	CDATA	No	-	Date/time of the revision
Revision	CDATA	Yes	-	Version number