



SBR

Standard Validation Rules

Specification

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public



Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

Abstract

This validation specification is the specification for the standard-validation-rule linkbase (svr linkbase) and its semantics as used by SBR Programmes. This Standard Validation Rules specification was created to support the validation of business logic relationships (business or validation rules) between XBRL facts in an XBRL instance document. This validation specification describes the svr linkbase that is used to define the validation rules and also the processing model for performing the validation for all rules on a given XBRL instance document.

Given the very specific need of the validation of business logic relationships in an XBRL instance document the scope of the specification is less broad than the Formula linkbase specification. We have however opted for a fast specification release process that would serve the immediate needs of SBR rather than wait for the release of a more comprehensive specification that would take too long from the point of view of the SBR and also is very complex and therefore will cost a lot of effort for software vendors to incorporate in their software. At the same time however we hope that the work presented in this specification can and will form a foundation for an XRI comprehensive validation specification.

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

Table of contents

Abstract	1
1 Introduction	4
1.1 Scope	4
2 Requirements	5
3 Assumptions.....	6
3.1 Monetary unit.....	6
3.2 Nillable.....	6
3.3 Association.....	6
3.4 Floating point.....	6
3.5 Preconditions	6
3.6 Context grouping	6
3.7 Expression language	6
3.8 Tuples	6
4 Syntax	7
4.1 XPath usage	7
4.2 Rule resources	7
4.3 Rule relationships	8
5 Context grouping patterns.....	9
5.1 Movement-complete.....	9
5.2 Movement-all	10
5.3 Single instant	11
5.4 Single duration.....	11
5.5 Single context	12
5.6 Duration-Instant	12
5.7 Instant-Duration	13
6 Validation functions.....	14
6.1 getItemFactForContext	14
6.2 getItemFactForContextOrZero	14
6.3 getItemFactsOrZero	15
6.4 getItemFactsForContextOrZero.....	15
6.5 getItemFactsForContextsOrZero	16
6.6 minimum	16
6.7 maximum	16
6.8 date.....	17
6.9 getUnit	17
6.10 testPattern	17
7 Processing model	18
7.1 Validation rule networks	18
7.2 Context grouping	18



Version: 1.0
Date: 22 june 2009
Status: Recommendation
Classification: Public

7.3	Context nodes	18
7.4	Rule evaluation	18
7.5	Producing validation result	19
8	<i>Schema</i>.....	21
8.1	standard-validation-rules-2008-08.xsd (normative).....	21

© State of the Netherlands and the Australian Government.

It is allowed to reproduce, distribute and produce derivative works under the following terms. The name of the SBR Programmes as the original author will be mentioned on every occasion. It shall not be suggested that the SBR Programmes support a derivative work, without an expressed prior approval in writing.

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

1 Introduction

A validation rule is a way of describing business logic relationships between XBRL facts in XBRL instance documents. When evaluated against an XBRL instance document, failing validation rules will produce messages to signal such failures. For example, a validation rule may express a check for consistency in the AssetsNet facts reported across multiple periods:

```
Current AssetsNet= Previous AssetsNet + ChangesInAssetsNet
```

The rule describes the use of the XBRL concepts for `AssetsNet` and `ChangesInAssetsNet` and that current Net Assets should be equal to previous Net Assets plus changes during the current period.

The general processing model for a validation rule is to apply the rule against an XBRL instance document. Validation rules are to be used to validate facts in the XBRL instance document.

1.1 Scope

The scope for this version of the Standard Validation Rules specification is limited to the used functionality as defined in the architecture for the SBR taxonomies. The validation rules that can be applied on dimensional taxonomies will be supported by the next version of the specifications. The set of expressions is currently limited to the expressions supported by XPath 1.0, the use of a broader set of expressions by using XPath 2.0 will be available in the next version of the specifications.

Version: 1.0
Date: 22 june 2009
Status: Recommendation
Classification: Public

2 Requirements

- This specification should define the standard-validation-rule linkbase and a processing model and output content model for performing the validation rules.
- Only validation rules to be evaluated against a single XBRL instance document should be considered.
- The specification should be as simple as possible in order for them to be easily implemented by software vendors.
- The requirements for the validation rules are:
 - The validation rules must support expressions with the following operators: \leq , $<$, $>$, \geq , $=$, \neq , and, or.
 - The validation rules must support expressions like if ... then ... else
 - The validation rules must support grouping within expressions with brackets.
 - The validation rules must support expressions that evaluated to true or false.
- This specification must allow for the validation rules to be outside the DTS of the target XBRL instance document so regular XBRL software is not confronted with unsupported syntax in the DTS.
- The processing model must support statistics about number of evaluations, failures and successes in the output content model.

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

3 Assumptions

In order to constrain the scope of this specification, the following assumptions have been made. Processors conforming to this specification are not required to validate these assumptions.

3.1 Monetary unit

All monetary facts in an XBRL instance document are reported against a single monetary unit. Therefore no currency conversion facilities are provided with this specification.

3.2 Nillable

No facts in the instance document are reported as "nill" (i.e. with xsi:nil attribute present and set to true). Therefore no interpretation issues for these sorts of facts need to be resolved by this specification.

3.3 Association

The validation rules governed by this specification are optionally used by XBRL processors. Validation rules files do not have to be part of a Discoverable Taxonomy Set so regular XBRL processors do not have to be affected by this specification.

3.4 Floating point

The validation rules governed by this specification are crafted to handle floating point rounding themselves. Therefore no floating point rounding rules are provided with this specification.

3.5 Preconditions

The validation rules governed by this specification are crafted to include their own preconditions. Therefore no precondition facilities are provided with this specification.

3.6 Context grouping

The validation rules governed by this specification are to be evaluated according to specific groups of xbrli:contexts. Such groups can be obtained by applying grouping patterns to the available contexts in an XBRL instance document. This specification provides a limited set of context grouping patterns among which are a movement analysis pattern, single instant pattern and single duration pattern. It is expected that there might be additional patterns identified in the future, for example based on domain-member relationships.

3.7 Expression language

The validation rules governed by this specification can be expressed with XPath 1.0 expressions that are to be evaluated in the context of an XPath context, and a small number of extension functions. Therefore no additional expression syntax needs to be defined by this specification.

3.8 Tuples

The validation rules governed by this specification can be applicable for facts in the root xbrl element or facts in repeating structures of tuples.

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

4 Syntax

This specification only provides a textual declaration of syntax constraints when those constraints are not expressed by the normative schema supplied with this specification.

Explanations of elements and attributes are only supplied when those explanations are not already provided in other specifications.

4.1 XPath usage

Some attributes defined by this specification contain values that are evaluated as XPath 1.0 expressions, see <http://www.w3.org/TR/xpath>. Wherever an XPath expression is mentioned in this specification it refers to an XPath 1.0 expression. The evaluation context for an XPath expression is determined by this specification and is depending the location of such expression. The statically known namespaces used in the expressions are those that are in scope for the element that has the attribute or element content that contains the XPath expression.

QNames in each XPath expression MUST be resolvable using namespace declarations that are in scope for the element that contains the XPath expression. See Namespaces in XML 1.0 <http://www.w3.org/TR/REC-xml> for more information on namespace declarations and their scope.

4.1.1 Custom functions

This specification declares a number of custom functions which MAY be used within the XPath expressions. Custom functions other than those declared in this specification MAY also be referenced by XPath expressions.¹

4.2 Rule resources

Validation rules are declared using the `svr:rule` resource element in an XBRL generic link. Such a declaration is referred to as a rule. The generic link is contained in a linkbase which is referred to as a standard-validation-rule linkbase, the `svr` linkbase.

4.2.1 expression

The value of a `rule` element is an XPath expression with a boolean result. It is referred to as the validation rule expression. When evaluated it MUST either evaluate successfully or fail. The expression is an XPath expression in which validation functions (see chapter 6) are used to identify the elements in the XBRL instance document.

4.2.2 contextGroupingRole

A `rule` has a `contextGroupingRole` attribute that is an URI identifying the context grouping pattern by which contexts appearing in the validated XBRL instance document must be grouped in order to apply the rule (see also chapter 5). Each context grouping pattern prescribes a way to determine one or more unique names to which `xbrli:contexts` are assigned when they are part of a group, for example by using fixed names such as 'previous', 'movement' and 'current'. When a context grouping pattern is applied to an XBRL instance document it will result in zero or more groups of contexts. The validation rule expression is to be evaluated for each group found for the context grouping pattern with which it is associated. The order of the groups may be specified by the

¹ Whenever a processor encounters a 'not available' custom function in a rule, the complete rule should be marked as incompatible with the used processor.

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

pattern. Whether only fully populated groups are to be used for evaluation is to be specified by the pattern.

4.2.3 *contextNodes*

A `rule` has a `contextNodes` attribute that is to be evaluated as an XPath expression against the validated XBRL instance document. The resulting XPath nodeset provides one or more XPath context nodes against which the `rule`'s value expression can be evaluated. This allows the rules to be made applicable for facts in the root `xbml` element or in repeating structures of tuples.

4.2.4 *message*

A `rule` has a `message` attribute of type `xsd:string` that provides the message that will be shown in the event of a failure of the rule. This may be a full text message or just a message qualifier.

4.2.5 *Rule name*

A `rule` has a `name` attribute of type `xsd:string` that provides an identifier for the rule. The name must be unique within the Discoverable Taxonomy Set of the applicable SVR linkbases.

4.2.6 *Rule definition*

A `rule` has a `definition` attribute of type `xsd:string` that provides human readable documentation of the rule.

4.3 Rule relationships

Validation rules are declared using the `svr:rule` resource element in an XBRL generic link. Rules may have relationships with other rules. Such relationships are referred to as preceding-following relationships. They are created using generic arcs with an arc-role of `http://sbr-xbrl.org/arcrole/2008/preceding-following`

The following constraints apply to arcs using the <http://sbr-xbrl.org/arcrole/2008/preceding-following> arcrole:

- The source and target must both be a rule;
- Have an order attribute with a numeric value.

No cycles are allowed in networks of preceding-following arcs. Validating processors MUST detect cycles in networks of preceding-following arcs.

4.3.1 *Number of rules*

Each network must have at least two (2) or more rules.

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

5 Context grouping patterns

The validation rules governed by this specification are to be evaluated according to specific groups of `xbrli:contexts`. Such groups can be obtained by applying grouping patterns to the available contexts in an XBRL instance document.

The context grouping patterns collect the various contexts from the XBRL instance document in coherent groups as specified in each pattern. Within each group the pattern will label the contexts with names as specified. During processing the rule will be evaluated sequentially against each group of contexts identified by the `contextGroupingRole` pattern.

The patterns allow rules to address facts from different contexts that were found in the XBRL instance document by using the labeled names for the contexts in conjunction with the fact names.

For example to get assets net for a context that is labeled 'current' in a movement analysis:

```
xyz:AssetsNet, 'current'
```

This specification defines the following context grouping patterns:

- movement analysis pattern with complete groups;
- movement analysis pattern with optionally empty places in the groups;
- single instant pattern;
- single duration pattern;
- single context pattern;
- duration with connecting instant pattern;
- instant with connecting duration pattern.

There might be additional patterns identified in the future, for example based on domain-member relationships. The additional patterns must be specified by a text that is available at the URL that is equal to the URI that appears in the `contextGroupingRole` attribute of a validation rule.

Processing grouping patterns

A validating processor MUST perform context grouping as specified by the pattern that is identified with the `contextGroupingRole` attribute on a validation rule before the rule's expression is evaluated. The processor MUST signal with a Consistency Check Failure when the identified pattern is unknown.

5.1 Movement-complete

The movement-complete pattern is used for rules where the movement between the previous and the current period is validated. Each element in the rule must refer to one of the contextnames 'previous', 'movement' or 'current'.

This pattern allows only groups that are complete. If one or more of the required contexts for a group are not available in the XBRL instance document, the group will be ignored during the validation of that XBRL instance document.

This means that rules are only applied to those sets of data in the XBRL instance document that cover the complete spectrum.

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

For example in an XBRL instance document with 4 contexts (shorthand) year2005-end2005-year2006-end2006 only

end2005-year2006-end2006

will be used in a movement-complete context group.

This context grouping pattern is identified with the following URI:

<http://www.sbr-xbrl.org/validation/contextgrouping/movement-complete>

Execution of the movement-complete pattern

When applying this context grouping pattern validating processors MUST:

- Create groups of contexts for which the combined periods form closed calendar periods with a start, a duration and an end for the same entity and scenario.
- Have three positions in each group that can be referred by the contextnames: 'previous', 'movement' and 'current'.
- Place each context with an instant period in a group in the position 'current'.
- Place each context with a duration period in an existing group, for which the context in the 'current' position has an instant period connecting to the end of the duration period, in the position 'movement'. If no matching group can be found create a new group for the context.
- Place each context with an instant period in an existing group, for which the context in the 'movement' position has the begin of a duration period connecting to the instant period, in the position 'previous'. If no matching group can be found create a new group for the context.
- Only allow groups that are complete to be returned so 'movement tests can be performed without testing for null.

5.2 Movement-all

The movement-all pattern is used for rules where the movement between the previous and the current period is validated. Each element in the rule must refer to one of the contextnames 'previous', 'movement' or 'current'.

This pattern allows groups that are complete or incomplete (= all). If one or more of the required contexts are not available in the XBRL instance document, the group will still be used during the validation of that XBRL instance document.

This means that rules are applied to those sets of data in the XBRL instance document that cover both the complete spectrum and 'half' sets of data at the beginning or end of the report.

For example in an instance report with 4 contexts (shorthand) year2005-end2005-year2006-end2006 both

year2005-end2005
and
end2005-year2006-end2006

will be used in a movement-all context groups.

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

This means that in the rule itself the 'test for null' MUST be included to get rules to pass for the 'half' context group of 2005.

This context grouping pattern is identified with the following URI:

<http://www.sbr-xbrl.org/validation/contextgrouping/movement-all>

Execution of the movement-all pattern

When applying this context grouping pattern validating processors MUST:

- Create groups of contexts for which the combined periods form closed calendar periods with a start, a duration and an end for the same entity and scenario.
- Have three positions in each group that can be referred by the contextnames: 'previous', 'movement' and 'current'.
- Place each context with an instant period in a group in the position 'current'.
- Place each context with a duration period in an existing group, for which the context in the 'current' position has an instant period connecting to the end of the duration period, in the position 'movement'. If no matching group can be found create a new group for the context.
- Place each context with an instant period in an existing group, for which the context in the 'movement' position has a begin of a duration period connecting to the instant period, in the position 'previous'. If no matching group can be found create a new group for the context.
- Allow groups that are incomplete so "not occurs" tests can be performed. Note that movement tests must include testing for null.

5.3 Single instant

The single-instant pattern is used when the rule only refers to elements with a periodtype 'instant'. For each element in the rule the reference to the contextname is 'context'.

This context grouping pattern is identified with the following URI:

<http://www.sbr-xbrl.org/validation/contextgrouping/single-instant>

Execution of the single instant pattern

When applying this context grouping pattern validating processors MUST:

- Create groups for each single instant context.
- Have one position in each group that can be referred by the contextname: 'context'.
- Place each context with an instant period in a group in the position 'context'.

5.4 Single duration

The single-duration pattern is used when the rule only refers to elements with a periodtype 'duration'. For each element in the rule the reference to the contextname is 'context'.

This context grouping pattern is identified with the following URI:

<http://www.sbr-xbrl.org/validation/contextgrouping/single-duration>

Execution of the single duration pattern

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

When applying this context grouping pattern validating processors MUST:

- Create groups for each single duration context.
- Have one position in each group that can be referred by the contextname: 'context'.
- Place each context with an duration period in a group in the position 'context'.

5.5 Single context

The single context pattern is used when it doesn't matter whether the elements used in the rule are instant or duration. For each element in the rule the reference to the contextname is 'context'.

This context grouping pattern is identified with the following URI:

<http://www.sbr-xbrl.org/validation/contextgrouping/single>

Execution of the pattern

When applying this context grouping pattern validating processors MUST:

- Create groups for each single context.
- Have one position in each group that can be referred by the contextname: 'context'.
- Place each context in a group in the position 'context'.

5.6 Duration-Instant

The duration-instant pattern is used for contexts that form closed calendar periods with a duration and an end (instant) for the same entity and scenario. For each element in the rule the reference to the contextname is either 'duration' or 'instant'.

This pattern allows for rules that check facts that are reported for a period against other facts that were reported for the last or end moment of that same period.

This context grouping pattern is identified with the following URI:

<http://www.sbr-xbrl.org/validation/contextgrouping/duration-instant>

Execution of the pattern

When applying this context grouping pattern validating processors MUST:

- Create groups of contexts for which the combined periods form closed calendar periods with a duration and an end for the same entity and scenario.
- Have two positions in each group that can be referred by the contextnames 'duration' and 'instant'.
- Place each context with an instant period in a group in the position 'instant'.
- Place each context with a duration period in an existing group, for which the context in the 'instant' position has an instant period connecting to the end of the duration period, in the position 'duration'. If no matching group can be found create a new group for the context.
- Only allow groups that are complete to be returned.

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

5.7 Instant-Duration

The instant-duration pattern is used for contexts that form closed calendar periods with a duration and an begin (instant) for the same entity and scenario. For each element in the rule the reference to the contextname is either 'duration' or 'instant'.

This pattern allows for rules that check facts that are reported for a period against other facts that were reported for the start or first moment of that same period.

This context grouping pattern is identified with the following URI:

<http://www.sbr-xbrl.org/validation/contextgrouping/instant-duration>

Execution of the pattern

When applying this context grouping pattern validating processors MUST:

- Create groups of contexts for which the combined periods form closed calendar periods with a duration and an end for the same entity and scenario.
- Have two positions in each group that can be referred by the contextnames 'duration' and 'instant'.
- Place each context with an instant period in a group in the position 'instant'.
- Place each context with a duration period in an existing group, for which the context in the 'instant' position has an instant period connecting to the begin of the duration period, in the position 'duration'. If no matching group can be found create a new group for the context.
- Only allow groups that are complete to be returned.

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

6 Validation functions

This specification provides a number of extension of validation functions which MAY be referenced within XPath expressions.

Validation functions other than those declared in this specification MAY also be referenced by XPath expressions.²

The items arguments in these validation functions are item facts from the XBRL instance document that are identified with an XPath expression using fact and optionally tuple names. The XPath expression MUST be rooted at the XPath context nodes that are obtained by evaluating the rule's contextNodes attribute. For example if the contextNodes attribute is /xbrli:xbrl and the total amount item fact must be addressed while it is to be reported inside a xyz-tuple the items expression could be:

```
./xyz-tuple:xyzData/xyz:TotalAmount
```

6.1 getItemFactForContext

This validation function provides access to facts in the XBRL instance document associated with the contexts that have been grouped into context groups.

```
node svr:getItemFactForContext (items, contextName)
```

6.1.1 Parameters

Items	A node-set of item facts. A processor MUST signal an error if the node-set contains nodes that are not item facts.
ContextName	A string identifying a position within the current context group. A processor MUST signal an error if the string does not correspond to a position within the current context group.

6.1.2 Return

Node	A node which is the item fact node that was present in the items nodeset and refers to the context in the named position in the current context group if both context and item are present and an empty nodeset otherwise. Return the first in document order if more then one fact was found.
------	--

6.2 getItemFactForContextOrZero

This validation function is used when omitting a fact should be interpreted as filling zero; for example in summation checks.

```
node svr:getItemFactForContextOrZero (items, contextName)
```

² Whenever a processor encounters a 'not available' validation function in a rule, the complete rule should be marked as incompatible with the used processor.

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

6.2.1 Parameters

Items A node-set of item facts. A processor MUST signal an error if the node-set contains nodes that are not item facts.

ContextName A string identifying a position within the current context group. A processor MUST signal an error if the string does not correspond to a position within the current context group.

6.2.2 Return

node A node which is the item fact node that was present in the items nodeset and refers to the context in the named position in the current context group if both context and item are present and a value of 0.0 otherwise. Return the first in document order if more than one fact was found.

6.3 getItemFactsOrZero

This validation function is used when omitting a fact should be interpreted as filling zero; for example in summation checks.

```
node-set svr:getItemFactsOrZero(items)
```

6.3.1 Parameters

Items A node-set of item facts. A processor MUST signal an error if the node-set contains nodes that are not item facts.

6.3.2 Return

node-set The node-set identified by Items if not empty, or a node-set with a node with no name and value 0.0 otherwise.

6.4 getItemFactsForContextOrZero

This validation function is used when omitting a fact should be interpreted as filling zero; for example in summation checks.

```
node-set svr:getItemFactsForContextOrZero(items,contextName)
```

6.4.1 Parameters

Items A node-set of item facts. A processor MUST signal an error if the node-set contains nodes that are not item facts.

contextName A string identifying a position within the current context group. A processor MUST signal an error if the string does not correspond to a position within the current context group.

6.4.2 Return

node-set A node-set which is the set of items present in the items node-set and referring to the context in the named position in the current context group if the context is present and the resulting node-set is not empty. Otherwise, a node with no name and value 0.0.

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

6.5 getItemFactsForContextsOrZero

This validation function is used to obtain facts for multiple contexts (for example one per subsidiary) and when omitting such facts should be interpreted as filling zero; for example in summation checks over multiple contributing contexts.

```
node-set svr:getItemFactsForContextsOrZero (items, contextName)
```

6.5.1 Parameters

Items A node-set of item facts. A processor MUST signal an error if the node-set contains nodes that are not item facts.

contextName A string identifying a position within the current context group. A processor MUST signal an error if the string does not correspond to a position within the current context group.

6.5.2 Return

node-set A node-set with the item fact nodes that were present in the items nodeset and referring to one of the named contexts in the current context group if both one or more contexts and items are present and a node-set with a node with no name and value 0.0 otherwise.

6.6 minimum

This validation function is used to define the numeric object with the smallest value.

```
object svr:minimum(object, object)
```

6.6.1 Parameters

Object An object that has a numeric value. Only Double, Node and NodeList with one Node objects are allowed. A processor MUST signal an error if other objects are specified.

Object An object that has a numeric value. Only Double, Node and NodeList with one Node objects are allowed. A processor MUST signal an error if other objects are specified.

6.6.2 Return

Object From the two argument objects, the object with the smallest value. The first object if they are equal.

6.7 maximum

This validation function is used to define the numeric object with the largest value.

```
object svr:maximum(object, object)
```

6.7.1 Parameters

Object An object that has a numeric value. Only Double, Node and NodeList with one Node objects are allowed. A processor MUST signal an error if other objects are specified..

Object An object that has a numeric value. Only Double, Node and NodeList with one Node objects are allowed. A processor MUST signal an error if other objects are specified.

Version: 1.0
Date: 22 june 2009
Status: Recommendation
Classification: Public

6.7.2 *Return*

Object From the two argument objects, the object with the largest value. The first object if they are equal.

6.8 **date**

This validation function is used to convert the date value to a value that can be used in compare operations.

```
double svr:date(object)
```

6.8.1 *Parameters*

Object An object that has a date value, patterns yyyy-MM-dd and dd-MM-yyyy must be supported. A processor MUST signal an error if the object is not a string conforming to the above pattern.

6.8.2 *Return*

Double The date value in the pattern yyyyMMdd so that comparison operations can be performed on the date values.

6.9 **getUnit**

This function is used to obtain the unit with the given string.

```
node svr:getUnit(string)
```

6.9.1 *Parameters*

String A unit id.

6.9.2 *Return*

Node An Xbrii:unit node that has the given string as value in the id attribute on the xbrii:unit or null if none found.

6.10 **testPattern**

This function is used to test if a given string can be matched by a regular expression. Rule expressions may use this function to evaluate datatype restrictions such as patterns for dates and strings.

```
boolean svr:testPattern(string,string)
```

6.10.1 *Parameters*

String A regular expression.

String A value string.

6.10.2 *Return*

boolean Returns true if the entire value string from the first to the last character can be matched by the regular expression. Otherwise returns false.

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

7 Processing model

This section defines the processing model for a svr linkbase. The processing model is defined to perform all evaluation cycles for all combinations of context groups and XPath context nodes for all rules on a given XBRL instance document.

7.1 Validation rule networks

The validating processor MUST discover all networks of preceding-following relationships between rule resources in svr linkbases that are associated with the XBRL instance document that is to be validated.

All networks must be evaluated in alphabetical order based on the xlink:role URI that was used on the extendedLinks that contributed relationships to the network. Only rules that have relationships with other rules will appear in a network.

All rules in a network that have no relationships pointing towards them are the network start rules. The network start rules must be evaluated in alphabetical order based on the name attribute on the rule. All rules in a network start rule's network of preceding-following relationships must be evaluated in the order of a depth-first traversal. Sibling nodes are ordered according to the numeric order of their order attributes.

7.2 Context grouping

Before evaluating a rule expression the validating processor MUST perform context grouping on the xbrli:contexts found in the validated XBRL instance document using the pattern identified with the contextGroupingRole attribute on the rule.

7.3 Context nodes

Before evaluating a rule expression the validating processor MUST evaluate the XPath expression that is the value of contextNodes attribute on the rule with the validated XBRL instance document as XPath context node. The result is a node-set with zero or more XPath context nodes to be used for evaluating the rule expression.

7.4 Rule evaluation

The validating processor MUST evaluate the validation rule expression for each combination of XPath context node and context group that were resolved for the current rule.

7.4.1 Example rule:

```
<svr:rule
xlink:label="xyz_AssetsNet_rule"
xlink:type="resource"
contextGroupingRole=
"http://www.sbr-xbrl.org/validation/contextgrouping/movement-all"
contextNodes="/"
message="AssetsNet is not consistent."
name="CheckAssetsNet"
definition="Check AssetsNet consistency.">
getItemFact(/xyz:AssetsNet,'current')=getItemFact(/xyz:AssetsNet,'previous')+getIte
mFact(/xyz:ChangesInAssetsNet,'movement')
</svr:rule>
```

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

7.5 Producing validation result

A validating processor **MUST** produce a `svr:validationresult` element for each validated XBRL instance document.

The `svr:validationresult` **MUST** have children when one or more validation rules were evaluated.

7.5.1 Networkresult

A validating processor **MUST** produce a `svr:networkresult` element per evaluated network of validation rules.

A `svr:networkresult` **MUST** have an attribute `xlink:role` that has the value of the `xlink:role` attribute on the extended link elements from which the relationships in the network originated. This allows validating applications to order messages per user oriented role e.g. balance-sheet or cash-flow-direct.

When networks of validation rules are evaluated failure of one or more rules will produce message results. All message results for each network **MUST** be collected in a `svr:networkresult` element. The `svr:networkresult` **MAY** have children. When no validation rule in the network failed `svr:networkresult` will have no children.

The `svr:networkresult` **MUST** have the following attributes:

- `rules`, an int that shows the number of rules that were evaluated for this network result;
- `cycles`, an int that shows the total number of times rule expressions were evaluated for this `networkresult`;
- `cyclesFailed`, an int that shows the total number of times rule expressions were evaluated for this `networkresult` with a result false;
- `cyclesSuccess`, an int that shows the total number of times rule expressions were evaluated for this `networkresult` with a result true.

7.5.2 Rule result

For each evaluation cycle of a validation rule expression that fails a `svr:ruleresult` element must be produced as a child of the current `svr:networkresult`.

A `svr:ruleresult` **MUST** have a attribute `message` which has a value equal to the value of the `message` attribute on the failed validation rule.

A `svr:ruleresult` **MUST** have a attribute `name` which has a value equal to the value of the `name` attribute on the failed validation rule.

A `svr:ruleresult` **MUST** have all `xbri:item` facts that were obtained from the validated XBRL instance document for the failed evaluation cycle of the rule's expression, as children, in the order by which they were obtained by the validating processor while evaluating the validation rule expression.

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

7.5.3 Example result:

```
<svr:validationresult
  xmlns:svr="http://sbr-xbrl.org/2008/standard-validation-rules"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:bd="http://www.taxonomie.sbr/2.0/domein/bd">
  <svr:networkresult
    xlink:role="http://www.taxonomie.sbr/2.0/domain/xyz/formset/balans/fs-xyz-abc">
    <svr:ruleresult
      message="AssetsNet is not consistent."
      name="vpb:AssetsNet">
      <xyz:AssetsNet ... contextRef="eind2008">190</xyz:AssetsNet>
      <xyz:AssetsNet ... contextRef="periode2008">100</xyz:AssetsNet>
      <xyz:AssetsNet ... contextRef="eind2007">100</xyz:AssetsNet>
    </svr:ruleresult>
  </svr:networkresult>
</svr:validationresult>
```

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

8 Schema

The following are the versions of the XML schemas provided as part of this specification. These are all normative. Non-normative versions (which should be identical to these except for appropriate comments indicating their non-normative status) are also provided as separate files for convenience of users of the specification.

8.1 standard-validation-rules-2008-08.xsd (normative)

```

<?xml version="1.0" encoding="UTF-8"?>
<schema
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:svr="http://sbr-xbrl.org/2008/standard-validation-rules"
  xmlns:xbrli="http://www.xbrl.org/2003/instance"
  xmlns:xl="http://www.xbrl.org/2003/XLink"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  elementFormDefault="qualified"
  targetNamespace="http://sbr-xbrl.org/2008/standard-validation-rules">

  <import namespace="http://www.w3.org/1999/xlink"
    schemaLocation="http://www.xbrl.org/2003/xlink-2003-12-31.xsd"/>
  <import namespace="http://www.xbrl.org/2003/instance"
    schemaLocation="http://www.xbrl.org/2003/xbrl-instance-2003-12-31.xsd"/>
  <import namespace="http://www.xbrl.org/2003/XLink"
    schemaLocation="http://www.xbrl.org/2003/xl-2003-12-31.xsd"/>

  <element id="xml-rule" name="rule" substitutionGroup="xl:resource">
    <annotation>
      <documentation>Definition of the rule resource element. The standard type value
of a rule instance is the rule expression.</documentation>
    </annotation>
    <complexType mixed="true">
      <complexContent mixed="true">
        <extension base="xl:resourceType">
          <attribute name="contextGroupingRole" type="xl:nonEmptyURI" use="required"/>
          <attribute name="contextNodes" type="string" use="required"/>
          <attribute name="message" type="string" use="required"/>
          <attribute name="name" type="QName" use="required"/>
          <attribute name="definition" type="string" use="required"/>
        </extension>
      </complexContent>
    </complexType>
  </element>

```

Version: 1.0
Date: 22 June 2009
Status: Recommendation
Classification: Public

```

<element id="xml-validationresult" name="validationresult">
  <annotation>
    <documentation>Definition of the validationresult element.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element maxOccurs="unbounded" minOccurs="1" ref="svr:networkresult"/>
    </sequence>
  </complexType>
</element>

<element id="xml-networkresult" name="networkresult">
  <annotation>
    <documentation>Definition of the networkresult element.</documentation>
  </annotation>
  <complexType>
    <sequence>
      <element maxOccurs="unbounded" minOccurs="0" ref="svr:ruleresult"/>
    </sequence>
    <attribute ref="xlink:role" use="required"/>
    <attribute name="rules" type="int" use="required"/>
    <attribute name="cycles" type="int" use="required"/>
    <attribute name="cyclesFailed" type="int" use="required"/>
    <attribute name="cyclesSuccess" type="int" use="required"/>
  </complexType>
</element>

<element id="xml-ruleresult" name="ruleresult">
  <annotation>
    <documentation>Definition of the ruleresult element.</documentation>
  </annotation>
  <complexType>
    <sequence>
      <element ref="xbrli:item"/>
    </sequence>
    <attribute name="name" type="string" use="required"/>
    <attribute name="message" type="string" use="required"/>
  </complexType>
</element>

</schema>

```