

Package ‘RXMLCDA’

April 1, 2011

Version 1.4.0

Date 2011-03-30

Title RXMLCDA

Author Patrick Meyer, Sebastien Bigaret.

Maintainer Patrick Meyer <patrick.meyer@telecom-bretagne.eu>

Description The RXMLCDA library for the R statistical software allows you to read many XMCDA tags and transform them into R variables which are then usable in your algorithms written in R. The library also allows to write certain R variables into XML files according to the XMCDA standard.

Depends R(>= 2.1.0), methods, XML

Suggests kappalab

Encoding latin1

License CeCILL-2

URL <http://www.decision-deck.org/xmcda>

R topics documented:

checkXSD	2
getAlternativesComparisons	3
getAlternativesComparisonsLabels	4
getAlternativesIDs	6
getAlternativesIntervalValues	7
getAlternativesValues	8
getCriteriaComparisons	9
getCriteriaComparisonsLabels	10
getCriteriaIDs	11
getCriteriaIntervalValues	12
getCriteriaPairsComparisons	13
getCriteriaPairsIntervalValues	14

getCriteriaPairsValues	15
getCriteriaValues	16
getMobiusCapacities	17
getNumberOfAlternatives	18
getNumberOfCriteria	19
getNumericValue	20
getParameters	20
getPerformanceTables	21
putAlternativesComparisonsLabels	22
putAlternativesPlot	23
putAlternativesValues	24
putAlternativeValue	25
putCapacity	26
putCriteriaPairsValues	27
putCriteriaPlot	28
putCriteriaValues	29
putCriterionValue	30
putErrorMessage	31
putLogMessage	31
putMessage	32
putPerformanceTable	32
putPointsCriterionFunction	33
Index	35

checkXSD

XMCD A tree validation.

Description

Checks if an XML tree is valid according to an XML schema.

Usage

```
checkXSD (tree)
```

Arguments

`tree` Object containing the XMCD A XML tree.

Value

The function returns 1 if the XML tree is validated by the XMCD A schema (currently XMCD A-2.*), else it returns 0.

Examples

```

tree = newXMLDoc()

newXMLNode("xmcda:XMCD", namespace = c("xsi" = "http://www.w3.org/2001/XMLSchema-instance",

root<-getNodeSet(tree, "/xmcda:XMCD")

criteria<-newXMLNode("criteria", parent=root[[1]], namespace=c())

criterion<-newXMLNode("criterion", attrs = c(id="g1"), parent=criteria, namespace=c())

y<-checkXSD(tree)

```

```

getAlternativesComparisons
      Get comparisons of alternatives

```

Description

Extracts `<alternativesComparisons>` from an XML tree written according to the XMCD standard and stores the performances of the concerned alternatives.

Usage

```
getAlternativesComparisons(tree, performanceTable, mcdaConcept = NULL)
```

Arguments

<code>tree</code>	Object containing the XMCD XML tree.
<code>performanceTable</code>	A matrix containing the performance table to be considered, whose dimnames are the alternatives' IDs (rows) and the criteria's IDs (columns).
<code>mcdaConcept</code>	A string containing the specific <code>mcdaConcept</code> attribute which should be searched for.

Value

The function returns a list structured as follows:

- The first elements contain the `<alternativesComparisons>` found in `<tree>` as matrices. These elements are named according to the `mcdaConcept` attribute if it can be found. Each line of each matrix corresponds to one constraint of the type "alternative a is preferred to alternative b with preference threshold delta". A line is structured as follows: the first `n` elements encode alternative a (via the values it takes in `performanceTable`), the next `n` elements encode alternative b, and the last element contains the preference threshold delta.
- `status` Either OK if all the `<criteria>` tags could be correctly read, or the description of the error.

Examples

```

tree = newXMLDoc()

newXMLNode("xmcda:XMCD", namespace = c("xsi" = "http://www.w3.org/2001/XMLSchema-instance",

root<-getNodeSet(tree, "/xmcda:XMCD")

aC<-newXMLNode("alternativesComparisons", parent=root[[1]] , namespace=c())

pairs<-newXMLNode("pairs", parent=aC, namespace=c())
pair<-newXMLNode("pair", parent=pairs, namespace=c())

initial<-newXMLNode("initial", parent=pair)
newXMLNode("alternativeID", "a01", parent=initial, namespace=c())

terminal<-newXMLNode("terminal", parent=pair, namespace=c())
newXMLNode("alternativeID", "a02", parent=terminal, namespace=c())

value<-newXMLNode("value", parent=pair, namespace=c())
newXMLNode("real", "0.5", parent=value, namespace=c())

alternativesIDs <- c("a01","a02")
criteriaIDs <- c("g1","g2")

performance.table<-matrix(nrow=length(alternativesIDs),ncol=length(criteriaIDs),dimnames = 1

for (i in 1:length(alternativesIDs)){
  for (j in 1:length(criteriaIDs)){
    performance.table[i,j] = runif(1)
  }
}

y<-getNodeSet(tree,"//alternativesComparisons")

x<-getAlternativesComparisons(y[[1]], performance.table)

```

```

getAlternativesComparisonsLabels
  Get comparisons of alternatives

```

Description

Extracts `<alternativesComparisons>` from an XML tree written according to the XMCD standard and stores the IDs of the concerned alternatives.

Usage

```
getAlternativesComparisonsLabels(tree, altIDs=NULL, mcdaConcept = NULL)
```

Arguments

tree	Object containing the XMCDAs XML tree.
altIDs	A vector containing the IDs of the alternatives to be considered for the extractions.
mcdaConcept	A string containing the specific mcdaConcept attribute which should be searched for.

Value

The function returns a list structured as follows:

-	The first elements contain the <alternativesComparisons> found in <tree> as matrices. These elements are named according to the mcdaConcept attribute if it can be found. Each line of each matrix corresponds to one constraint of the type "alternative a is preferred to alternative b with preference threshold delta". A line is structured as follows: the first element encodes the ID of alternative a, the second element encodes the ID of alternative b, and the last element contains the preference threshold delta.
status	Either OK if all the <alternativesComparisons> tags could be correctly read, or the description of the error.

Examples

```
tree = newXMLDoc()

newXMLNode("xmcdas:XMCDAs", namespace = c("xsi" = "http://www.w3.org/2001/XMLSchema-instance",
root<-getNodeSet(tree, "/xmcdas:XMCDAs")

aC<-newXMLNode("alternativesComparisons", parent=root[[1]], namespace=c())

pairs<-newXMLNode("pairs", parent=aC, namespace=c())
pair<-newXMLNode("pair", parent=pairs, namespace=c())

initial<-newXMLNode("initial", parent=pair, namespace=c())
newXMLNode("alternativeID", "a01", parent=initial, namespace=c())

terminal<-newXMLNode("terminal", parent=pair, namespace=c())
newXMLNode("alternativeID", "a02", parent=terminal, namespace=c())

value<-newXMLNode("value", parent=pair, namespace=c())
newXMLNode("real", "0.5", parent=value, namespace=c())

alternativesIDs <- c("a01","a02")

y<-getNodeSet(tree,"//alternativesComparisons")

x<-getAlternativesComparisonsLabels(y[[1]], alternativesIDs)
```

getAlternativesIDs *Get IDs of alternatives*

Description

Gets the IDs of alternatives, stored in the <alternatives> tag, from an XML tree written according to the XMCCA standard.

Usage

```
getAlternativesIDs(tree, mcdaConcept = NULL)
```

Arguments

tree	Object containing the XMCCA XML tree.
mcdaConcept	A string containing the specific mcdaConcept attribute which should be searched for.

Value

The function returns a list structured as follows:

-	The first elements contain vectors with the alternatives' IDs which have been found in each <alternatives> tag. These elements are named according to the mcdaConcept attribute if it can be found.
status	Either OK if all the <alternatives> tags could be correctly read, or the description of the error.

Examples

```
tree = newXMLDoc()

newXMLNode("xmcca:XMCCA", namespace = c("xsi" = "http://www.w3.org/2001/XMLSchema-instance",

root<-getNodeSet(tree, "/xmcca:XMCCA")

alternatives<-newXMLNode("alternatives", attrs=c(mcdaConcept="actions"), parent=root[[1]], n

alternative<-newXMLNode("alternative", attrs = c(id="x1"), parent=alternatives, namespace=c()

alternative<-newXMLNode("alternative", attrs = c(id="x2"), parent=alternatives, namespace=c()

alternative<-newXMLNode("alternative", attrs = c(id="x3"), parent=alternatives, namespace=c()

y<-getNodeSet(tree, "//alternatives")

x<-getAlternativesIDs(y[[1]])
```

`getAlternativesIntervalValues`*Get interval values related to alternatives*

Description

Gets intervals of values related to alternatives, stored in the `<alternativesValues>` tag, from an XML tree written according to the XMCDa standard.

Usage

```
getAlternativesIntervalValues(tree, alternativesIDs, mcdaConcept = NULL)
```

Arguments

<code>tree</code>	Object containing the XMCDa XML tree.
<code>alternativesIDs</code>	A vector containing the IDs of the alternatives to be considered for the extractions.
<code>mcdaConcept</code>	A string containing the specific <code>mcdaConcept</code> attribute which should be searched for.

Value

The function returns a list structured as follows:

- The first elements contain matrices representing each an `<alternativesValues>`. Each line of each matrix corresponds to an element of the type "the interval of values assigned to alternative a is [x,y]". A line is structured as follows: the first element encodes the index of the ID of alternative a in `alternativesIDs`, the second element encodes the value x, and the last element encodes the value y. These elements are named according to the `mcdaConcept` attribute if it can be found.
- `status` Either OK if all the `<alternativesValues>` tags could be correctly read, or the description of the error.

Examples

```
tree <- xmlTreeParse(system.file("extdata", "testFile.xml", package="RXMCDa"), useInternalNodes = TRUE)
altIDs <- getAlternativesIDs(tree)
altVals <- getAlternativesIntervalValues(tree, altIDs[[1]], mcdaConcept="alternativesIntervalValues")
```

```
getAlternativesValues
```

Get values related to alternatives

Description

Gets values related to alternatives, stored in the `<alternativesValues>` tag, from an XML tree written according to the XMCDa standard.

Usage

```
getAlternativesValues(tree, alternativesIDs, mcdaConcept = NULL)
```

Arguments

<code>tree</code>	Object containing the XMCDa XML tree.
<code>alternativesIDs</code>	A vector containing the IDs of the alternatives to be considered for the extractions.
<code>mcdaConcept</code>	A string containing the specific <code>mcdaConcept</code> attribute which should be searched for.

Value

The function returns a list structured as follows:

<code>-</code>	The first elements contain matrices representing each an <code><alternativesValues></code> . Each line of each matrix corresponds to an element of the type "the value assigned to alternative a is x". A line is structured as follows: the first element encodes the index of the ID of alternative a in <code>alternativesIDs</code> , the second element encodes the value x. These elements are named according to the <code>mcdaConcept</code> attribute if it can be found.
<code>status</code>	Either OK if all the <code><alternativesValues></code> tags could be correctly read, or the description of the error.

Examples

```
tree <- xmlTreeParse(system.file("extdata", "testFile.xml", package="RXMCDa"), useInternalNode
altIDs <- getAlternativesIDs(tree)
altVals <- getAlternativesValues(tree, altIDs[[1]])
```

```
getCriteriaComparisons
```

Get comparisons of criteria

Description

Gets comparisons of criteria, stored in the `<criteriaComparisons>` tag, from an XML tree written according to the XMCDa standard.

Usage

```
getCriteriaComparisons(tree, criteriaIDs, mcdaConcept = NULL)
```

Arguments

<code>tree</code>	Object containing the XMCDa XML tree.
<code>criteriaIDs</code>	A vector containing the IDs of the criteria to be considered for the extractions.
<code>mcdaConcept</code>	A string containing the specific <code>mcdaConcept</code> attribute which should be searched for.

Value

The function returns a list structured as follows:

<code>-</code>	The first elements contain the <code><criteriaComparisons></code> found in <code><tree></code> as matrices. These elements are named according to the <code>mcdaConcept</code> attribute if it can be found. Each line of each matrix corresponds to one constraint of the type "criterion g1 is preferred to criterion g2 with preference threshold delta". A line is structured as follows: the first element encodes the index of criterion g1 in <code>criteriaIDs</code> , the next element encodes the ID of criterion g2, and the last element contains the preference threshold delta.
<code>status</code>	Either OK if all the <code><criteriaComparisons></code> tags could be correctly read, or the description of the error.

Examples

```
tree <- xmlTreeParse(system.file("extdata", "testFile.xml", package="RXMCDa"), useInternalNodes = TRUE)
critIDs <- getCriteriaIDs(tree)
comps <- getCriteriaComparisons(tree, critIDs[[1]])
```

```
getCriteriaComparisonsLabels
```

Get comparisons of alternatives

Description

Extracts <criteriaComparisons> from an XML tree written according to the XMCDa standard and stores the IDs of the concerned criteria.

Usage

```
getCriteriaComparisonsLabels(tree, critIDs=NULL, mcdaConcept = NULL)
```

Arguments

tree	Object containing the XMCDa XML tree.
critIDs	A vector containing the IDs of the criteria to be considered for the extractions.
mcdaConcept	A string containing the specific mcdaConcept attribute which should be searched for.

Value

The function returns a list structured as follows:

-	The first elements contain the <alternativesComparisons> found in <tree> as matrices. These elements are named according to the mcdaConcept attribute if it can be found. Each line of each matrix corresponds to one constraint of the type "criterion g1 is preferred to criterion g2 with preference threshold delta". A line is structured as follows: the first element encodes the ID of criterion g1, the second element encodes the ID of criterion g2, and the last element contains the preference threshold delta.
status	Either OK if all the <criteriaComparisons> tags could be correctly read, or the description of the error.

Examples

```
tree <- xmlTreeParse(system.file("extdata", "testFile.xml", package="RXMCDa"), useInternalNodes=TRUE)
critIDs <- getCriteriaIDs(tree)
comps <- getCriteriaComparisonsLabels(tree, critIDs[[1]])
```

getCriteriaIDs	<i>Get IDs of criteria</i>
----------------	----------------------------

Description

Gets the IDs of criteria, stored in the <criteria> tag, from an XML tree written according to the XMCDa standard.

Usage

```
getCriteriaIDs(tree, mcdaConcept = NULL)
```

Arguments

tree	Object containing the XMCDa XML tree.
mcdaConcept	A string containing the specific mcdaConcept attribute which should be searched for.

Value

The function returns a list structured as follows:

-	The first elements contain vectors with the criteria's IDs which have been found in each <criteria> tag. These elements are named according to the mcdaConcept attribute if it can be found.
status	Either OK if all the <criteria> tags could be correctly read, or the description of the error.

Examples

```
tree = newXMLDoc()
newXMLNode("xmcd:XMCDa", namespace = c("xsi" = "http://www.w3.org/2001/XMLSchema-instance",
root<-getNodeSet(tree, "/xmcd:XMCDa")
criteria<-newXMLNode("criteria", parent=root[[1]], namespace=c())
newXMLNode("criterion", attrs = c(id="g1"), parent=criteria, namespace=c())
newXMLNode("criterion", attrs = c(id="g2"), parent=criteria, namespace=c())
y<-getNodeSet(tree, "//criteria")
x<-getCriteriaIDs(y[[1]])
```

```
getCriteriaIntervalValues
```

Get interval values related to criteria

Description

Extracts the number of criteria from an XML tree written according to the XMCDa standard.

Usage

```
getCriteriaIntervalValues(tree, criteriaIDs, mcdaConcept = NULL)
```

Arguments

<code>tree</code>	Object containing the XMCDa XML tree.
<code>criteriaIDs</code>	A vector containing the IDs of the criteria to be considered for the extractions.
<code>mcdaConcept</code>	A string containing the specific <code>mcdaConcept</code> attribute which should be searched for.

Value

The function returns a list structured as follows:

<code>-</code>	The first elements contain the number of criteria of each <code><criteria></code> found in <code><tree></code> . These elements are named according to the <code>mcdaConcept</code> attribute if it can be found.
<code>status</code>	Either OK if all the <code><criteria></code> tags could be correctly read, or the description of the error.

Examples

```
tree <- xmlTreeParse(system.file("extdata", "testFile.xml", package="RXMCDa"), useInternalNode
critIDs <- getCriteriaIDs(tree)
intVals <- getCriteriaIntervalValues(tree, critIDs[[1]])
```

```
getCriteriaPairsComparisons
```

Get comparisons of pairs of criteria

Description

Gets comparisons of pairs of criteria, stored in the <criteriaComparisons> tag (pairs are represented as sets of two elements), from an XML tree written according to the XMCDa standard.

Usage

```
getCriteriaPairsComparisons(tree, criteriaIDs, mcdaConcept = NULL)
```

Arguments

tree	Object containing the XMCDa XML tree.
criteriaIDs	A vector containing the IDs of the criteria to be considered for the extractions.
mcdaConcept	A string containing the specific mcdaConcept attribute which should be searched for.

Value

The function returns a list structured as follows:

-	The first elements contain the <criteriaComparisons> found in <tree> as matrices. These elements are named according to the mcdaConcept attribute if it can be found. Each line of each matrix corresponds to one constraint of the type "criterion g1 is preferred to criterion g2 with preference threshold delta". A line is structured as follows: the first element encodes the index of criterion g1 in criteriaIDs, the next element encodes the ID of criterion g2, and the last element contains the preference threshold delta.
status	Either OK if all the <criteriaComparisons> tags could be correctly read, or the description of the error.

Examples

```
tree <- xmlTreeParse(system.file("extdata", "testFile.xml", package="RXMCDa"), useInternalNodes = TRUE)
critIDs <- getCriteriaIDs(tree)
comps <- getCriteriaPairsComparisons(tree, critIDs[[1]])
```

```
getCriteriaPairsIntervalValues
```

Get interval values related to pairs of criteria

Description

Gets interval values related to pairs of criteria, stored in the <criteriaValues> tag (pairs are represented as sets of two elements), from an XML tree written according to the XMCDa standard.

Usage

```
getCriteriaPairsIntervalValues(tree, criteriaIDs, mcdaConcept = NULL)
```

Arguments

tree	Object containing the XMCDa XML tree.
criteriaIDs	A vector containing the IDs of the criteria to be considered for the extractions.
mcdaConcept	A string containing the specific mcdaConcept attribute which should be searched for.

Value

The function returns a list structured as follows:

-	The first elements contain the <criteriaValues> found in <tree> as matrices. These elements are named according to the mcdaConcept attribute if it can be found. Each line of each matrix corresponds to one constraint of the type "the interval of values assigned to the couple of criteria (g1,g2) is [x,y]". A line is structured as follows: the first element encodes the index of criterion g1 in criteriaIDs, the next element encodes the ID of criterion g2, and the last two elements contain x and y.
status	Either OK if all the <criteriaValues> tags could be correctly read, or the description of the error.

Examples

```
tree <- xmlTreeParse(system.file("extdata", "testFile.xml", package="RXMCDa"), useInternalNodes=TRUE)
critIDs <- getCriteriaIDs(tree)
pairsVals <- getCriteriaPairsIntervalValues(tree, critIDs[[1]], mcdaConcept="interactionInterval")
```

```
getCriteriaPairsValues
```

Get values related to pairs of criteria

Description

Gets values related to pairs of criteria, stored in the `<criteriaValues>` tag (pairs are represented as sets of two elements), from an XML tree written according to the XMCDa standard.

Usage

```
getCriteriaPairsValues(tree, criteriaIDs, mcdaConcept = NULL)
```

Arguments

<code>tree</code>	Object containing the XMCDa XML tree.
<code>criteriaIDs</code>	A vector containing the IDs of the criteria to be considered for the extractions.
<code>mcdaConcept</code>	A string containing the specific <code>mcdaConcept</code> attribute which should be searched for.

Value

The function returns a list structured as follows:

<code>-</code>	The first elements contain the <code><criteriaValues></code> found in <code><tree></code> as matrices. These elements are named according to the <code>mcdaConcept</code> attribute if it can be found. Each line of each matrix corresponds to one statement of the type "the value assigned to the couple of criteria (g1,g2) is x". A line is structured as follows: the first element encodes the index of criterion g1 in <code>criteriaIDs</code> , the next element encodes the ID of criterion g2, and the last elements contain x.
<code>status</code>	Either OK if all the <code><criteriaValues></code> tags could be correctly read, or the description of the error.

Examples

```
tree <- xmlTreeParse(system.file("extdata", "testFile.xml", package="RXMCDa"), useInternalNode
critIDs <- getCriteriaIDs(tree)
pairsVals <- getCriteriaPairsValues(tree, critIDs[[1]], mcdaConcept="interactionValues")
```

getCriteriaValues *Get values related to criteria*

Description

Gets values related to criteria, stored in the <criteriaValues> tag, from an XML tree written according to the XMCDa standard.

Usage

```
getCriteriaValues(tree, criteriaIDs, mcdaConcept = NULL)
```

Arguments

tree	Object containing the XMCDa XML tree.
criteriaIDs	A vector containing the IDs of the criteria to be considered for the extractions.
mcdaConcept	A string containing the specific mcdaConcept attribute which should be searched for.

Value

The function returns a list structured as follows:

-	The first elements contain matrices representing each an <criteriaValues>. Each line of each matrix corresponds to an element of the type "the value assigned to criterion g1 is x". A line is structured as follows: the first element encodes the index of the ID of criterion g1 in criteriaIDs, the second element encodes the value x. These elements are named according to the mcdaConcept attribute if it can be found.
status	Either OK if all the <criteriaValues> tags could be correctly read, or the description of the error.

Examples

```
tree <- xmlTreeParse(system.file("extdata", "testFile.xml", package="RXMCDa"), useInternalNodes = TRUE)
critIDs <- getCriteriaIDs(tree)
critVals <- getCriteriaValues(tree, critIDs[[1]])
```

```
getMobiusCapacities
```

Get Mobius capacities

Description

Gets Mobius capacities, stored in the `<criteriaValues>` tag, from an XML tree written according to the XMCD standard.

Usage

```
getMobiusCapacities(tree, criteriaIDs, numberOfCriteria, kadditivity, mcdaConcept =
```

Arguments

<code>tree</code>	Object containing the XMCD XML tree.
<code>criteriaIDs</code>	A vector containing the IDs of the criteria to be considered for the extractions.
<code>numberOfCriteria</code>	The total number of criteria to be considered.
<code>kadditivity</code>	The level of k-additivity to be considered.
<code>mcdaConcept</code>	A string containing the specific <code>mcdaConcept</code> attribute which should be searched for.

Value

The function returns a list structured as follows:

<code>-</code>	The first elements contain the <code><criteriaValues></code> found in <code><tree></code> as capacities (see <code>kappalab</code> package for details). These elements are named according to the <code>mcdaConcept</code> attribute if it can be found.
<code>status</code>	Either OK if all the <code><criteriaValues></code> tags could be correctly read, or the description of the error.

Examples

```
tree <- xmlTreeParse(system.file("extdata", "testFile.xml", package="RXMCD"), useInternalNode
critIDs <- getCriteriaIDs(tree)
capa <- getMobiusCapacities(tree, critIDs[[1]], 5, 5, mcdaConcept="mobiusCapacity")
```

```
getNumberOfAlternatives
    Get number of alternatives
```

Description

Extracts the number of alternatives from the <alternatives> tag from an XML tree written according to the XMCD standard.

Usage

```
getNumberOfAlternatives(tree, mcdaConcept = NULL)
```

Arguments

tree	Object containing the XMCD XML tree.
mcdaConcept	A string containing the specific mcdaConcept attribute which should be searched for.

Value

The function returns a list structured as follows:

-	The first elements contain the number of alternatives of each <alternatives> tag found in <tree>. These elements are named according to the mcdaConcept attribute if it can be found.
status	Either OK if all the <alternatives> tags could be correctly read, or the description of the error.

Examples

```
tree = newXMLDoc()
newXMLNode("xmcd:XMCD", namespace = c("xsi" = "http://www.w3.org/2001/XMLSchema-instance",
root<-getNodeSet(tree, "/xmcd:XMCD")
alternatives<-newXMLNode("alternatives", parent=root[[1]], namespace=c())
newXMLNode("alternative", attrs = c(id="x1"), parent=alternatives, namespace=c())
newXMLNode("alternative", attrs = c(id="x2"), parent=alternatives, namespace=c())
y<-getNodeSet(tree, "//alternatives")
x<-getNumberOfAlternatives(y[[1]])
```

```
getNumberOfCriteria
```

Get number of criteria

Description

Extracts the number of criteria from an XML tree written according to the XMCDa standard.

Usage

```
getNumberOfCriteria(tree, mcdaConcept = NULL)
```

Arguments

<code>tree</code>	Object containing the XMCDa XML tree.
<code>mcdaConcept</code>	A string containing the specific <code>mcdaConcept</code> attribute which should be searched for.

Value

The function returns a list structured as follows:

- The first elements contain the number of criteria of each `<criteria>` found in `<tree>`. These elements are named according to the `mcdaConcept` attribute if it can be found.
- `status` Either OK if all the `<criteria>` tags could be correctly read, or the description of the error.

Examples

```
tree = newXMLDoc()

newXMLNode("xmcd:XMCDa", namespace = c("xsi" = "http://www.w3.org/2001/XMLSchema-instance",

root<-getNodeSet(tree, "/xmcd:XMCDa")

criteria<-newXMLNode("criteria", parent=root[[1]], namespace=c())

criterion<-newXMLNode("criterion", attrs = c(id="g1"), parent=criteria, namespace=c())

y<-getNodeSet(tree, "//criteria")

x<-getNumberOfCriteria(y[[1]])
```

getNumericValue *Get a numeric value*

Description

Extracts a numeric value (integer, real or rational) from an XML tree written according to the XMCD standard and returns it as a real number.

Usage

```
getNumericValue(tree)
```

Arguments

tree Object containing the XMCD XML tree (in practice, only the part containing the <value> to be read).

Value

The function returns a float based on the numeric value read.

Examples

```
tree = newXMLDoc()

newXMLNode("xmcd:XMCD", namespace = c("xsi" = "http://www.w3.org/2001/XMLSchema-instance",

root<-getNodeSet(tree, "/xmcd:XMCD")

val<-newXMLNode("value", parent=root[[1]], namespace=c())

newXMLNode("real", 3.14, parent=val, namespace=c())

y<-getNodeSet(tree, "//value")

x<-getNumericValue(y)
```

getParameters *Get parameters*

Description

Gets the parameters defined in the <methodParameters> tag from an XML tree written according to the XMCD standard.

Usage

```
getParameters(tree, name = NULL)
```

Arguments

`tree` Object containing the XMCD A XML tree.
`name` A string containing the specific name attribute which should be searched for.

Value

The function returns a list structured as follows:

- The first elements contain the parameters which have been read in each `parameter` tag (either a `<label>`, `<real>` or `<integer>`). These elements are named according to the `name` attribute if it can be found.
- `status` Either OK if all the `<methodParameters>` tags could be correctly read, or the description of the error.

Examples

```
tree = newXMLDoc()
newXMLNode("xmcd a:XMCD A", namespace = c("xsi" = "http://www.w3.org/2001/XMLSchema-instance",
root<-getNodeSet(tree, "/xmcd a:XMCD A")
parameters<-newXMLNode("methodParameters", parent=root[[1]], namespace=c())
parameter <- newXMLNode("parameter", attrs = c(name="numIt"), parent=parameters, namespace=c()
value <- newXMLNode("value", parent=parameter, namespace=c())
newXMLNode("integer", value=3, parent=value, namespace=c())
y<-getNodeSet(tree, "//methodParameters")
x<-getParameters(y[[1]])
```

```
getPerformanceTables
```

Get performance tables

Description

Extracts the performance tables stored in the `<performanceTable` tags from an XML tree written according to the XMCD A standard.

Usage

```
getPerformanceTables(tree, altIDs = NULL, critIDs = NULL, mcdaConcept = NULL)
```

Arguments

<code>tree</code>	Object containing the XMCDa XML tree.
<code>altIDs</code>	A vector containing the IDs of the alternatives to be considered for the extractions.
<code>critIDs</code>	A vector containing the IDs of the criteria to be considered for the extractions.
<code>mcdaConcept</code>	A string containing the specific <code>mcdaConcept</code> attribute which should be searched for.

Value

The function returns a list structured as follows:

-	The first elements contain the performance tables found in each <code><performanceTable></code> of <code><tree></code> as matrices. The rows of each matrix are labelled according to the alternatives' IDs and the columns according to the criteria IDs. These elements are named according to the <code>mcdaConcept</code> attribute if it can be found.
<code>status</code>	Either OK if all the <code><performanceTable></code> tags could be correctly read, or the description of the error.

Examples

```
tree <- xmlTreeParse(system.file("extdata", "testFile.xml", package="RXMCDA"), useInternalNodeNames = TRUE)
tables <- getPerformanceTables(tree)
```

```
putAlternativesComparisonsLabels
      Put comparisons of alternatives
```

Description

Writes `<alternativesComparisons>` in an XML tree written according to the XMCDa standard.

Usage

```
putAlternativesComparisonsLabels(tree, alternativesComparisons, mcdaConcept = NULL)
```

Arguments

tree	Object containing the XMCDa XML tree.
alternativesComparisons	A matrix containing the <alternativesComparisons> as a matrix. Each line of each matrix corresponds to one constraint of the type "alternative a is preferred to alternative b with preference threshold delta". A line is structured as follows: the first element encodes the ID of alternative a, the second element encodes the ID of alternative b, and the last element contains a valuation.
mcdaConcept	A string containing the specific mcdaConcept attribute which should be written.

Value

The function returns a list structured as follows:

status	Either OK if all the <alternativesComparisons> tags could be correctly put, or the description of the error.
--------	--

Examples

```
tree = newXMLDoc()
newXMLNode("xmcd:XMCDa", namespace = c("xsi" = "http://www.w3.org/2001/XMLSchema-instance",
comps <- rbind(c("x", "y", "0.07"), c("y", "z", "0.01"))
altIDs <- c("x","y","z")
putAlternativesComparisonsLabels(tree,comps, mcdaConcept="newComparisons")
```

```
putAlternativesPlot
```

Put plot in base64 related to alternatives

Description

Puts a plot coded in base64 as an alternativeValue tag in an XML tree written according to the XMCDa standard.

Usage

```
putAlternativesPlot(tree, base64plot, alternativesIDs, mcdaConcept=NULL, name=NULL)
```

Arguments

<code>tree</code>	Object containing the XMCCA XML tree.
<code>base64plot</code>	String containing the base64 encoding of the plot.
<code>alternativesIDs</code>	A vector of alternatives' IDs.
<code>mcdaConcept</code>	A string containing the specific <code>mcdaConcept</code> attribute which should be written.
<code>name</code>	A string containing the specific <code>name</code> attribute which should be written.

Value

The function returns a list structured as follows:

<code>status</code>	Either OK if the <code><alternativeValue></code> tag could be correctly put, or the description of the error.
---------------------	---

`putAlternativesValues`

Put values related to alternatives

Description

Puts values related to alternatives as an `alternativesValues` tag in an XML tree written according to the XMCCA standard.

Usage

```
putAlternativesValues(tree, alternativesValues, alternativesIDs, mcdaConcept = NULL)
```

Arguments

<code>tree</code>	Object containing the XMCCA XML tree.
<code>alternativesValues</code>	A matrix containing the values of the alternatives which have to be stored. Each line of this matrix represents a statement of the form "alternative x has value y". The first element of each line stores the index of the alternative x in <code>alternativesIDs</code> , the second element stores the value y.
<code>alternativesIDs</code>	A vector containing the alternatives' IDs.
<code>mcdaConcept</code>	A string containing the specific <code>mcdaConcept</code> attribute which should be written.

Value

The function returns a list structured as follows:

`status` Either OK if all the `<alternativesValues>` tags could be correctly put, or the description of the error.

Examples

```
altIDs <- c("x", "y", "z")
altVals <- rbind(c(1,1), c(2,0.5), c(3,0.2))
tree = newXMLDoc()
newXMLNode("xmcd:XMCD", namespace = c("xsi" = "http://www.w3.org/2001/XMLSchema-instance",
putAlternativesValues(tree, altVals, altIDs)
altVals2 <- getAlternativesValues(tree, altIDs)
```

```
putAlternativeValue
```

Put a value related to an alternative

Description

Puts a value related to an alternative (or a set of alternatives) as a `alternativeValue` tag in an XML tree written according to the XMCD standard.

Usage

```
putAlternativeValue(tree, alternativeValue, alternativesIDs = NULL, mcdaConcept = M
```

Arguments

`tree` Object containing the XMCD XML tree.

`alternativeValue` A float representing the value assigned to the alternative.

`alternativesIDs` A string representing the alternative's ID or a vector representing a set of alternatives IDs.

`mcdaConcept` A string containing the specific `mcdaConcept` attribute which should be written.

Value

The function returns a list structured as follows:

`status` Either OK if the `<alternativeValue>` tag could be correctly put, or the description of the error.

Examples

```
altID <- c("x")
tree = newXMLDoc()
newXMLNode("xmcd:XMCDA", namespace = c("xsi" = "http://www.w3.org/2001/XMLSchema-instance",
putAlternativeValue(tree, 0.8, alternativesIDs = altID)
```

<code>putCapacity</code>	<i>Put a capacity</i>
--------------------------	-----------------------

Description

Puts values related to a capacity as a `criteriaValues` tag in an XML tree written according to the XMCDA standard.

Usage

```
putCapacity(tree, capacity, criteriaIDs, mcdaConcept = NULL)
```

Arguments

`tree` Object containing the XMCDA XML tree.

`capacity` An object of type `capacity` (see `kappalab` package).

`criteriaIDs` A vector containing the criteria's IDs.

`mcdaConcept` A string containing the specific `mcdaConcept` attribute which should be written.

Value

The function returns a list structured as follows:

`status` Either OK if all the `<criteriaValues>` tags could be correctly put, or the description of the error.

Examples

```

library(kappalab)

tree = newXMLDoc()

newXMLNode("xmcd:XMCDA", namespace = c("xsi" = "http://www.w3.org/2001/XMLSchema-instance",

mu<-capacity(0:15)

a <- Mobius(mu)

critIDs <- c("g1","g2","g3","g4")

putCapacity(tree, a, critIDs, mcdaConcept="capacity")

```

```
putCriteriaPairsValues
```

Put values related to pairs of criteria

Description

Puts values related to pairs of criteria as a `criteriaValues` tag in an XML tree written according to the XMCDA standard.

Usage

```
putCriteriaPairsValues(tree, criteriaPairsValues, criteriaIDs, mcdaConcept = NULL)
```

Arguments

<code>tree</code>	Object containing the XMCDA XML tree.
<code>criteriaPairsValues</code>	A as matrix representing the values assigned to pairs of criteria. Each line of the matrix corresponds to one statement of the type "the value assigned to the couple of criteria (g1,g2) is x". A line is structured as follows: the first element encodes the index of criterion g1 in <code>criteriaIDs</code> , the next element encodes the index of the ID of criterion g2, and the last elements contain x.
<code>criteriaIDs</code>	A vector containing the criteria's IDs.
<code>mcdaConcept</code>	A string containing the specific <code>mcdaConcept</code> attribute which should be written.

Value

The function returns a list structured as follows:

<code>status</code>	Either OK if all the <code><criteriaValues></code> tags could be correctly put, or the description of the error.
---------------------	--

Examples

```

tree = newXMLDoc()

newXMLNode("xmcd:XMCD", namespace = c("xsi" = "http://www.w3.org/2001/XMLSchema-instance",
critIDs <- c("g1", "g2", "g3", "g4")

pairsVals <- rbind(c(1,2,0.17), c(2,3,0.5), c(3,4,0.16))

putCriteriaPairsValues(tree, pairsVals, critIDs)

```

putCriteriaPlot *Put a plot in base64 related to criteria*

Description

Puts a plot coded in base64 as a `criterionValue` tag in an XML tree written according to the XMCD standard.

Usage

```
putCriteriaPlot(tree, base64plot, criteriaIDs, mcdaConcept=NULL, name=NULL)
```

Arguments

<code>tree</code>	Object containing the XMCD XML tree.
<code>base64plot</code>	String containing the base64 encoding of the plot.
<code>criteriaIDs</code>	A vector of criteria's IDs.
<code>mcdaConcept</code>	A string containing the specific <code>mcdaConcept</code> attribute which should be written.
<code>name</code>	A string containing the specific <code>name</code> attribute which should be written.

Value

The function returns a list structured as follows:

<code>status</code>	Either OK if the <code><criterionValue></code> tag could be correctly put, or the description of the error.
---------------------	---

putCriteriaValues *Put values related to criteria*

Description

Puts values related to criteria as a `criteriaValues` tag in an XML tree written according to the XMCDa standard.

Usage

```
putCriteriaValues(tree, criteriaValues, criteriaIDs, mcdaConcept = NULL)
```

Arguments

<code>tree</code>	Object containing the XMCDa XML tree.
<code>criteriaValues</code>	A matrix representing a <code><criteriaValues></code> . Each line of each matrix corresponds to an element of the type "the value assigned to criterion g1 is x". A line is structured as follows: the first element encodes the index of the ID of criterion g1 in <code>criteriaIDs</code> , the second element encodes the value x. These elements are named according to the <code>mcdaConcept</code> attribute if it can be found.
<code>criteriaIDs</code>	A vector containing the criteria's IDs.
<code>mcdaConcept</code>	A string containing the specific <code>mcdaConcept</code> attribute which should be written.

Value

The function returns a list structured as follows:

<code>status</code>	Either OK if all the <code><criteriaValues></code> tags could be correctly put, or the description of the error.
---------------------	--

Examples

```
critIDs <- c("x", "y", "z")  
  
critVals <- rbind(c(1,1), c(2,0.5), c(3,0.2))  
  
tree = newXMLDoc()  
  
newXMLNode("xmcd:XMCDA", namespace = c("xsi" = "http://www.w3.org/2001/XMLSchema-instance"),  
putCriteriaValues(tree, critVals, critIDs)
```

putCriterionValue *Put a value related to a criterion*

Description

Puts a value related to a criterion (or a set of criteria) as a `criterionValue` tag in an XML tree written according to the XMCDa standard.

Usage

```
putCriterionValue(tree, criterionValue, criteriaIDs = NULL, mcdaConcept = NULL)
```

Arguments

<code>tree</code>	Object containing the XMCDa XML tree.
<code>criterionValue</code>	A float representing the value assigned to the criterion.
<code>criteriaIDs</code>	A string representing the criterion's ID or a vector representing a set of criteria IDs.
<code>mcdaConcept</code>	A string containing the specific <code>mcdaConcept</code> attribute which should be written.

Value

The function returns a list structured as follows:

<code>status</code>	Either OK if all the <code><criterionValue></code> tags could be correctly put, or the description of the error.
---------------------	--

Examples

```
critID <- c("x")

tree = newXMLDoc()

newXMLNode("xmcd:XMCDa", namespace = c("xsi" = "http://www.w3.org/2001/XMLSchema-instance",
putCriterionValue(tree,0.8,criteriaIDs = critID)
```

putErrorMessage *Put an error message*

Description

Puts an error message in a <methodMessages> tag of an XML tree written according to the XMCD standard.

Usage

```
putErrorMessage(tree, errorMessage, name = NULL)
```

Arguments

tree Object containing the XMCD XML tree.
errorMessage A string containing the text of the error message.
name A string containing the specific name attribute which should be searched for.

Value

The function returns a list structured as follows:

status Either OK if the <errorMessage> tag could be correctly put, or the description of the error.

putLogMessage *Put a log message*

Description

Puts an log message in a <methodMessages> tag of an XML tree written according to the XMCD standard.

Usage

```
putLogMessage(tree, logMessage, name = NULL)
```

Arguments

tree Object containing the XMCD XML tree.
logMessage A string containing the text of the log message.
name A string containing the specific name attribute which should be searched for.

Value

The function returns a list structured as follows:

status	Either OK if the <logMessage> tag could be correctly put, or the description of the error.
--------	--

putMessage	<i>Put a message</i>
------------	----------------------

Description

Puts an message in a <methodMessages> tag of an XML tree written according to the XMCDA standard.

Usage

```
putMessage(tree, message, name = NULL)
```

Arguments

tree	Object containing the XMCDA XML tree.
message	A string containing the text of the log message.
name	A string containing the specific name attribute which should be put.

Value

The function returns a list structured as follows:

status	Either OK if the <message> tag could be correctly put, or the description of the error.
--------	---

putPerformanceTable	<i>Put a performance table in the XMCDA tree</i>
---------------------	--

Description

Puts a performance table (performanceTable tag) in an XML tree written according to the XMCDA standard.

Usage

```
putPerformanceTable(tree, performanceTable, mcdaConcept = NULL)
```

Arguments

tree	Object containing the XMCDAs XML tree.
performanceTable	A matrix representing the performance table (the lines are named according to the alternatives' IDs, the columns are named according to the criteria IDs).
mcdaConcept	A string containing the specific mcdaConcept attribute which should be written.

Value

The function returns a list structured as follows:

status	Either OK if the <performanceTable> tag could be correctly put, or the description of the error.
--------	--

Examples

```
performanceTable <- rbind(c(1,2,3),c(4,5,6))
rownames(performanceTable) <- c("x","y")
colnames(performanceTable) <- c("g1","g2","g3")

tree = newXMLDoc()

newXMLNode("xmcdas:XMCDAs", namespace = c("xsi" = "http://www.w3.org/2001/XMLSchema-instance",
putPerformanceTable(tree,performanceTable)
```

```
putPointsCriterionFunction
```

Put value functions defined by sets of points

Description

Puts value functions defined by sets of points in a criterionFunction tag under the criterion tag in an XML tree written according to the XMCDAs standard.

Usage

```
putPointsCriterionFunction(tree, points, mcdaConcept = NULL)
```

Arguments

<code>tree</code>	Object containing the XMCDAs XML tree.
<code>points</code>	A list, where each element is named by the ID of a criterion, and contains a matrix representing the points (each line is a point, the first column represents the abscissa, the second the ordinate).
<code>mcdaConcept</code>	A string containing the specific <code>mcdaConcept</code> attribute which should be written.

Value

The function returns a list structured as follows:

<code>status</code>	Either OK if all the <code><criterionFunction></code> tags could be correctly put, or the description of the error.
---------------------	---

Examples

```
tree = newXMLDoc()

newXMLNode("xmcdasXMCDAs", namespace = c("xsi" = "http://www.w3.org/2001/XMLSchema-instance",

x<-list()
x<-c(x, list(g1=rbind(c(1,2), c(3,4))))
x<-c(x, list(g2=rbind(c(5,6), c(7,8), c(9,10))))
x<-c(x, list(g3=rbind(c(11,12))))
x<-c(x, list(g4=rbind(c(13,14), c(15,16))))

putPointsCriterionFunction(tree, x)
```

Index

*Topic **methods**

checkXSD, 1
getAlternativesComparisons, 2
getAlternativesComparisonsLabels, 4
getAlternativesIDs, 5
getAlternativesIntervalValues, 6
getAlternativesValues, 7
getCriteriaComparisons, 8
getCriteriaComparisonsLabels, 9
getCriteriaIDs, 10
getCriteriaIntervalValues, 11
getCriteriaPairsComparisons, 12
getCriteriaPairsIntervalValues, 13
getCriteriaPairsValues, 14
getCriteriaValues, 15
getMobiusCapacities, 16
getNumberOfAlternatives, 17
getNumberOfCriteria, 18
getNumericValue, 19
getParameters, 19
getPerformanceTables, 20
putAlternativesComparisonsLabels, 21
putAlternativesPlot, 22
putAlternativesValues, 23
putAlternativeValue, 24
putCapacity, 25
putCriteriaPairsValues, 26
putCriteriaPlot, 27
putCriteriaValues, 28
putCriterionValue, 29
putErrorMessage, 30
putLogMessage, 30
putMessage, 31
putPerformanceTable, 31
putPointsCriterionFunction, 32
checkXSD, 1
getAlternativesComparisons, 2
getAlternativesComparisonsLabels, 4
getAlternativesIDs, 5
getAlternativesIntervalValues, 6
getAlternativesValues, 7
getCriteriaComparisons, 8
getCriteriaComparisonsLabels, 9
getCriteriaIDs, 10
getCriteriaIntervalValues, 11
getCriteriaPairsComparisons, 12
getCriteriaPairsIntervalValues, 13
getCriteriaPairsValues, 14
getCriteriaValues, 15
getMobiusCapacities, 16
getNumberOfAlternatives, 17
getNumberOfCriteria, 18
getNumericValue, 19
getParameters, 19
getPerformanceTables, 20
putAlternativesComparisonsLabels, 21
putAlternativesPlot, 22
putAlternativesValues, 23
putAlternativeValue, 24
putCapacity, 25
putCriteriaPairsValues, 26
putCriteriaPlot, 27
putCriteriaValues, 28
putCriterionValue, 29
putErrorMessage, 30
putLogMessage, 30

putMessage, [31](#)
putPerformanceTable, [31](#)
putPointsCriterionFunction, [32](#)