

```
//+-----  
//+ Overview:  
//+ *Part 1: XTM - data model  
//+ *Part 2: Object summaries  
//+ *Part 3: TMCL - data model  
//+-----
```

```
//+-----  
//+ Part 1: XTM - data model:  
//+ The first part describes the xtm's data model, here will be all elements  
//+ defined in the xtm defined as json objects and finally there will be used  
//+ as json objects in a json-fragment-object.  
//+  
//+ this json model depends on the xtm version 2.0 and contains the following  
//+ objects:  
//+ *resourceData  
//+ *variant  
//+ *name  
//+ *name  
//+ *occurrence  
//+ *topic  
//+ *role  
//+ *association  
//+ *topicStub  
//+ *fragment  
//+  
//+ At the end of this file are some example json objects, you can also  
//+ validate json data on "http://www.jsonlint.com/".  
//+ Note all values, although they are null values e.g. the "type" field in  
//+ a name object should be set to a value - in this case "null".  
//+-----
```

```
//+-----  
//+ resourceData  
//+-----  
{  
  "datatype" : "Text",  
  "value" : "Text"  
}
```

```
//+-----  
//+ variant  
//+-----  
{  
  "itemIdentities" : [ "Text" , "..." ],  
  "scopes" : [ [ "PSI-1-t1", "PSI-2-t1", "..." ], [ "PSI-1-t2", "PSI-2-t2",  
  "..." ], [ "..." ] ],  
  "resourceRef" : "Text",  
  "resourceData" : { <resourceData> }  
}
```

```

//+-----
//+ name
//+-----
{
  "itemIdentities" : [ "Text", "..." ],
  "type" : [ "PSI-1", "PSI-2", "..." ],
  "scopes" : [ [ "PSI-1-t1", "PSI-2-t1", "..." ], [ "PSI-1-t2", "PSI-2-t2",
"..." ], [ "..." ] ],
  "value" : "Text",
  "variants" : [ {<variant>}, { <...> } ]
}

//+-----
//+ occurrence
//+-----
{
  "itemIdentities" : [ "Text", "..." ],
  "type" : [ "PSI-1", "PSI-2", "..." ],
  "scopes" : [ [ "PSI-1-t1", "PSI-2-t1", "..." ], [ "PSI-1-t2", "PSI-2-t2",
"..." ], [ "..." ] ],
  "resourceRef" : "Text",
  "resourceData" : { <resourceData> }
}

//+-----
//+ topic
//+-----
{
  "id" : "Text",
  "itemIdentities" : [ "Text", "..." ],
  "subjectLocators" : [ "Text", "..." ],
  "subjectIdentifiers" : [ "Text", "..." ],
  "instanceOfs" : [ [ "PSI-1-t1", "PSI-2-t1", "..." ], [ "PSI-1-t2", "PSI-2-t2",
"..." ], [ "..." ] ],
  "names" : [ { <name> }, { <...> } ],
  "occurrences" : [ { <occurrence> }, { <...> } ]
}

//+-----
//+ role
//+-----
{
  "itemIdentities" : [ "Text", "..." ],
  "type" : [ "PSI-1", "PSI-2", "..." ],
  "topicRef" : [ "PSI-1", "PSI-2", "..." ]
}

//+-----
//+ association
//+-----
{
  "itemIdentities" : [ "Text", "..." ],
  "type" : [ "PSI-1", "PSI-2", "..." ],
  "scopes" : [ [ "PSI-1-t1", "PSI-2-t1", "..." ], [ "PSI-1-t2", "PSI-2-t2",
"..." ], [ "..." ] ],
  "roles" : [ { <role> }, { <...> } ]
}

```

```

//+-----
//+ topicStub
//+-----
{
  "id" : "Text",
  "itemIdentities" : [ "Text", "..." ],
  "subjectLocators" : [ "Text", "..." ],
  "subjectIdentifiers" : [ "Text", "..." ]
}

//+-----
//+ fragment
//+   The field tmIds should have only one tm-id in the list, because
//+   there will be used only the first, if the fragment is an incoming one
//+   outgoing fragments have a list with more tmIds but at least one
//+-----
{
  "topic" : { <topic> },
  "topicStubs" : [ { <topicStub> }, { <...> } ],
  "associations" : [ { <association> }, { <...> } ],
  "tmIds" : [ "id-1", "id-2", "..." ]
}

```

```

//+-----
//+ Part 2: Object summaries
//+ The second part contains object summaries of exisiting objects in
//+ isidorus.
//+
//+ *psiSummary
//+ *topicSummary
//+-----

//+-----
//+ psiSummary
//+ The json list is made of inner json-lists.
//+ Every inner json list represents one topic with all psis owned by the
//+ topic. The outer list represents a set of all topics exist in isidorus.
//+-----
[ [ "topic-1-psi-1", "topic-1-psi-2", <...> ], [ "topic-2-psi-1", "topic-2-
psi-2", <...> ], <...> ]

//+-----
//+ topicSummary
//+ Contains the topic id,subjetcIdentifiers, itemIdentities,
//+ subjectLocators, nameSummaries and occurrenceSummaries
//+-----
{
  "id" : "Text",
  "itemIdentities" : [ "Text", "..." ],
  "subjectLocators" : [ "Text", "..." ],
  "subjectIdentifiers" : [ "Text", "..." ],
  "instanceOfs" : [ [ "PSI-1-t1", "PSI-2-t1", "..." ], [ "PSI-1-t2", "PSI-2-t2",
  "..." ], [ "..." ] ],
  "names" : [ "name-1", "name-2", <...> ],
  "occurrences" : [ "occurrence-1", "occurrence-2", <...> ]
}

```

```

//+-----
//+ Part 3: TMCL - data model
//+ This part explains how the tmcl-rules/suggestions of isidorus are
//+ sent via HTTP as JSON-data.
//+-----

//+-----
//+ exclusiveInstances
//+ This message constains a list of topics represented as a list of topic
//+ psis which are exclusive instances for the owner.
//+-----
{
  "owner" : ["psi-1", "..."],
  "exclusives" : [ [ "topic-1-psi-1", "topic-2-psi-2", "..." ], [ "topic-2-psi",
"..." ], <...> ]
}

//+-----
//+ simpleConstraint
//+ This object contains a regexp member with the regular expression of the
//+ constraint, a cardMin member with the minimum cardinality of
//+ the referenced element which is represented as an unsignedInt
//+ and a cardMax member which describes the maximum cardinality of this
//+ element, this member contains an unsigendInt or the string
//+ "MAX_INT".
//+-----
{
  "regexp" : "regular expression",
  "cardMin" : "unsigned integer in string representation",
  "cardMax" : "unsigend integer in string representation or the string MAX_INT"
}

//+-----
//+ subjectIdentifierConstraint
//+ This object contains a regexp member with the regular expression of the
//+ subjectIdentifier, a cardMin member with the minimum cardinality of
//+ this subjectIdentifier in a topic which is represented as an unsignedInt
//+ and a cardMax member which describes the maximum cardinality of this
//+ subjectIdentifier, this member contains an unsigendInt or the string
//+ "MAX_INT".
//+-----
<simpleConstraint>

//+-----
//+ subjectLocatorConstraint
//+ This object contains a regexp member with the regular expression of the
//+ subjectLocator, a cardMin member with the minimum cardinality of
//+ this subjectLocator in a topic which is represented as an unsignedInt
//+ and a cardMax member which describes the maximum cardinality of this
//+ subjectLocator, this member contains an unsigendInt or the string
//+ "MAX_INT".
//+-----
<simpleConstraint>

```

```

//+-----
//+ scopeConstraint
//+   The scopeConstraint-Object contains a list of all available scopes of
//+   for an association/name/occurrence element - this depends where this
//+   json-object is contained.
//+   The member availableScopeTypes contains a list of lists of topics in
//+   form of psi-lists.
//+   cardMin defines the minimum number of all scopes of the parent element
//+   (association/name/occurrence).
//+   cardMax defines the maximum number of all scopes of the parent element.
//+-----
{
  "scopeTypes" : [ [ [ "psi-1-1", "psi-1-2", "..." ], [ "subtype-psi-1",
"..." ], <...> ], [ "psi-2-1" "..."], <...> ],
  "cardMin" : "unsigned integer in string representation",
  "cardMax" : "unsigend integer in string representation or the string MAX_INT"
}

//+-----
//+ topicNameConstraint
//+   nametypescope constains the original nametype and all valid subtypes
//+   with the specific scope constraints.
//+   constraints contains the constraints for the owner topic.
//+   Note scopeConstraints is a list, because, the can be more scope
//+   constriants, e.g. one constraints wants 2 scopes of the type "en", and
//+   "de", and another, wants 3 scopes of the type "fr", "pl" and "sp".
//+-----
{
  "nametypescopes" : [ {
    "nameType" : [psi-1, psi-2, "..." ],
    "scopeConstraints" : [ <scopeConstraints> ]
  },
  {
    "nameType" : [subtype-1-psi-1, subtype-1-psi-2,
"..." ],
    "scopeConstraints" : [ <scopeConstraints> ]
  },
  <...>
]
  "constraints" : [ <simpleConstraint>, < ... > ]
}

//+-----
//+ uniqueOccurrenceConstraint
//+   This object owns a regexp member with the regular expression of the
//+   occurrence which should be unique. So only occurrences that match the
//+   occurrenceType and the regexp will be checked.
//+   occurrenceType represents the topicType of the occurrence.
//+   cardMin describes the minimum number of all matched occurrences within
//+   all instances of this topicType's instances.
//+   cardMax describes the maximum number of all matched occurrences within
//+   all instances of this topicType's instances.
//+-----
<simpleConstraint>

```

```

//+-----
//+ topicOccurrenceConstraint
//+   occurrenceTypes contains a list of a json-sub-object. This sub-object
//+   contains an occurrenceType a specific list of scopeConstraints for
//+   the occurrenceType and a specific datatypeConstraint which contains
//+   the datatype for the occurrenceType.
//+   The entire list of occurrenceTypes contains the not only the
//+   original occurrenceType but also the subtypes of this occurrenceType.
//+   constraints is a constraint list of depending to the owner topic.
//+   uniqueConstraint is a list of uniqueConstraints which also depends on
//+   the owner topic.
//+-----
{
  "occurrenceTypes" : [ {
    "occurrenceType" : [ "psi-1", "psi-2", "..." ],
    "scopeConstraints" : [ <scopeConstraints> ],
    "datatypeConstraint" : "datatype"
  },
  {
    "occurrenceType" : [ "subtype-1-psi-1", "subtype-1-
psi-2", "..." ],
    "scopeConstraints" : [ <scopeConstraints> ],
    "datatypeConstraint" : "datatype"
  },
  <...>
  ],
  "constraints" : [ <simpleConstraints>, <...> ],
  "uniqueConstraint" : [ <uniqueConstraints>, <...> ]
}

//+-----
//+ associationRoleConstraint
//+   This object defines a list of psis of the roletype topic of which
//+   the role is an instance of.
//+   cardMin and cardMax defines the number of roles with the defined roletype
//+   in an association of a certain associationtype (the objects owner).
//+-----
{
  "roleType" : [ [ "topic-psi-1", "topic-psi-2", "..." ], ["subtype-1-psi-1",
"..."], <...> ],
  "cardMin" : "unsigned integer in string representation",
  "cardMax" : "unsigend integer in string representation or the string MAX_INT"
}

```

```

//+-----
//+ rolePlayerConstraint
//+   Defines the player of a certain role with a given type in an association
//+   of a given type.
//+   palyers is the psi-list representation of a list of all available
//+   players.
//+   roleTypes is a list of topics represented by a list of psi-lists.
//+   cardMin and cardMax defines the number of times the topicType (= player)
//+   can be the player in a role of a given type (= roleTypes) in an
//+   association of a given type (= objects owner).
//+-----
{
  "players" : [ [ "topic-psi-1", "topic-psi-2", "..." ], [ "topic-2-psi-1",
"...", <...> ]
  "roleTypes" : [ [ "topic-psi-1", "topic-psi-2", "..." ], [ "subtype-psi-1",
"..." ], <...> ],
  "cardMin" : "unsigned integer in string representation",
  "cardMax" : "unsigend integer in string representation or the string MAX_INT"
}

//+-----
//+ otherRoleConstraint
//+   This JSON-Object defines the number and types of (other-) roles in an
//+   association of a given type with a role of a give type.
//+   roleType is a allowed role with the player topicType.
//+   otherRoleType is the second role with the player otherTopicType.
//+   The values cardMin and cardMax defines the cardinality of otherRoleType.
//+-----
{
  "playerType" : [ "topic-psi-1", "topic-psi-2", "..." ],
  "roleType" : [ "topic-psi-1", "topic-psi-2", "..." ],
  "otherRoleType" : [ "topic-psi-1", "topic-psi-2", "..." ],
  "otherPlayerType" : [ "topic-psi-1", "topic-psi-2", "..." ],
  "cardMin" : "unsigned integer in string representation",
  "cardMax" : "unsigend integer in string representation or the string MAX_INT"
}

//+-----
//+ associationConstraint
//+   The associationConstraint describes how an association of a given type
//+   has to be defined.
//+   associationRoleTypeConstraint constains all available roletypes for this
//+   association.
//+   rolePlayerConstraint constains all players for certain roles of a given
//+   type.
//+   associationTypeScopes contains all available scopes for this association.
//+-----
{
  "associationType" : [ "topic-psi-1", "topic-psi-2" ],
  "associationRoleConstraints" : [ <associationRoleConstraint>, <...> ],
  "rolePlayerConstraints" : [ <rolePlayerConstraints>, <...> ],
  "otherRoleConstraints" : [ <otherRoleConstraint>, <...> ],
  "scopeConstraints" : { <scopeConstraint> }
}

```



```

//+-----
//+ topicConstraint
//+   The topicConstraint contains the members:
//+   *exclusiveInstances which contains a topic-list of topic-psis depending
//+   on the users exclusive-instance-constraints
//+   *subjectIdentifierConstraints which defines the subjectIdentifiers
//+   *subjectLocatorConstraints which defines the subjectLocators
//+   *topicNameConstraints which defines the topic names
//+   *topicOccurrenceConstraints which defines the topic occurrences
//+   *uniqueOccurrenceConstraints which defines the uniqueness of topic
//+   occurrences
//+-----
{
  "exclusiveInstances" : <exclusiveInstances>,
  "subjectIdentifierConstraints" : [ <subjectIdentifierConstraint>, <...> ],
  "subjectLocatorConstraints" : [ <subjectLocatorConstraint>, <...> ],
  "topicNameConstraints" : [ <topicNameConstraint>, <...> ],
  "topicOccurrenceConstraints" : [ <topicOccurrenceConstraint>, <...> ],
  "abstractConstraint" : <boolean>
}

//+-----
//+ fragmentConstraint
//+   This JSON-Object contains all constraints necessary for an entire
//+   fragment.
//+   topicConstraint contains an object with all constraints of all baseTypes
//+   of the mainTopic.
//+   associationConstraints contains a list of all association constraints
//+   depending on all baseTypes of the main topic.
//+-----
{
  "topicConstraints" : <topicConstraint>,
  "associationsConstraints" : [ <associationConstraint>, <...> ]
}

```