



The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement no. 284552 "CRISMA"

Deliverable No.			
Subproject No.	SP3	Subproject Title	Integrated Crisis Modelling System
Work package No.	WP34	Work package Title	ICMS Framework Building Block Implementation
Authors		Pascal Dihé (CIS)	
Status (F = Final; D = Draft)		D	
File Name		Core Control and Communication Information Models v1.5.doc	
Dissemination level (PU = Public; RE = Restricted; CO = Confidential)		CO	

Contact	Anna-Mari.Heikkila@vtt.fi Crisma.Coordinator@vtt.fi
Project	www.crismaproject.eu

Keywords	
Deliverable leader	Name: Pascal Dihé Partner: CIS Contact: pascal.dihe@cismet.de
Contractual Delivery date to the EC	
Actual Delivery date to the EC	

Document Revision History

Disclaimer

The content of the publication herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the documents is believed to be accurate, the authors(s) or any other participant in the CRISMA consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the CRISMA Consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the CRISMA Consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

Table of Contents

1. PURPOSE OF THIS DOCUMENT	6
1.1.1. Changes to previous version of the CCIM	6
2. CLASSES OF THE CORE CONTROL AND COMMUNICATION INFORMATION MODEL ..	7
2.1.1. Relation to conceptual business logic.....	9
3. TECHNICAL REALISATION.....	11
4. GENERIC EXTENSION MECHANISMS.....	13
4.1.1.1. Extension by inheritance	14
4.1.1.2. Extension Points.....	14
5. JSON ENCODING.....	17
5.1. Categories	17
5.2. Classifications.....	18
5.3. DataDescriptors	19
5.4. DataItems	20
5.5. InitialisationDescriptors	23
5.6. ManipulationDescriptors	25
5.7. RenderingDescriptors	26
5.8. SimulationDescriptors	28
5.9. Transitions	30
5.10. Worldstates.....	32
6. MODEL DOCUMENTATION	35
Core Control and Communication Information Models	35
BaseEntity	42
Categories.....	43
Classifications	44
DataDescriptors.....	45
DataItems	46
ExternalResource	48
GenericReference	48
IccFunctionDescriptors	49
InitialisationDescriptors	50
ManipulationDescriptors	51
RenderingDescriptors	52
SimulationDescriptors	53
Transitions.....	54
Worldstates	55

List of Figures

Figure 1: Core CCIM Classes and Relationships	7
Figure 2: Scenario expressed with help of the Core CCIM	10
Figure 3: Mapping the Core CCIM to the Conceptual Business Logic (detailed)	10
Figure 4: Core CCIM Database Model (excerpt)	11
Figure 5: Mapping UML Models to REST Models.....	12
Figure 6: DataItems Instance JSON.....	12
Figure 7: Core CCIM with extension points for Application CCIMs	13
Figure 8: Extension by inheritance	14
Figure 9: Extension Points of the Core CCIM.....	15
Figure 10: Categories JSON EntityInfo	17
Figure 11: Classifications JSON EntityInfo	18
Figure 12: DataDescriptors JSON EntityInfo	19
Figure 13: DataItems JSON EntityInfo	20
Figure 14: InitialisationDescriptors JSON EntityInfo	23
Figure 15: ManipulationDescriptors JSON EntityInfo.....	25
Figure 16: RenderingDescriptors JSON EntityInfo	26
Figure 17: SimulationDescriptors JSON EntityInfo	28
Figure 18: Transitions JSON EntityInfo	30
Figure 19: Worldstates JSON EntityInfo.....	32

Executive Summary

This document is a proposal for the Core Control and Communication Information Model for the CRISMA Framework.

1. Purpose of this document

This document provides a specification of the Core Control and Communication Information Model (Core CCIM) of the CRISMA Framework. This information model is the basis for all application specific information models (Application CCIMs). The Core CCIM defines the information classes of the Conceptual Business Logic of CRISMA on an abstract and generic level.

The main purpose of the Core CCIM is to allow the management of World States and their Transitions as defined by the Conceptual Business Logic in a uniform manner across all CRISMA Federations. Thus, it leverages the development of generic Building Blocks that can be used in any CRISMA Federation with minimal or no configuration and development effort.

1.1.1. Changes to previous version of the CCIM

The following changes to the previous version (v1.4) of the CCIM have been made:

- The *iccData* attribute of the *Worldstates* class is no longer a single DataItem but an array of DataItems. It is recommended to add three distinct DataItems for Indicators, Criteria and Costs to this array. The actual structure and content of the Indicators, Criteria and Costs however is not constrained by the CCIM. Thus, the *actualAccessInfo* of the an ICC DataItem may contain application specific data like JSON, XML, key value pairs, etc.
- A relation from DataItems to Categories has been added. Thus, worldstate data slots and ICC data can now not only categorised on class level (DataDescriptor) but also on instance level (DataItem).

2. Classes of the core control and communication information model

The Core CCIM defines a minimal set of classes and relationships needed to realise the Conceptual Business Logic of the CRISMA Framework. The Core CCIM can be seen as the least common denominator on which all CRISMA Federations (Applications) have to agree.

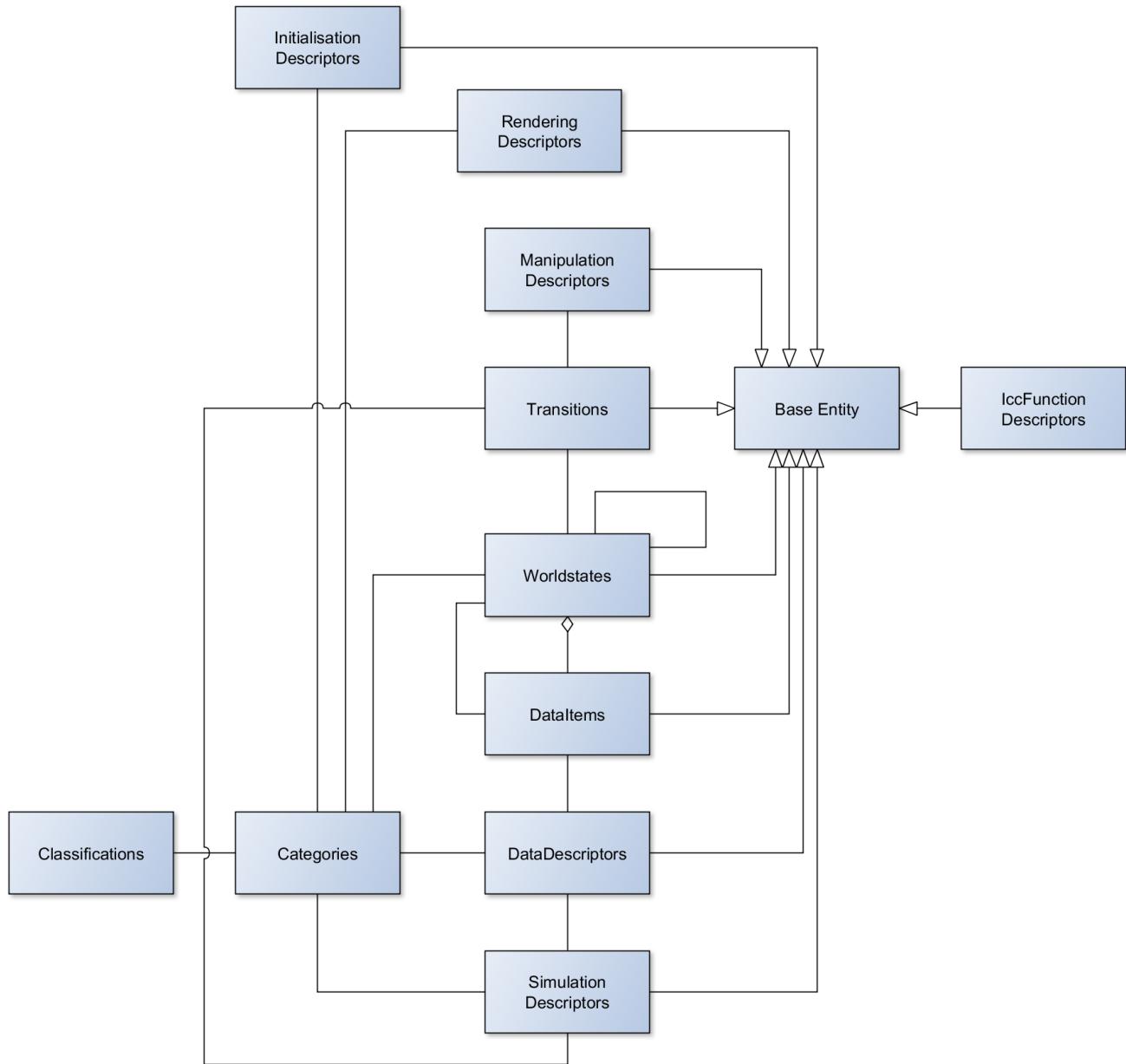


Figure 1: Core CCIM Classes and Relationships

The Core CCIM can be subdivided into two parts: A static part and a dynamic part. The static part has to be defined during the design phase of a CRISMA Application and represents

invariant domain information of this particular CRISMA Application. Once defined, it is not supposed to change.

This part of the CCIM is also called the domain definition of the CRISMA Application. The CCIM Classes of the domain definition are:

- **Categories**

A class representing an arbitrary categories that can be used to categorise scenarios, simulations, etc.

- **Classifications**

A class that represents a classification of categories, e.g. scenario types, crisis management phases, etc.

- **DataDescriptors**

A DataDescriptor provides invariant meta-information about an arbitrary dataset (World State data, simulation control parameters, ...).

- **IccFunctionDescriptors**

IccFunctionDescriptors provide Information about the ICC Functions that are executed after each World State Transition. ICC Functions Descriptors are optional since most functions will be realised by the application logic of a specific Building Block Implementation. Indicator Functions for example are implemented by the Indicator Building Block Software Component.

- **InitialisationDescriptors**

Initialisation descriptors describe user interfaces and business logic (e.g. Composite UI Modules, Setup and Configuration Building Blocks) that are needed to initialize the set of Simulation Models, the initial World State, descriptors, etc.

- **ManipulationDescriptors**

ManipulationDescriptors refer to World State Transitions and describe the user interfaces and business logic (e.g. Composite UI Modules, World States Widget Building Block and associated World State Views) to manipulate the changeable parameters of a World State.

- **RenderingDescriptors**

A Rendering Descriptor describes a User interfaces to visualize the World State (World States Widget Building Block and related Viewer Widgets).

- **SimulationDescriptors**

A SimulationDescriptor provides information about a Federated Simulation (a simulation consisting of one or more Simulation Models wrapped as one WPS service). It holds references (Generic Reference) to concrete descriptions on how to execute the simulation (Simulation Model Integration Building Block) and by which user interface (Simulation Model Interaction Building Block) to control (e.g. parameterise, monitor) the simulation.

The dynamic part of the core CCIM represents the changeable parameters of the world modelled as World States and Transitions. It is represented by the following classes:

- **DataItems**

This class represent an actual data set (instance), e.g. World State data, ICC data, simulation control parameters, etc.

- **Transitions**

This class represents a concrete World State Transition. It contains information on the performed manipulation or simulation, the actual Simulation Control Parameters and further meta-information (e.g. which parts of the World State were changed).

- **World States**

This class represents a concrete World State. A snapshot of the World or World State consists of all data related to a specific crisis simulation experiment.

Please note that **TransitionPoints** are not explicitly modelled in the Core CCIM since they are implicitly defined of by the Simulation Cases which are part of the application logic of the respective CRISMA Application. Thus, the application logic is responsible to create appropriate Transitions that are valid in the scope of a specific Simulation Case Graph. A validation of scenarios (sequences of Transitions) as instances of a Simulation Case Graph on the information model level is therefore not supported.

In summary, the Core CCIM leverages interoperability between Federates and Federations within the whole CRISMA System (the sum of all CRISMA Federations) while extensions leverage interoperability between Federates within one CRISMA Federation. Further details and examples on the intended usage of the Core CCIM are given in the Application Architecture Template in **Fehler! Verweisquelle konnte nicht gefunden werden..** A technical specification of the Core CCIM in UML can be found in **Fehler! Verweisquelle konnte nicht gefunden werden..**

2.1.1. Relation to conceptual business logic

As explained in the previous section, the Core CCIM maps the Conceptual Business Logic to classes of a modular information model.

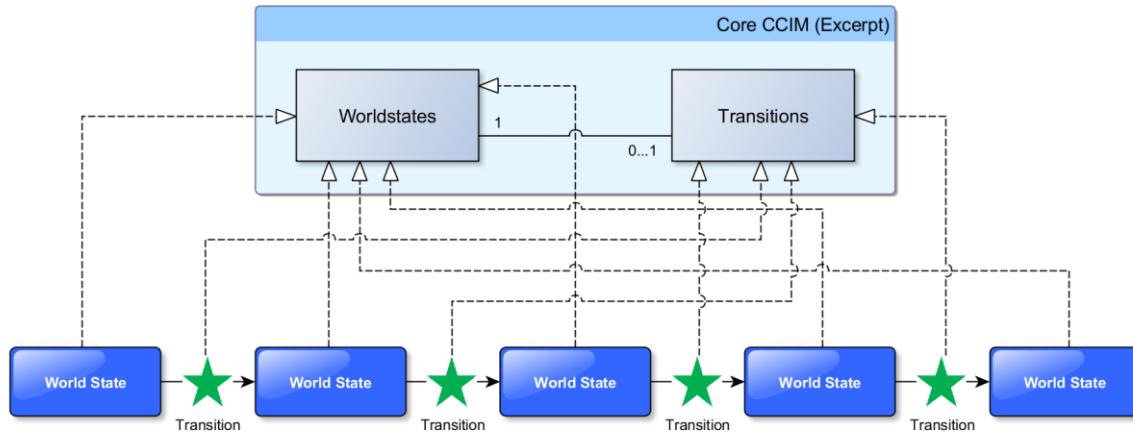


Figure 2: Scenario expressed with help of the Core CCIM

Figure 2 shows how a Simulated Crisis Management Scenario can be expressed by instances of Core CCIM Classes and Figure 3 provides a more detailed view on the relationships of the Core CCIM to the Conceptual Business Logic.

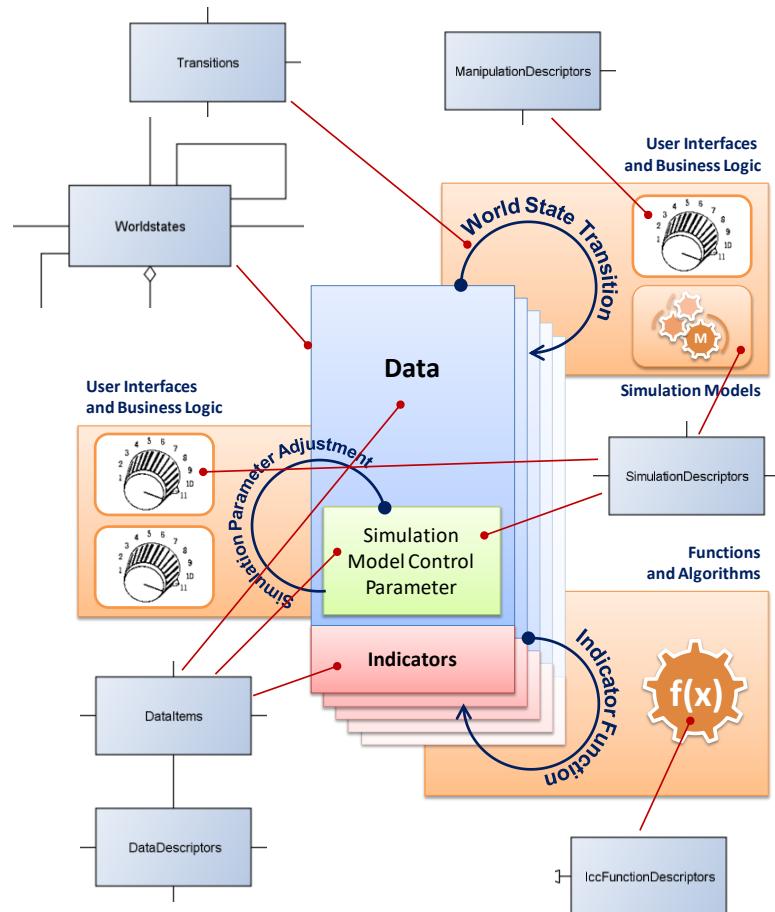


Figure 3: Mapping the Core CCIM to the Conceptual Business Logic (detailed)

3. Technical realisation

Technically, the Core CCIM is implemented as part of a generic and object-oriented relational database model of the Software Component that implements the ICMM Building Block. Thus, implementing a CCIM is a configuration task and does not require any changes to the ICMM itself. Figure 4 shows an excerpt of the Core CCIM implemented as database model of the ICMM.

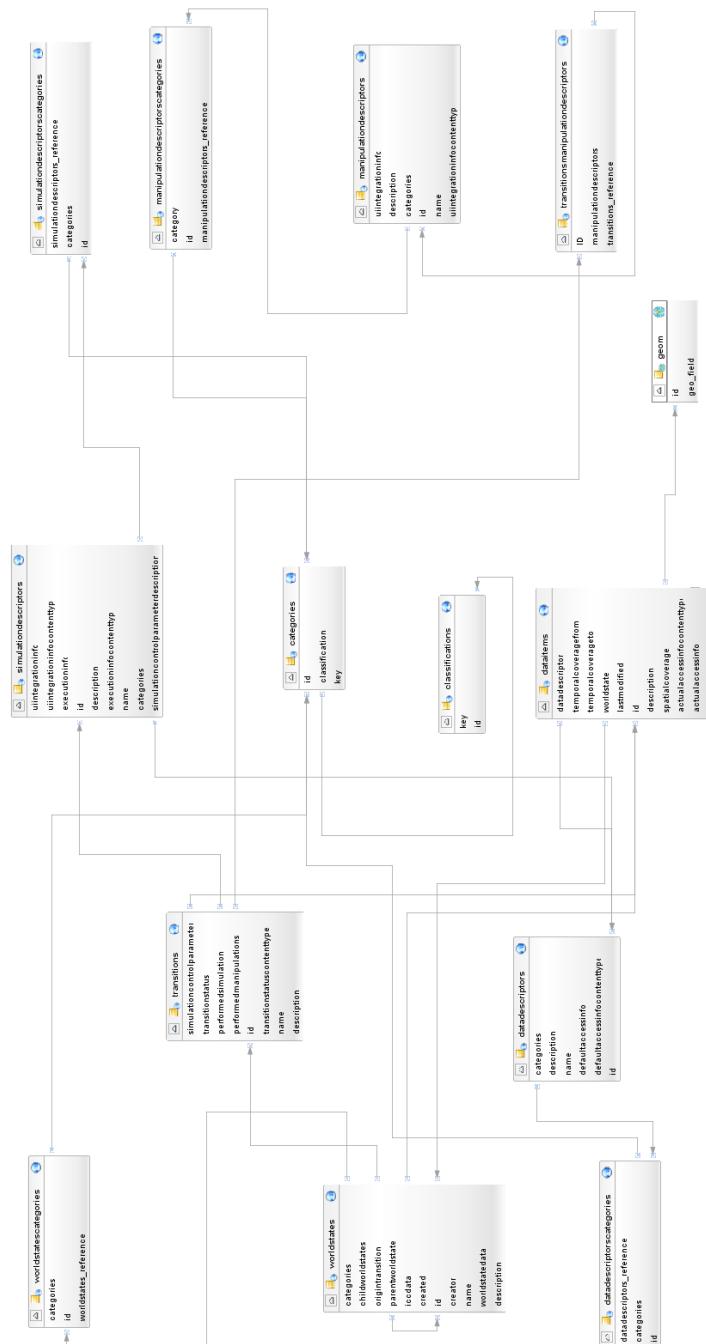


Figure 4: Core CCIM Database Model (excerpt)

Please note that the diagram shown in Figure 4 is not up-to-date and has not been adapted to the last changes to the Core CCIM as described in section 2!

According to the rules for the specification of information models a CCIM modelled in UML has to be encoded in Java Script Object Notation Format (JSON).

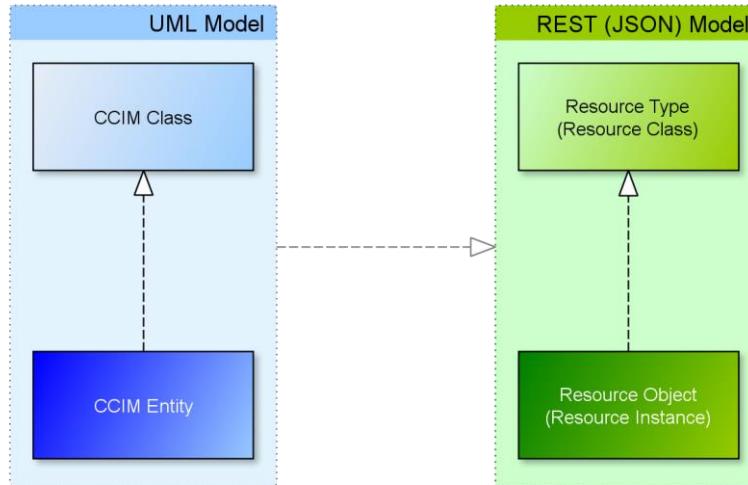


Figure 5: Mapping UML Models to REST Models

The relation between the UML and JSON models is shown in Figure 5. On the level of the UML Model, an instance of a CCIM Class is called CCIM Entity. When the UML Model is encoded in JSON, the CCIM Class becomes a Resource Type or Resource Class and the CCIM Entity a Resource Object or Resource Instance.

The actual CCIM Resource Instances can be accessed via the REST APIs of the ICMM. Figure 6 shows an example of such a Resource Instance.

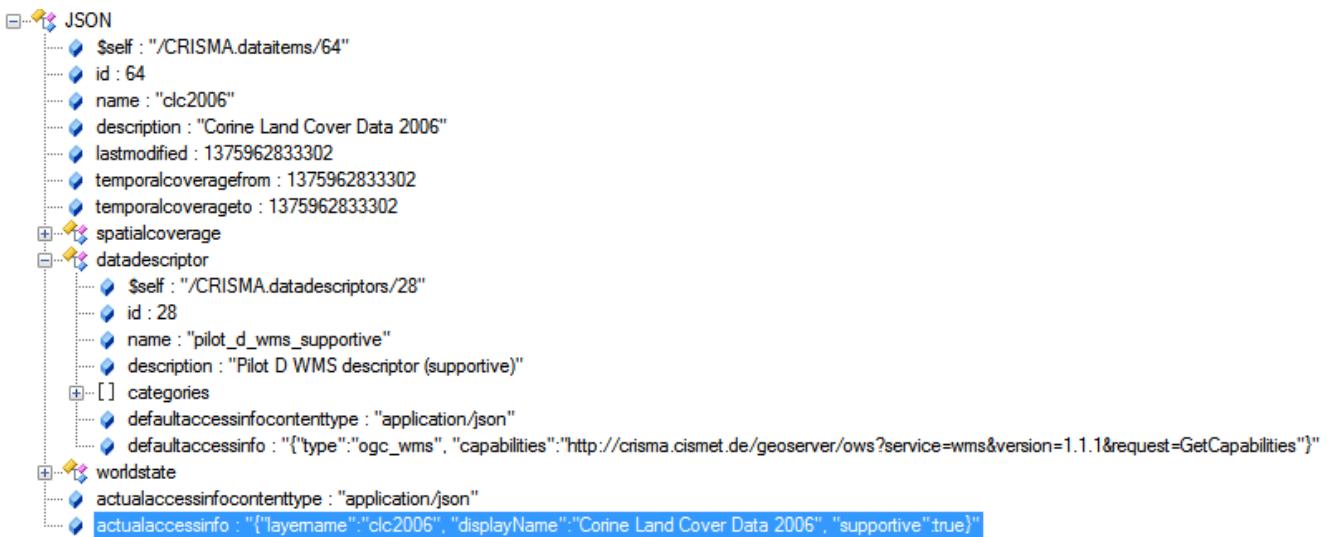


Figure 6: Dataitems Instance JSON

4. Generic extension mechanisms

Since CRISMA Applications (Federations) are very heterogeneous in nature, the Core CCIM is optimised for genericity and reusability and provides only rudimentary meta-information about World States, Transitions, etc.

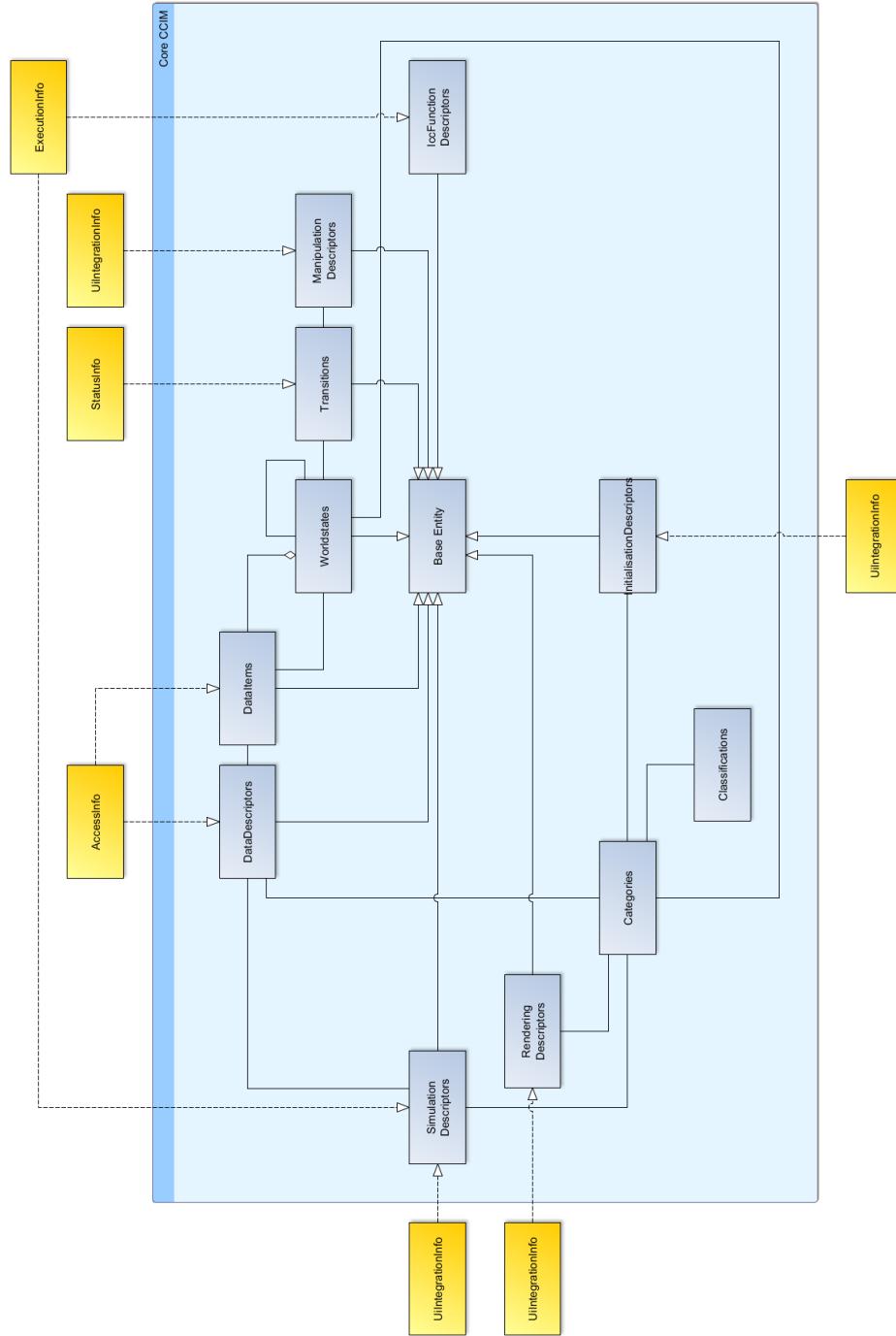


Figure 7: Core CCIM with extension points for Application CCIMs

According to this, it does not define a concrete meta-information schema for data access, the description of manipulation and simulations, etc. Therefore, each CRISMA Federation has to define some federation-specific extensions of the Core CCIM.

For this purpose, the Core CCIM provides two extension mechanisms which are described in the subsequent section: Extension by inheritance and predefined Extension Points. Figure 7 below gives an overview on the generic extension points for Application CCIMs.

4.1.1.1. *Extension by inheritance*

The first possibility for extending the Core CCIM is to directly inherit from Core CCIM Classes (Figure 8). In this way the extended CCIM Classes will become directly part of the ICMM data model (see section 3). The main advantage of this approach is that entities based on such classes can benefit from the built-in search, access control and validation functionalities of the ICMM.

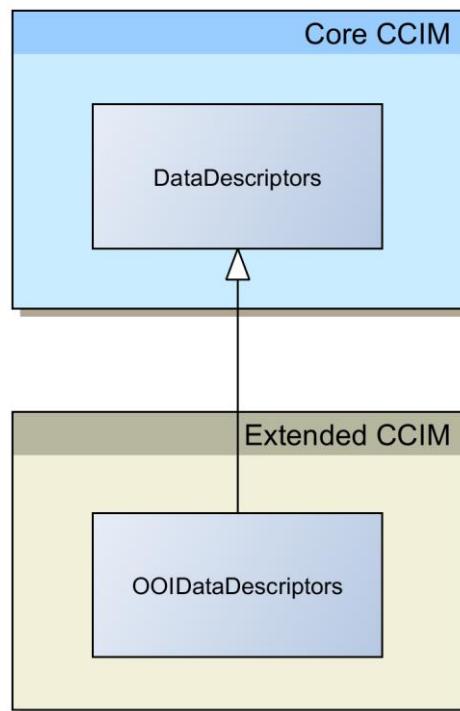


Figure 8: Extension by inheritance

One disadvantage of this approach is that a data model has to be designed and installed into an ICMM instance. Although this is an administrative task which is supported by graphical tools, it requires a certain amount of expert knowledge about the ICMM internals.

4.1.1.2. *Extension Points*

A second, more convenient and easy to use mechanism for the extension of the Core CCIM and CCIMs in general are predefined Extension Points. Extension Points are realised in the Core CCIM by so called Generic References.

A Generic References in a CCIM is a reference to an entity which is not modelled as class of the enclosing CCIM. Such a Generic Reference contains in general domain specific information that may be used by (application specific) components. The actual content type of the Generic Reference is determined by a dedicated content type attribute. Thus, a Generic Reference may point e.g. to a JSON or XML document or even a link to a JSON or XML document.

Interestingly, Generic References may point to arbitrary JSON objects including JSON objects that comply with a CCIM Class. With help of this mechanism it is possible to establish relations to CCIM Classes that are neither part of the Core CCIM nor known during the design time of the Core CCIM. Therefore, although not required, it is nevertheless good practice to model such entities that should be represented by Generic References also compliant to CCIM Classes.

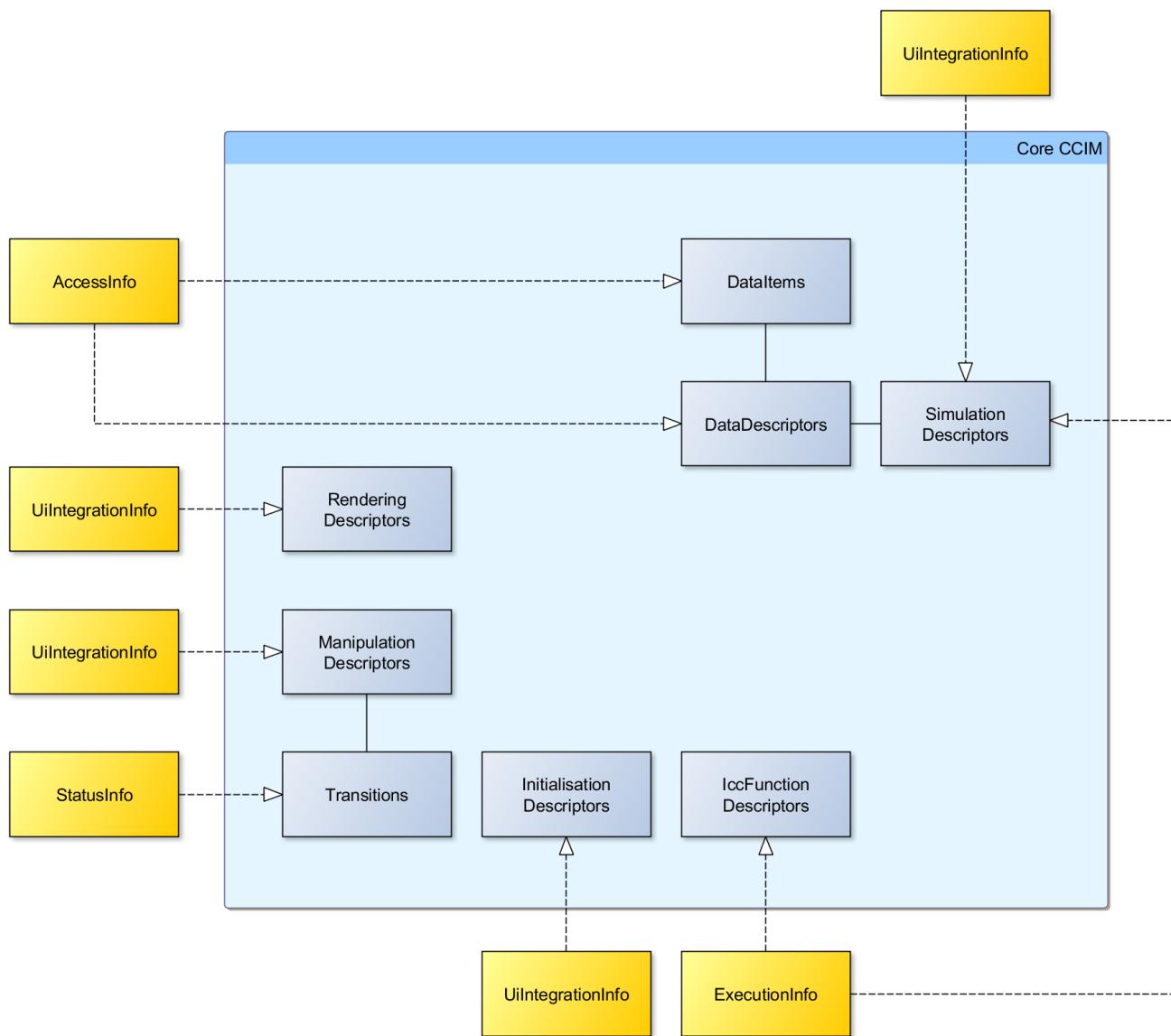


Figure 9: Extension Points of the Core CCIM

In any case, it should be clear that it is always up to the designer of a concrete CRISMA Application to define the actual structure and content of Generic References. Therefore, a Generic Reference which may be suitable for a particular CRISMA Application may not be usable in another Application.

Figure 9 shows the Extension Points of the Core CCIM, shown as yellow boxes. Each of those Generic References is modelled in the Core CCIM as two String attributes: The actual content of the reference and the content type of the reference.

The following types of Generic References are defined in the Core CCIM:

- **AccessInfo** is the extension Point of the DataDescriptors and DataItems Classes. It can be divided into defaultAccessInfo (relevant for DataDescriptors) and actualAccessInfo (relevant for DataItems).
 - **defaultAccessInfo** describes the actual data source, e.g. a link to an external meta data document describing the data set or other types of access information and credentials (e.g. WFS URI). It represents invariant meta-information that stays the same for all DataItems that are described by a DataDescriptor.
 - **actualAccessInfo** is the actual information on how to access the data. In rare cases or for very small and simple datasets (e.g. ICC Data) it may contain also the data itself. In general, it provides information to identify an individual data item or a specific version of a data item (e.g. a specific request parameter, a filename, etc.).
- **UIIntegrationInfo** is the Extension Point of the Rendering-, Manipulation-, Simulation- and InitialisationDescriptors. It provides the actual information on how to instantiate and integrate the respective user interface (Composite UI Module). Depending on the CRISMA Application it may contain an URI that points to a Composite UI Module instance or more detailed information.
- **ExecutionInfo** is the Extension Point of Simulation- and IccFunctionDescriptors. It provides information on how to execute a federated simulation or an ICC Function. In general, it contains information on the WPS that provides the simulation (Simulation Model Integration Building Block). The concrete details depend on the types of simulations available in a particular CRISMA Application.
- **StatusInfo** is the Extension Point of the Transition Class. It provides detailed Transition status information. It may be represented by an arbitrary CRISMA JSON object that is defined for a specific Application CCIM and may be injected dynamically, e.g. by the Simulation Model Integration or Interaction Building Blocks. It may also point to the raw Transition status, e.g. a log file, a link to a website with status messages, a WPS URL, etc.

Several domain-independent Generic References modelled as JSON entities are provided in the Application Architecture Template.

5. JSON Encoding

The following subchapters provide the JSON Encoding of the current Core CCIM Classes.

5.1. Categories

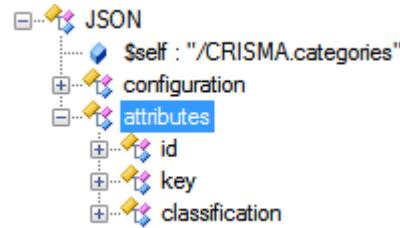


Figure 10: Categories JSON EntityInfo

```

{
    "$self" : "/CRISMA.categories",
    "configuration" : {
        "name" : "Categories",
        "policy" : "STANDARD",
        "attributePolicy" : "STANDARD",
        "pK Field" : "id"
    },
    "attributes" : {
        "id" : {
            "$self" : "/CRISMA.categories/id",
            "name" : "id",
            "position" : 0,
            "javaclassname" : "java.lang.Integer"
        },
        "key" : {
            "$self" : "/CRISMA.categories/key",
            "visible" : true,
            "name" : "key",
            "position" : 1,
            "javaclassname" : "java.lang.String"
        },
        "classification" : {
            "$self" : "/CRISMA.categories/classification",
            "visible" : true,
            "optional" : true,
            "name" : "classification",
            "position" : 2,
            "foreignKey" : true,
            "referenceType" : "/CRISMA.classifications"
        }
    }
}
  
```

}

5.2. *Classifications*

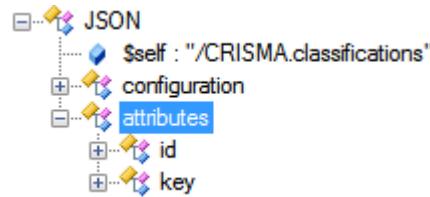


Figure 11: Classifications JSON EntityInfo

```

    {
        "$self" : "/CRISMA.classifications",
        "configuration" : {
            "name" : "Classifications",
            "policy" : "STANDARD",
            "attributePolicy" : "STANDARD",
            "pK Field" : "id"
        },
        "attributes" : {
            "id" : {
                "$self" : "/CRISMA.classifications/id",
                "name" : "id",
                "position" : 0,
                "javaclassname" : "java.lang.Integer"
            },
            "key" : {
                "$self" : "/CRISMA.classifications/key",
                "visible" : true,
                "name" : "key",
                "position" : 1,
                "javaclassname" : "java.lang.String"
            }
        }
    }

```

5.3. DataDescriptors

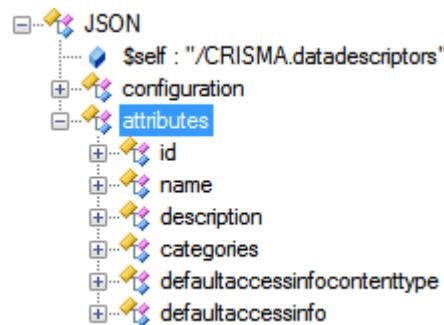


Figure 12: DataDescriptors JSON EntityInfo

```

{
    "$self": "/CRISMA.datadescrptors",
    "configuration": {
        "name": "DataDescriptors",
        "policy": "STANDARD",
        "attributePolicy": "STANDARD",
        "pK Field": "id"
    },
    "attributes": {
        "id": {
            "$self": "/CRISMA.datadescrptors/id",
            "name": "id",
            "position": 0,
            "javaclassname": "java.lang.Integer"
        },
        "name": {
            "$self": "/CRISMA.datadescrptors/name",
            "visible": true,
            "optional": true,
            "name": "name",
            "position": 1,
            "javaclassname": "java.lang.String"
        },
        "description": {
            "$self": "/CRISMA.datadescrptors/description",
            "visible": true,
            "optional": true,
            "name": "description",
            "position": 2,
            "javaclassname": "java.lang.String"
        },
        "categories": {
            "$self": "/CRISMA.datadescrptors/categories",
            "visible": true
        }
}

```

```
        "array" : true,
        "optional" : true,
        "name" : "categories",
        "position" : 3,
        "referenceType" : "/CRISMA.categories"
    },
    "defaultaccessinfocontenttype" : {
        "$self" : "/CRISMA.datadescr/defaultaccessinfocontenttype",
        "visible" : true,
        "optional" : true,
        "name" : "defaultAccessInfoContentType",
        "position" : 4,
        "javaclassname" : "java.lang.String"
    },
    "defaultaccessinfo" : {
        "$self" : "/CRISMA.datadescr/defaultaccessinfo",
        "visible" : true,
        "optional" : true,
        "name" : "defaultAccessInfo",
        "position" : 5,
        "javaclassname" : "java.lang.String"
    }
}
```

5.4. *DataItems*

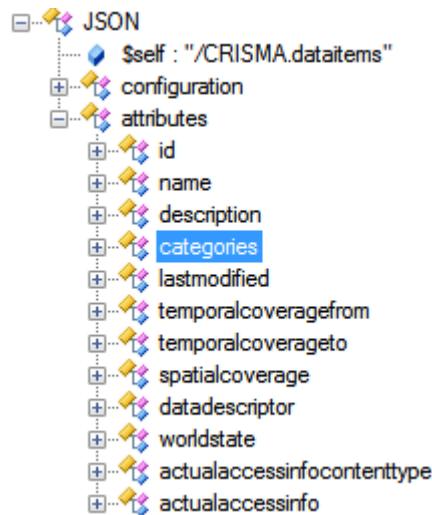


Figure 13: DataItems JSON EntityInfo

```
"$self": "/CRISMA.dataitems",
"configuration": {
    "name": "DataItems",
    "policy": "STANDARD",
    "attributePolicy": "STANDARD",
    "pK Field": "id"
},
"attributes": {
    "id": {
        "$self": "/CRISMA.dataitems/id",
        "name": "id",
        "position": 0,
        "javaclassname": "java.lang.Integer"
    },
    "name": {
        "$self": "/CRISMA.dataitems/name",
        "visible": true,
        "optional": true,
        "name": "name",
        "position": 1,
        "javaclassname": "java.lang.String"
    },
    "description": {
        "$self": "/CRISMA.dataitems/description",
        "visible": true,
        "optional": true,
        "name": "description",
        "position": 2,
        "javaclassname": "java.lang.String"
    },
    "categories": {
        "$self": "/CRISMA.datadescriptors/categories",
        "visible": true,
        "array": true,
        "optional": true,
        "name": "categories",
        "position": 3,
        "referenceType": "/CRISMA.categories"
    },
    "lastmodified": {
        "$self": "/CRISMA.dataitems/lastmodified",
        "visible": true,
        "optional": true,
        "name": "lastModified",
        "position": 4,
        "javaclassname": "java.sql.Timestamp"
    },
    "temporalcoveragefrom": {
        "$self": "/CRISMA.dataitems/temporalcoveragefrom",
        "visible": true,
        "optional": true,
        "name": "temporalcoveragefrom",
        "position": 5,
        "javaclassname": "java.util.Date"
    }
}
```

```

        "visible": true,
        "optional": true,
        "name": "temporalCoverageFrom",
        "position": 5,
        "javaclassname": "java.sql.Timestamp"
    },
    "temporalcoverageto": {
        "$self": "/CRISMA.dataitems/temporalcoverageto",
        "visible": true,
        "optional": true,
        "name": "temporalCoverageTo",
        "position": 6,
        "javaclassname": "java.sql.Timestamp"
    },
    "spatialcoverage": {
        "$self": "/CRISMA.dataitems/spatialcoverage",
        "visible": true,
        "optional": true,
        "name": "spatialCoverage",
        "position": 7,
        "foreignKey": true,
        "referenceType": "/CRISMA.geom"
    },
    "datadescriptor": {
        "$self": "/CRISMA.dataitems/datadescriptor",
        "visible": true,
        "optional": true,
        "name": "dataDescriptor",
        "position": 8,
        "foreignKey": true,
        "referenceType": "/CRISMA.datadescrptors"
    },
    "worldstate": {
        "$self": "/CRISMA.dataitems/worldstate",
        "visible": true,
        "optional": true,
        "name": "worldstate",
        "position": 9,
        "foreignKey": true,
        "referenceType": "/CRISMA.worldstates"
    },
    "actualaccessinfocontenttype": {
        "$self": "/CRISMA.dataitems/actualaccessinfocontenttype",
        "visible": true,
        "optional": true,
        "name": "actualAccessInfoContentType",
        "position": 10,
        "javaclassname": "java.lang.String"
    },

```

```

        "actualaccessinfo": {
            "$self": "/CRISMA.dataitems/actualaccessinfo",
            "visible": true,
            "optional": true,
            "name": "actualAccessInfo",
            "position": 11,
            "javaclassname": "java.lang.String"
        }
    }
}
}

```

5.5. InitialisationDescriptors

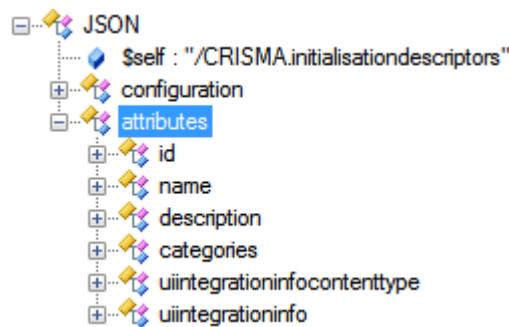


Figure 14: InitialisationDescriptors JSON EntityInfo

```

{
    "$self": "/CRISMA.initialisationdescriptors",
    "configuration": {
        "name": "InitialisationDescriptors",
        "policy": "STANDARD",
        "attributePolicy": "STANDARD",
        "pK_Field": "id"
    },
    "attributes": {
        "id": {
            "$self": "/CRISMA.initialisationdescriptors/id",
            "name": "id",
            "position": 0,
            "javaclassname": "java.lang.Integer"
        },
        "name": {
            "$self": "/CRISMA.initialisationdescriptors/name",
            "visible": true,
            "optional": true,
            "name": "name",
            "position": 2,
            "javaclassname": "java.lang.String"
        }
    }
}

```

```
        "javaclassname": "java.lang.String"
    },
    "description": {
        "$self": "/CRISMA.initialisationdescriptors/description",
        "visible": true,
        "optional": true,
        "name": "description",
        "position": 3,
        "javaclassname": "java.lang.String"
    },
    "categories": {
        "$self": "/CRISMA.initialisationdescriptors/categories",
        "visible": true,
        "array": true,
        "optional": true,
        "name": "categories",
        "position": 4,
        "referenceType": "/CRISMA.categories"
    },
    "uiintegrationinfocontenttype": {
        "$self": "/CRISMA.initialisationdescriptors/uiintegrationinfocontenttype",
        "visible": true,
        "optional": true,
        "name": "uiIntegrationInfoContentType",
        "position": 5,
        "javaclassname": "java.lang.String"
    },
    "uiintegrationinfo": {
        "$self": "/CRISMA.initialisationdescriptors/uiintegrationinfo",
        "visible": true,
        "optional": true,
        "name": "uiIntegrationInfo",
        "position": 6,
        "javaclassname": "java.lang.String"
    }
}
```

5.6. ManipulationDescriptors

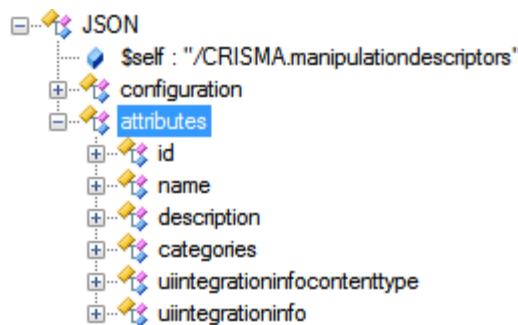


Figure 15: ManipulationDescriptors JSON EntityInfo

```

    {
        "$self": "/CRISMA.manipulationdescriptors",
        "configuration": {
            "name": "ManipulationDescriptors",
            "policy": "STANDARD",
            "attributePolicy": "STANDARD",
            "pK Field": "id"
        },
        "attributes": {
            "id": {
                "$self": "/CRISMA.manipulationdescriptors/id",
                "name": "id",
                "position": 0,
                "javaclassname": "java.lang.Integer"
            },
            "name": {
                "$self": "/CRISMA.manipulationdescriptors/name",
                "visible": true,
                "optional": true,
                "name": "name",
                "position": 1,
                "javaclassname": "java.lang.String"
            },
            "description": {
                "$self": "/CRISMA.manipulationdescriptors/description",
                "visible": true,
                "optional": true,
                "name": "description",
                "position": 2,
                "javaclassname": "java.lang.String"
            },
            "categories": {
                "$self": "/CRISMA.manipulationdescriptors/categories",
                "visible": true,
                "array": true
            }
        }
    }

```

```
        "optional": true,
        "name": "categories",
        "position": 3,
        "referenceType": "/CRISMA.categories"
    },
    "uiintegrationinfocontenttype": {
        "$self": "/CRISMA.manipulationdescriptors/uiintegrationinfocontenttype",
        "visible": true,
        "optional": true,
        "name": "uiIntegrationInfoContentType",
        "position": 4,
        "javaclassname": "java.lang.String"
    },
    "uiintegrationinfo": {
        "$self": "/CRISMA.manipulationdescriptors/uiintegrationinfo",
        "visible": true,
        "optional": true,
        "name": "uiIntegrationInfo",
        "position": 5,
        "javaclassname": "java.lang.String"
    }
}
```

5.7. *RenderingDescriptors*

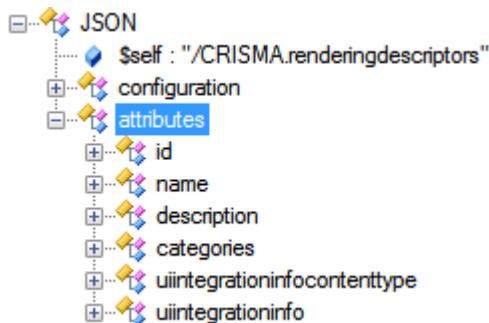


Figure 16: RenderingDescriptors JSON EntityInfo

```
{  
    "$self": "/CRISMA.renderingdescriptors",  
    "configuration": {  
        "name": "RenderingDescriptors",  
        "policy": "STANDARD",  
        "attributePolicy": "STANDARD",  
        "pK_Field": "id"  
    },  
}
```

```

"attributes": {
    "id": {
        "$self": "/CRISMA.renderingdescriptors/id",
        "name": "id",
        "position": 0,
        "javaclassname": "java.lang.Integer"
    },
    "name": {
        "$self": "/CRISMA.renderingdescriptors/name",
        "visible": true,
        "optional": true,
        "name": "name",
        "position": 1,
        "javaclassname": "java.lang.String"
    },
    "description": {
        "$self": "/CRISMA.renderingdescriptors/description",
        "visible": true,
        "optional": true,
        "name": "description",
        "position": 2,
        "javaclassname": "java.lang.String"
    },
    "categories": {
        "$self": "/CRISMA.renderingdescriptors/categories",
        "visible": true,
        "array": true,
        "optional": true,
        "name": "categories",
        "position": 3,
        "referenceType": "/CRISMA.categories"
    },
    "uiintegrationinfocontenttype": {
        "$self": "/CRISMA.renderingdescriptors/uiintegrationinfocontenttype",
        "visible": true,
        "optional": true,
        "name": "uiIntegrationInfoContentType",
        "position": 4,
        "javaclassname": "java.lang.String"
    },
    "uiintegrationinfo": {
        "$self": "/CRISMA.renderingdescriptors/uiintegrationinfo",
        "visible": true,
        "optional": true,
        "name": "uiIntegrationInfo",
        "position": 5,
        "javaclassname": "java.lang.String"
    }
}

```

}

5.8. *SimulationDescriptors*

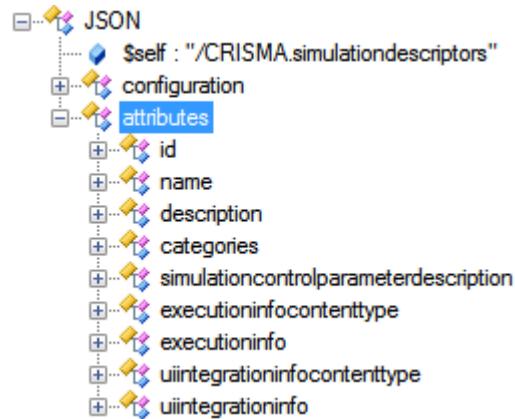


Figure 17: SimulationDescriptors JSON EntityInfo

```

{
    "$self": "/CRISMA.simulationdescriptors",
    "configuration": {
        "name": "SimulationDescriptors",
        "policy": "STANDARD",
        "attributePolicy": "STANDARD",
        "pK Field": "id"
    },
    "attributes": {
        "id": {
            "$self": "/CRISMA.simulationdescriptors/id",
            "name": "id",
            "position": 0,
            "javaclassname": "java.lang.Integer"
        },
        "name": {
            "$self": "/CRISMA.simulationdescriptors/name",
            "visible": true,
            "optional": true,
            "name": "name",
            "position": 1,
            "javaclassname": "java.lang.String"
        },
        "description": {
            "$self": "/CRISMA.simulationdescriptors/description",
            "visible": true,
            "optional": true,
            "name": "description",
            "position": 2,
            "javaclassname": "java.lang.String"
        }
    }
}
  
```

```

        "name": "description",
        "position": 2,
        "javaclassname": "java.lang.String"
    },
    "categories": {
        "$self": "/CRISMA.simulationdescriptors/categories",
        "visible": true,
        "array": true,
        "optional": true,
        "name": "categories",
        "position": 3,
        "referenceType": "/CRISMA.categories"
    },
    "simulationcontrolparameterdescription": {
        "$self": "/CRISMA.simulationdescriptors/simulationcontrolparameterdescription",
        "visible": true,
        "optional": true,
        "name": "simulationControlParameterDescription",
        "position": 4,
        "foreignKey": true,
        "referenceType": "/CRISMA.datadescriptors"
    },
    "executioninfocontenttype": {
        "$self": "/CRISMA.simulationdescriptors/executioninfocontenttype",
        "visible": true,
        "optional": true,
        "name": "executionInfoContentType",
        "position": 5,
        "javaclassname": "java.lang.String"
    },
    "executioninfo": {
        "$self": "/CRISMA.simulationdescriptors/executioninfo",
        "visible": true,
        "optional": true,
        "name": "executionInfo",
        "position": 6,
        "javaclassname": "java.lang.String"
    },
    "uiintegrationinfocontenttype": {
        "$self": "/CRISMA.simulationdescriptors/uiintegrationinfocontenttype",
        "visible": true,
        "optional": true,
        "name": "uiIntegrationInfoContentType",
        "position": 7,
        "javaclassname": "java.lang.String"
    },
    "uiintegrationinfo": {
        "$self": "/CRISMA.simulationdescriptors/uiintegrationinfo",
        "visible": true,

```

```
        "optional": true,  
        "name": "uiIntegrationInfo",  
        "position": 8,  
        "javaclassname": "java.lang.String"  
    }  
}  
}
```

5.9. *Transitions*

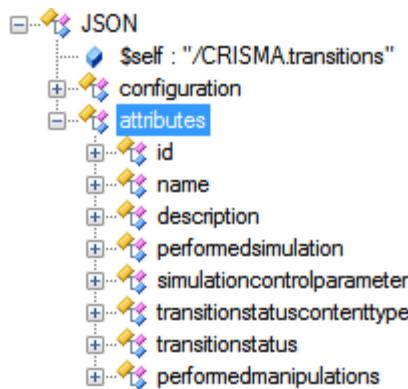


Figure 18: Transitions JSON EntityInfo

```
{
    "$self": "/CRISMA.transitions",
    "configuration": {
        "name": "Transitions",
        "policy": "STANDARD",
        "attributePolicy": "STANDARD",
        "pK Field": "id"
    },
    "attributes": {
        "id": {
            "$self": "/CRISMA.transitions/id",
            "name": "id",
            "position": 0,
            "javaclassname": "java.lang.Integer"
        },
        "name": {
            "$self": "/CRISMA.transitions/name",
            "visible": true,
            "optional": true,
            "name": "name",
            "position": 1,
            "javaclassname": "java.lang.String"
        }
    }
}
```

```
},
"description": {
    "$self": "/CRISMA.transitions/description",
    "visible": true,
    "optional": true,
    "name": "description",
    "position": 2,
    "javaclassname": "java.lang.String"
},
"performedsimulation": {
    "$self": "/CRISMA.transitions/performedsimulation",
    "visible": true,
    "optional": true,
    "name": "performedSimulation",
    "position": 4,
    "foreignKey": true,
    "referenceType": "/CRISMA.simulationdescriptors"
},
"simulationcontrolparameter": {
    "$self": "/CRISMA.transitions/simulationcontrolparameter",
    "visible": true,
    "optional": true,
    "name": "simulationControlParameter",
    "position": 5,
    "foreignKey": true,
    "referenceType": "/CRISMA.dataitems"
},
"transitionstatuscontenttype": {
    "$self": "/CRISMA.transitions/transitionstatuscontenttype",
    "visible": true,
    "optional": true,
    "name": "transitionStatusContentType",
    "position": 6,
    "javaclassname": "java.lang.String"
},
"transitionstatus": {
    "$self": "/CRISMA.transitions/transitionstatus",
    "visible": true,
    "optional": true,
    "name": "transitionStatus",
    "position": 7,
    "javaclassname": "java.lang.String"
},
"performedmanipulations": {
    "$self": "/CRISMA.transitions/PerformedManipulations",
    "visible": true,
    "array": true,
    "optional": true,
    "name": "performedManipulations",
    "position": 8
},
```

```
        "position": 8,  
        "referenceType": "/CRISMA.manipulationdescriptors"  
    }  
}  
}
```

5.10. Worldstates

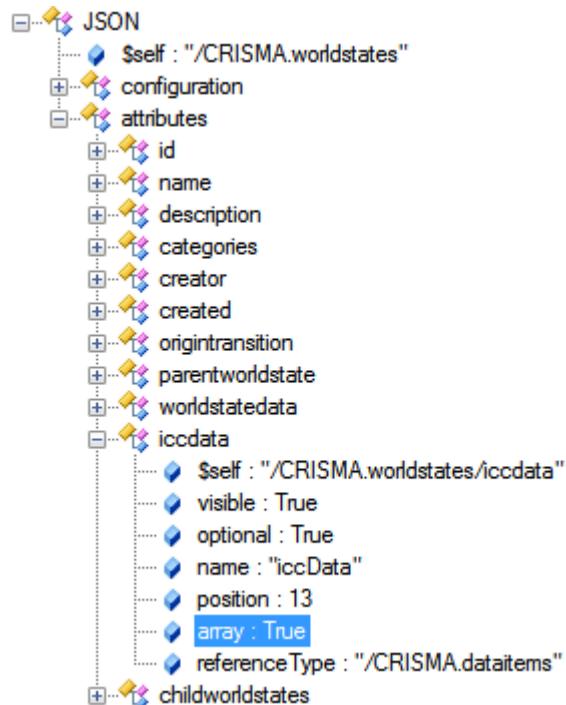


Figure 19: Worldstates JSON EntityInfo

```
{  
    "$self": "/CRISMA.worldstates",  
    "configuration": {  
        "name": "Worldstates",  
        "policy": "STANDARD",  
        "attributePolicy": "STANDARD",  
        "pK Field": "id"  
    },  
    "attributes": {  
        "id": {  
            "$self": "/CRISMA.worldstates/id",  
            "name": "id",  
            "position": 0,  
            "javaclassname": "java.lang.Integer"  
        }  
    }  
}
```

```

},
"name": {
    "$self": "/CRISMA.worldstates/name",
    "visible": true,
    "optional": true,
    "name": "name",
    "position": 1,
    "javaclassname": "java.lang.String"
},
"description": {
    "$self": "/CRISMA.worldstates/description",
    "visible": true,
    "optional": true,
    "name": "description",
    "position": 2,
    "javaclassname": "java.lang.String"
},
"categories": {
    "$self": "/CRISMA.worldstates/categories",
    "visible": true,
    "array": true,
    "optional": true,
    "name": "categories",
    "position": 3,
    "referenceType": "/CRISMA.categories"
},
"creator": {
    "$self": "/CRISMA.worldstates/creator",
    "visible": true,
    "optional": true,
    "name": "creator",
    "position": 5,
    "javaclassname": "java.lang.String"
},
"created": {
    "$self": "/CRISMA.worldstates/created",
    "visible": true,
    "optional": true,
    "name": "created",
    "position": 6,
    "javaclassname": "java.sql.Timestamp"
},
"origintransition": {
    "$self": "/CRISMA.worldstates/origintransition",
    "visible": true,
    "optional": true,
    "name": "originTransition",
    "position": 8,
    "foreignKey": true,
}

```

```

        "referenceType": "/CRISMA.transitions"
    },
    "parentworldstate": {
        "$self": "/CRISMA.worldstates/parentworldstate",
        "visible": true,
        "optional": true,
        "name": "parentWorldstate",
        "position": 9,
        "foreignKey": true,
        "referenceType": "/CRISMA.worldstates"
    },
    "worldstatedata": {
        "$self": "/CRISMA.worldstates/worldstatedata",
        "visible": true,
        "optional": true,
        "name": "worldstateData",
        "position": 12,
        "array": true,
        "referenceType": "/CRISMA.dataitems"
    },
    "iccdatas": {
        "$self": "/CRISMA.worldstates/iccdatas",
        "visible": true,
        "optional": true,
        "name": "iccData",
        "position": 13,
        "array": true,
        "referenceType": "/CRISMA.dataitems"
    },
    "childworldstates": {
        "$self": "/CRISMA.worldstates/childworldstates",
        "visible": true,
        "optional": true,
        "name": "childWorldstates",
        "position": 14,
        "array": true,
        "referenceType": "/CRISMA.worldstates"
    }
}
}
}

```

6. Model Documentation

Core Control and Communication Information Models

Type: Package

Status: Proposed. Version . Phase 1.0.

Package: CCIMs

Detail: *Created on 14.06.2013 11:35:36. Last modified on 04.07.2013 12:46:10*

GUID: {1820DFE4-C5F8-4bc2-A1DC-CDEEE7C520EA}

This package contains common and generic Core Control and Communication Information Models. They can be adapted and extended for each individual CRISMA Application.

Categories - (Class diagram)

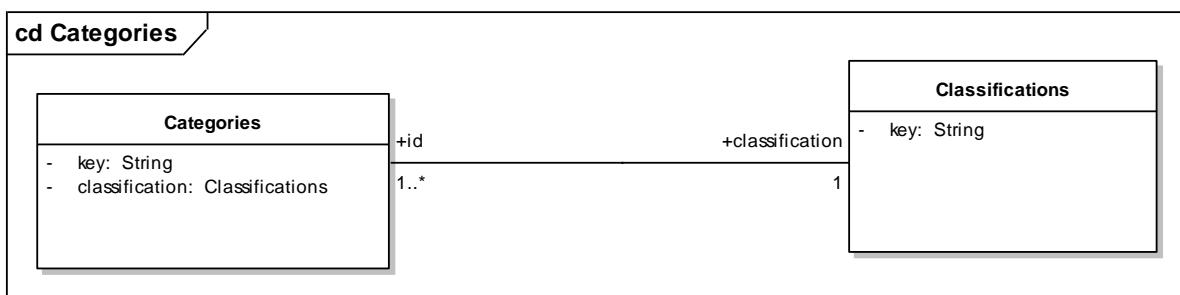
Created By: Pascal Dihé on 20.06.2013

Last Modified: 04.07.2013

Version: 1.0. False

GUID: {C648B407-A3D5-4131-AE4F-21A3B2DA3924}

Categories are used to categorize worldstates, data items and descriptors, transitions, etc. They can be used for search and filtering, e.g. to filter for all data descriptors that represent worldstate data, for all transitions that represent a manipulation, etc.



Complete - (Class diagram)

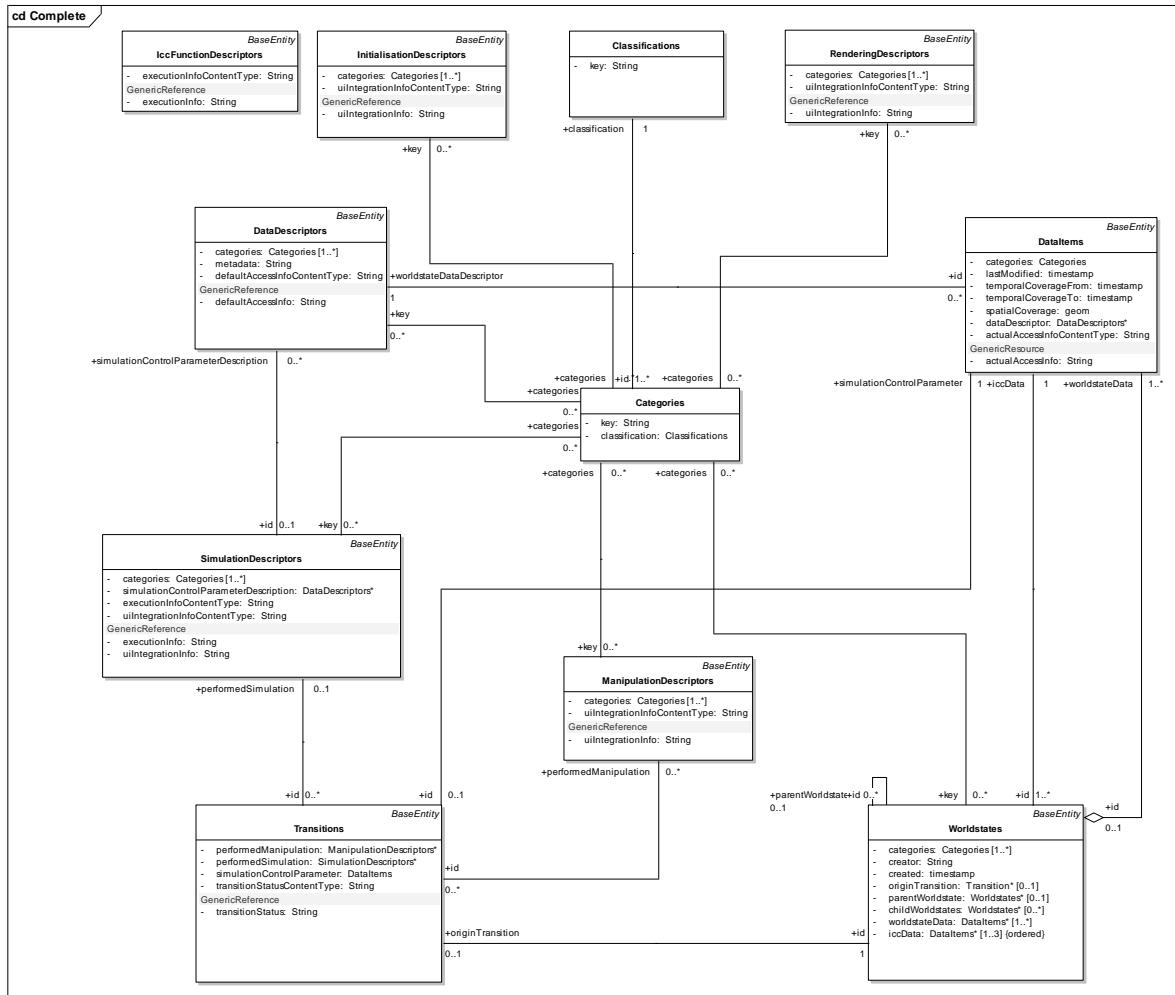
Created By: Pascal Dihé on 25.06.2013

Last Modified: 05.02.2014

Version: 1.0. False

GUID: {C17B2263-F28A-4f4b-918B-C59FA7D1E244}

This class diagram gives a complete overview of the current Core CCIM of CRISMA.



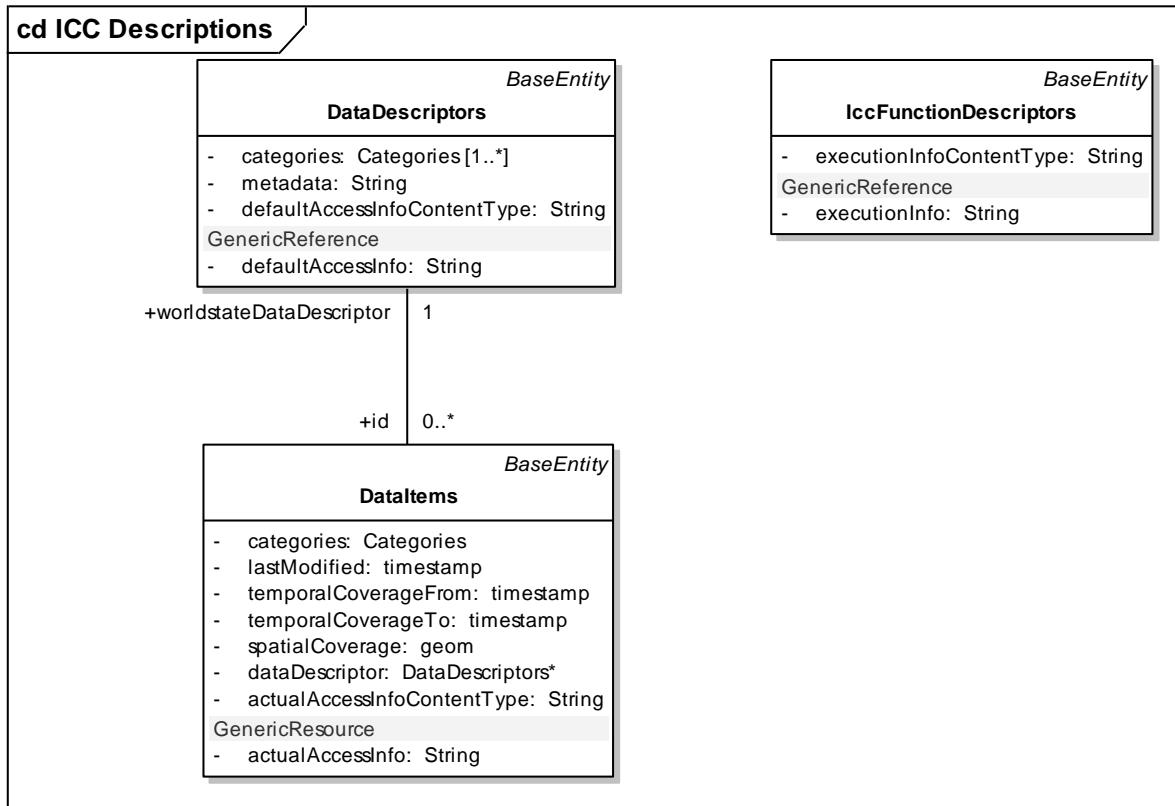
ICC Descriptions - (*Class diagram*)

Created By: Pascal Dihé on 21.11.2013

Last Modified: 05.02.2014

Version: 1.0. False

GUID:



Metamodel - (Class diagram)

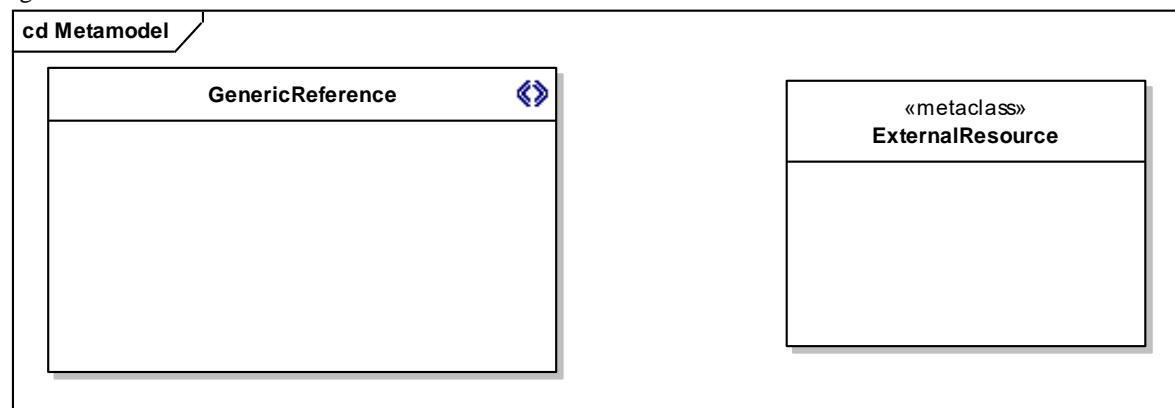
Created By: Pascal Dihé on 20.06.2013

Last Modified: 04.07.2013

Version: 1.0. False

GUID: {0938C8FA-F6FB-4628-B16D-7DC9B7FF01BB}

This diagram visualises the meta model of the Core CCIM.



Simulation Case - (Class diagram)

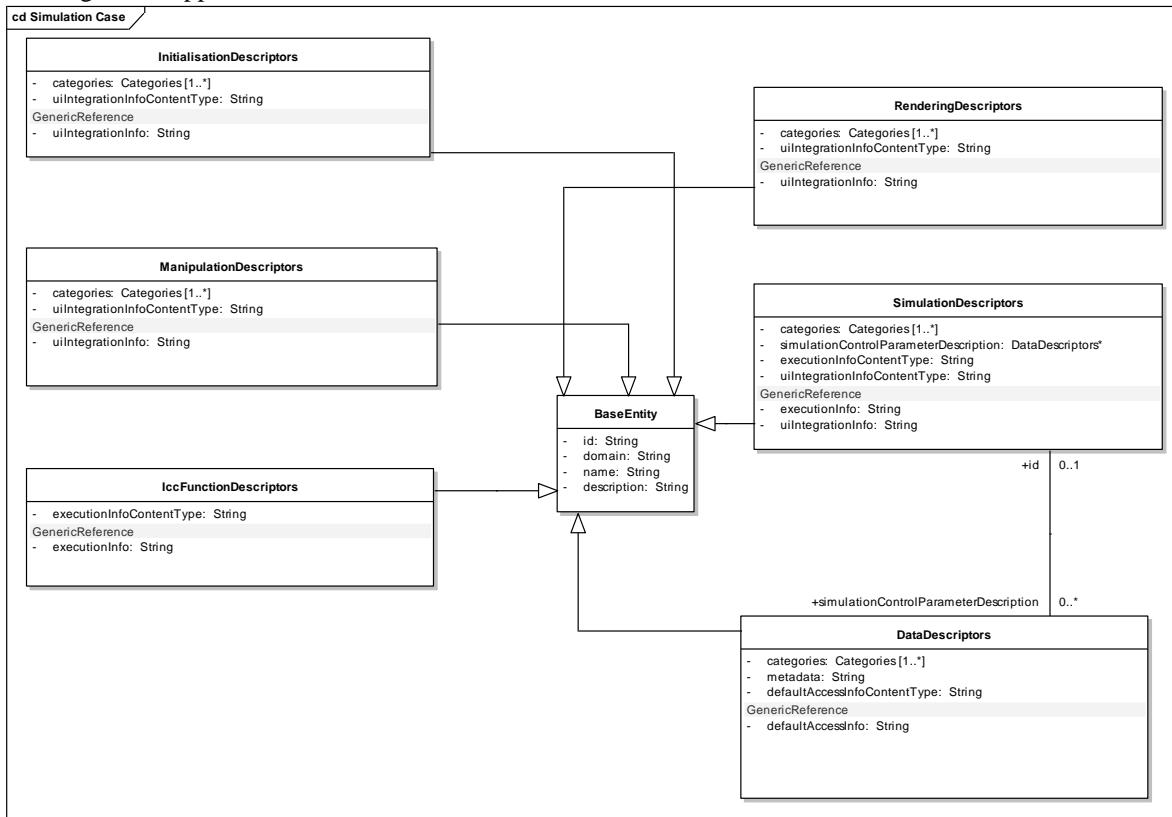
Created By: Pascal Dihé on 19.06.2013

Last Modified: 04.07.2013

Version: 1.0. False

GUID: {9F103F38-B1E5-4a1b-8F6F-9DCC4AF53B7B}

This class diagram visualises all classes of the Core CCIM that form a Simulation Case, that is, a concrete setup of a CRISIS Management Application.



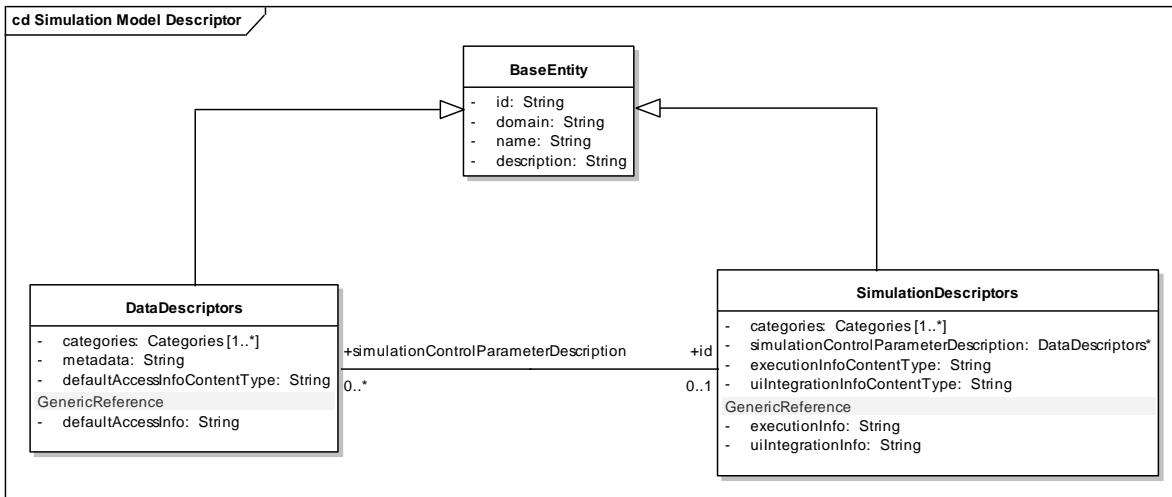
Simulation Model Descriptor - (Class diagram)

Created By: Pascal Dihé on 21.11.2013

Last Modified: 21.11.2013

Version: 1.0. False

GUID: {638087EB-3953-43ba-B561-D33882624402}



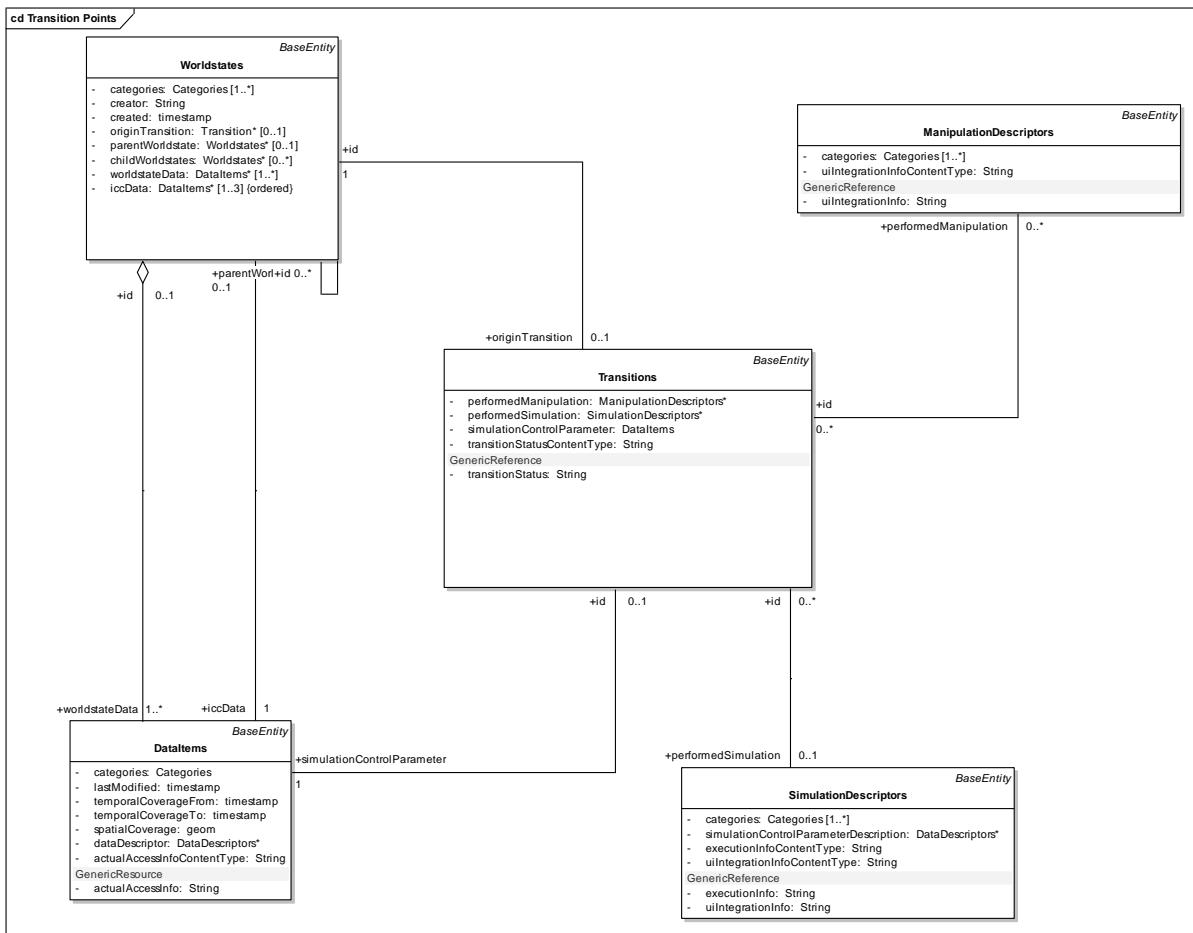
Transition Points - (Class diagram)

Created By: Pascal Dihé on 25.11.2013

Last Modified: 25.11.2013

Version: 1.0. False

GUID: {9E382EE5-280C-47a5-A27D-C84F9182E9EE}



Transitions - (Class diagram)

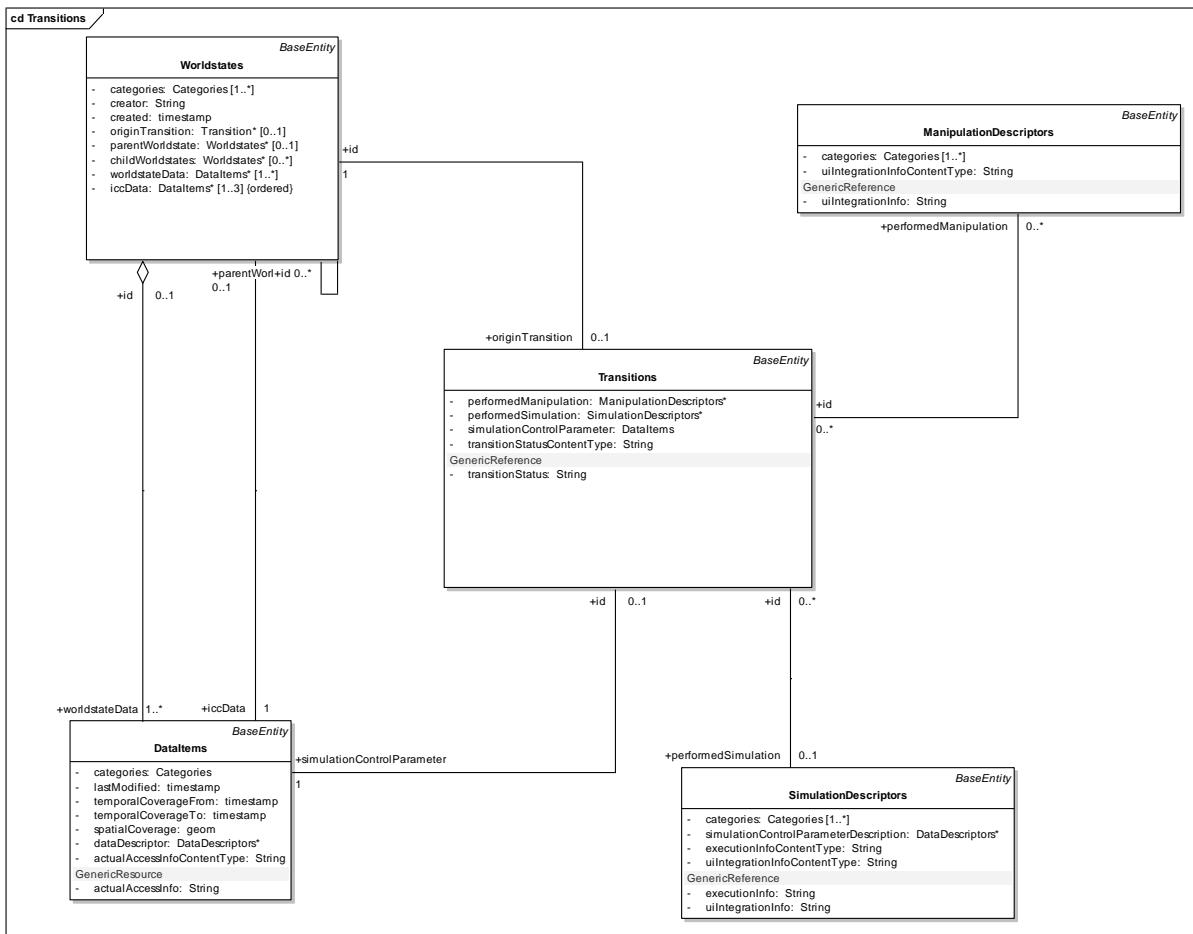
Created By: Pascal Dihé on 20.06.2013

Last Modified: 24.09.2013

Version: 1.0. False

GUID: {B051FEB1-B26B-4f15-B3C6-392C58B66B65}

A transition produces a World State from another World state. The World State Transition maintains the structure of the World State and does neither change the dimension of the world nor the type of elements that constitute the world state (Homomorphism). This transition is either induced by a manipulation of existing World State Data or by the execution of a simulation model (Federated Simulation). The Federated Simulation is parameterized through a set of Simulation Model Control Parameters that are considered part of the Transition.



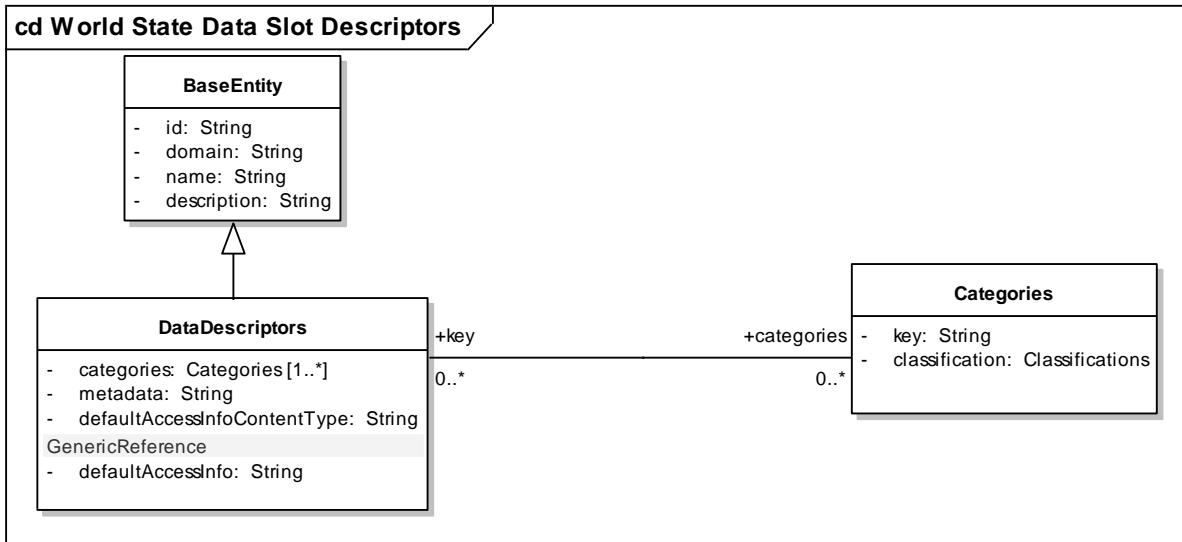
World State Data Slot Descriptors - (Class diagram)

Created By: Pascal Dihé on 19.11.2013

Last Modified: 09.05.2014

Version: 1.0. False

GUID: {A016D1E5-F97C-49c0-BD1B-1DB90D2F6264}



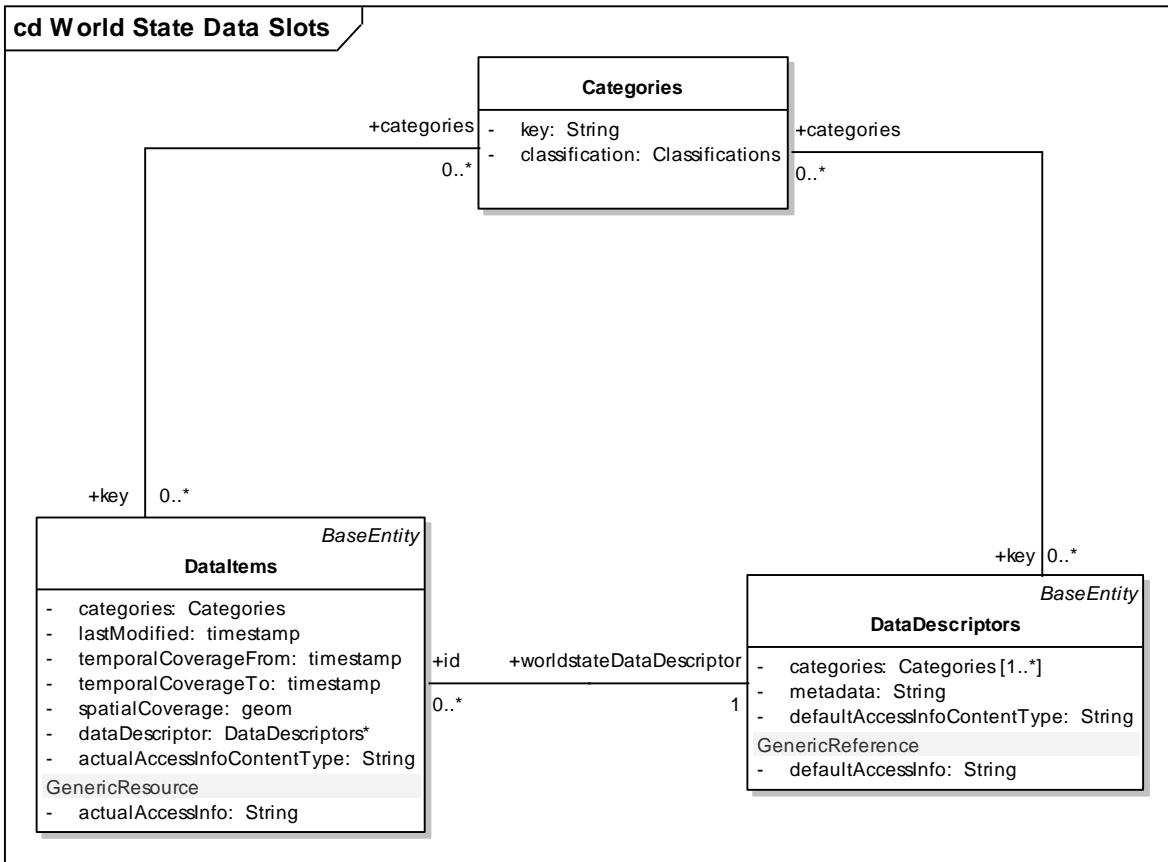
World State Data Slots - (Class diagram)

Created By: Pascal Dihé on 19.11.2013

Last Modified: 09.05.2014

Version: 1.0. False

GUID: {EF520ECF-C2DF-4304-886B-6DF3F96F4951}



Worldstate - (Class diagram)

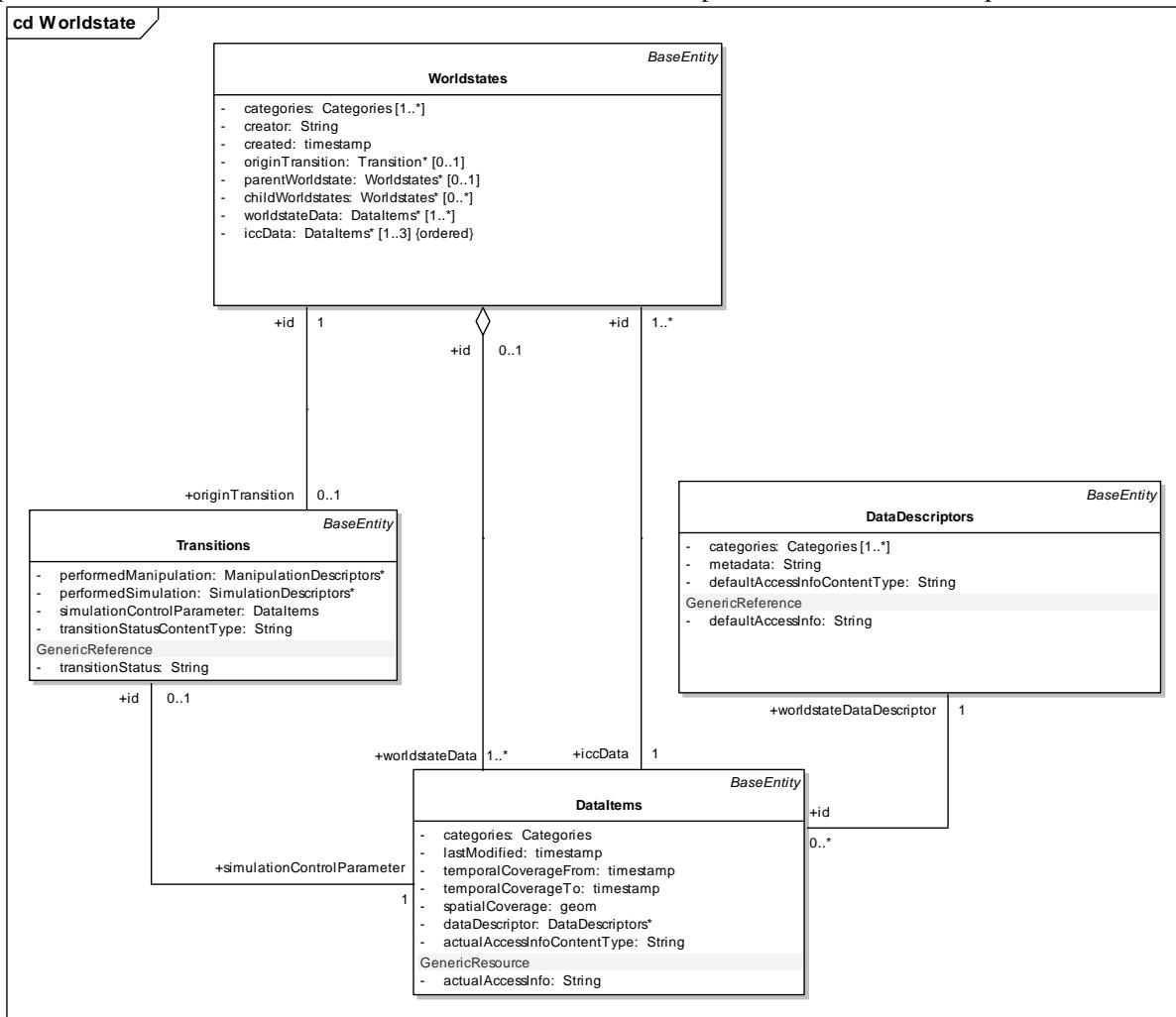
Created By: Pascal Dihé on 18.06.2013

Last Modified: 09.05.2014

Version: 1.0. False

GUID: {074D81D0-1495-4248-B00D-6568483B3D9D}

A snapshot of the World or World State consists of all data related to a specific crisis simulation experiment.



BaseEntity

Type: Class

Status: Proposed. Version 1.0. Phase 1.0.

Package: Core Control and Communication Information Models

Keywords:

Detail: Created on 18.06.2013. Last modified on 04.07.2013.

GUID: {CF28601E-3309-4da9-8B7D-6AD1B1721F0A}

A Base Class that defines common properties needed by all classes. All class should be derived from this class.

Attributes

Attribute	Notes	Constraints and tags
id <u>String</u> Private	Unique id of the entity.	<i>Default:</i>
domain <u>String</u> Private	Name of the domain the entity belongs to.	<i>Default:</i>
name <u>String</u> Private	Optional name of the entity.	<i>Default:</i>
description <u>String</u> Private	Optional description of the entity.	<i>Default:</i>

Categories

Type: Class
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Core Control and Communication Information Models *Keywords:*
 Detail: *Created on 21.06.2013. Last modified on 04.07.2013.*
 GUID: {6EC90ED3-C094-494e-B553-8AE21FD7F0AE}

A class representing an arbitrary category that can be used to categorise scenarios, simulations, etc.

Attributes

Attribute	Notes	Constraints and tags
key <u>String</u> Private	a static key of the category that can be referenced in applications. In contrast to the name property, it will not be changed (e.g. depending on the language of the application).	<i>Default:</i>
classification <u>Classifications</u> Private	Further classification of the category, e.g. scenario types, crisis management phases, etc.	<i>Default:</i>

Classifications

Type: Class
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Core Control and Communication Information Models **Keywords:**
 Detail: *Created on 21.06.2013. Last modified on 04.07.2013.*
 GUID: {EC57392C-FCB0-43c4-90D6-4E5FD5D97AB2}

A class that represents a classification of categories, e.g. scenario types, crisis management phases,

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
key String Private	A static key that can be referenced in applications. In contrast to the name property, it will not be changed (e.g. depending on the language of the application).	<i>Default:</i>

DataDescriptors

Type: **Class BaseEntity**

Status: Proposed. Version 1.0. Phase 1.0.

Package: Core Control and Communication Information Models *Keywords:*

Detail: *Created on 20.06.2013. Last modified on 04.07.2013.*

GUID: {349420FE-0115-4a6a-9A01-041D6396FD6A}

A DataDescriptor provides invariant meta-information about an arbitrary dataset (worldstate data, simulation control parameters, ICC Date). A DataItem is associated with a DataDescriptor. The DataDescriptor contains thereby static information like the URI of a data source (Data Integration Building Block - data access service, e.g. OGC WFS) while the DataItem provides information to identify an individual data item or a specific version of a data item (e.g. a specific request parameter, a filename, etc.). A DataDescriptor can also be considered as describing the type or source of the data and the DataItem as representing a concrete data set of this type.

Attributes

Attribute	Notes	Constraints and tags
categories Categories Private Collection [1..*]	Categories that describe the Data Descriptors, e.g. worldstate data, simulation control parameter data, etc.	<i>Default:</i>
metadata String Private	This field contains either a link to an external INSPIRE metadata document describing the data set (e.g. the URL of an XLS file, a metadata catalogue entry, a website, etc.) or the INSPIRE metadata itself (XML, JSON, TEXT, ...). This is descriptive information about the data source, thus it is not required that the contents of this field are machine-processable.	<i>Default:</i>

Attribute	Notes	Constraints and tags
	Nevertheless, it is recommended to provide the INSPIRE metadata in JSON format so that it can easily be processed by web-based clients.	
defaultAccessInfoContent <u>Type String</u> Private	Content Type of the specific access info for this Data Descriptor.	<i>Default:</i>
defaultAccessInfo <u>String</u> Private GenericReference	A GenericReference that describes the technical access information of actual data source, e.g. URI and credentials (e.g. WFS URI). This is invariant meta-information that stays the same for all DataItems that are described with this DataDescriptor.	<i>Default:</i>

DataItems

Type: Class BaseEntity
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Core Control and Communication Information Models *Keywords:*
 Detail: *Created on 18.06.2013. Last modified on 04.07.2013.*
 GUID: {4F90B8A4-5056-4a06-8D91-0089DF4B183E}

This class represent an actual data set (instance), e.g. worldstate data, icc data, simulation control parameters

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
categories <u>Categories</u> Private	Categories of the DataItem. May be different from the category of the DataDescriptor.	<i>Default:</i>
lastModified <u>timestamp</u> Private	Last modification timestamp of the data item.	<i>Default:</i>
temporalCoverageFrom <u>timestamp</u> Private	The temporal coverage of the data (start date).	<i>Default:</i>
temporalCoverageTo <u>timestamp</u> Private	The temporal coverage of the data (end date).	<i>Default:</i>
spatialCoverage <u>geom</u> Private	The spatial coverage (geometry) of the data.	<i>Default:</i>
dataDescriptor <u>DataDescriptors</u> Private	Reference to a DataDescriptor object containing invariant control and meta-information about the worldstate data, e.g. access information (URI) for a Data Integration Building Block.	<i>Default:</i>

Attribute	Notes	Constraints and tags
actualAccessInfoContent <u>Type String</u> Private	Content Type of the actual access info of the DataItem.	<i>Default:</i>
actualAccessInfo <u>String</u> Private GenericResource	This is the actual information on how to access the data or for very small and simple datasets (e.g. ICC Data) it may contain also the data itself. In general it provides information to identify an individual data item or a specific version of a data item (e.g. a specific request parameter, a filename, etc.). It is also related to a DataDescriptor whereby the DataDescriptor may provide invariant and general access information (base URI, database name, table name, ...) and the actual access information to identify a data set may consist of request parameters, table names, filenames, etc. Like any GenericReference, Actual access information may be described by an Application CCIM.	<i>Default:</i>

ExternalResource

Type: Metaclass
 Status: Proposed. Version 1.0. Phase 1.0.
 Package: Core Control and Communication Information Models Keywords:
 Detail: *Created on 20.06.2013. Last modified on 04.07.2013.*
 GUID: {C65F553C-6859-491a-B28A-9F35EBDBAA8B}

This meta class represents an external resource that is referenced by a GenericReference. The External Resource may be defined by another application specific CCIM. Thus, External Resource is a defined extension point of the Core CCIM. The actual content of the External Resource is described by a contentType property.

GenericReference

Type: Stereotype

Status: Proposed. Version 1.0. Phase 1.0.
Package: Core Control and Communication Information Models **Keywords:**
Detail: *Created on 20.06.2013. Last modified on 04.07.2013.*
GUID: {0A347F7F-A628-4b84-8A91-A11F5B15202D}

This stereotype represents a generic reference to an external resource, e.g. a JSON or XML document, a link to an JSON or XML document, etc. In general domain specific information that may be used by (application specific) components. The content of a generic reference is identified by a content type attribute. Generic References of type CCIM Reference describe a reference to an arbitrary CCIM JSON object (CCIM entity). The CCIM Reference is used to establish relations to CCIM entities that are not known during the design time of the Core CCIM. Generic References provide an extensions point where application specific CCIMs can be plugged-in. CCIM Reference is not an instantiable Class but a stereotype of a string attribute. The value of the attribute contains either a complete CCIM JSON object or a link to a CCIM JSON object (an object with one \$ref property).

IccFunctionDescriptors

Type: Class BaseEntity
Status: Proposed. Version 1.0. Phase 1.0.
Package: Core Control and Communication Information Models **Keywords:**
Detail: *Created on 18.06.2013. Last modified on 04.07.2013.*
GUID: {9218A8A2-684B-440a-9DA4-6F702F5E6505}

IccFunctionDescriptors provide Information about the ICC Functions that are executed after each worldstate transition.

Attributes

Attribute	Notes	Constraints and tags
executionInfoContentTyp <u>e</u> <u>String</u> Private	Content Type of the specific execution info for this IccFunctionDescriptor.	<i>Default:</i>

Attribute	Notes	Constraints and tags
executionInfo <u>String</u> Private GenericReference	Information on how to execute the ICC Function, may include a execute command, a service URL, an API call, etc. The concrete details depend on the types of ICC functions available in a particular CRISMA Application.	<i>Default:</i>

InitialisationDescriptors

Type: **Class** BaseEntity

Status: Proposed. Version 1.0. Phase 1.0.

Package: Core Control and Communication Information Models **Keywords:**

Detail: *Created on 25.06.2013. Last modified on 04.07.2013.*

GUID: {7962B77C-0E06-46d5-A5D6-853821BF1606}

Initialisation descriptors describe user interfaces and business logic (e.g. Composite UI Modules, Setup and Configuration Building Blocks) that are needed to initialize the set of simulation models, the initial worldstate, descriptors, etc.. This refers to the initial setup and configuration of the CRISMA Application.

Attributes

Attribute	Notes	Constraints and tags
categories <u>Categories</u> Private Collection [1..*]	Categories that describe the Initialisation Descriptors.	<i>Default:</i>
uiIntegrationInfoContent Type <u>String</u> Private	Content Type of the specific ui integration info for this Initialisation Descriptor.	<i>Default:</i>

Attribute	Notes	Constraints and tags
uiIntegrationInfo <u>String</u> Private GenericReference	This is the actual information on how to instantiate and integrate the respective initialisation user interface (Composite UI Module), depending on the CRISMA Application, this field may contain a URI that point to a Composite UI Module instance..	<i>Default:</i>

ManipulationDescriptors

Type: **Class** BaseEntity

Status: Proposed. Version 1.0. Phase 1.0.

Package: Core Control and Communication Information Models

Keywords:

Detail: *Created on 18.06.2013. Last modified on 04.07.2013.*

GUID: {453E5592-C89B-488a-998C-6124BC31381D}

ManipulationDescriptors refer to worldstate transition and describe the user interfaces and business logic (e.g. Composite UI Modules, Worldstates Widget Building Block and associated Worldstate Views) to manipulate the changeable parameters of a worldstate.

Attributes

Attribute	Notes	Constraints and tags
categories <u>Categories</u> Private Collection [1..*]	Categories that describe the Manipulation Descriptor.	<i>Default:</i>
uiIntegrationInfoContent <u>Type</u> <u>String</u> Private	Content Type of the specific ui integration info for this Manipulation Descriptor.	<i>Default:</i>

Attribute	Notes	Constraints and tags
uiIntegrationInfo <u>String</u> Private GenericReference	This is the actual information on how to instantiate and integrate the respective manipulation user interface (Composite UI Module). Depending on the CRISMA Application, this field may contain a URI that point to a Composite UI Module instance which is able to manipulate worldstate data thus leading to a new worldstate (worldstate transition).	<i>Default:</i>

RenderingDescriptors

Type: **Class** BaseEntity

Status: Proposed. Version 1.0. Phase 1.0.

Package: Core Control and Communication Information Models *Keywords:*

Detail: *Created on 24.06.2013. Last modified on 04.07.2013.*

GUID: {B8DBD495-C169-4b13-A6FD-AC8E4CE93822}

A Rendering Descriptor describes a User interfaces to visualize the worldstate (Worldstates Widget Building Block and related Viewer Widgets).

Attributes

Attribute	Notes	Constraints and tags
categories <u>Categories</u> Private Collection [1..*]	Categories that describe the Rendering Descriptors	<i>Default:</i>
uiIntegrationInfoContent <u>Type</u> <u>String</u> Private	Content Type of the ui integration info the Rendering Descriptor.	<i>Default:</i>

Attribute	Notes	Constraints and tags
uiIntegrationInfo <u>String</u> Private GenericReference	This is the actual information on how to instantiate and integrate the respective (worldstate) user interface (Composite UI Module, Worldstates Widget Building Block). Depending on the CRISMA Application, this field may contain a URI that point to a Composite UI Module.	<i>Default:</i>

SimulationDescriptors

Type: **Class** BaseEntity

Status: Proposed. Version 1.0. Phase 1.0.

Package: Core Control and Communication Information Models **Keywords:**

Detail: *Created on 18.06.2013. Last modified on 04.07.2013.*

GUID: {6D275E96-AE09-46db-861E-0C2D54C99EC7}

A SimulationDescriptors provides information about a Federated Simulation (a simulation consisting of one or more simulation models wrapped as one WPS service). It holds references (Generic Reference) to concrete descriptions on how to execute the simulation (Simulation Model Integration Building Block) and by which user interface (Simulation Model Interaction Building Block) to control (e.g. parameterise, monitor) the simulation.

Attributes

Attribute	Notes	Constraints and tags
categories <u>Categories</u> Private Collection [1..*]	Categories that describe the Simulation Descriptors, e.g. the type of the simulation (impact scenario, ...)	<i>Default:</i>
simulationControlParam eterDescription <u>DataDescriptors</u> Private	A description of the simulation model control parameters.	<i>Default:</i>

Attribute	Notes	Constraints and tags
executionInfoContentTyp e <u>String</u> Private	Content Type of the specific execution info for this Simulation Descriptor.	<i>Default:</i>
uiIntegrationInfoContent Type <u>String</u> Private	Content Type of the specific ui integration info for this Simulation Descriptor.	<i>Default:</i>
executionInfo <u>String</u> Private GenericReference	Information on how to execute the federated simulation. In general it will contain information on the WPS that provides the simulation (Simulation Model Integration Building Block). The concrete details depend on the types of simulations available in a particular CRISMA Application.	<i>Default:</i>
uiIntegrationInfo <u>String</u> Private GenericReference	This is the actual information on how to instantiate and integrate the respective simulation control user interface (Composite UI Module, Simulation Model Interaction Building Block). Depending on the CRISMA Application, this field may contain a URI that point to a Composite UI Module instance which is able to configure a simulation, monitor its progress, etc..	<i>Default:</i>

Transitions

Type: Class **BaseEntity**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Core Control and Communication Information Models **Keywords:**
Detail: *Created on 20.06.2013. Last modified on 04.07.2013.*
GUID: {7B609E89-8DCA-48e4-8B07-C9103C0CF0C7}

This class represents a concrete worldstate transition. It contains information on the performed manipulation or simulation, the actual simulation control parameters and further meta-information (e.g. which parts of the worldstate were changed).

Attributes

Attribute	Notes	Constraints and tags
performedManipulation <u>ManipulationDescriptors</u> Private	This property will contain about the type of manipulations that have been performed.	<i>Default:</i>
performedSimulation <u>SimulationDescriptors</u> Private	This property will contain about the type of simulation that has been performed.	<i>Default:</i>
simulationControlParameter <u>DataItems</u> Private	This property will contain the actual simulation control parameter data.	<i>Default:</i>
transitionStatusContentType <u>String</u> Private	Content Type of the specific transition status of this transition.	<i>Default:</i>
transitionStatus <u>String</u> Private GenericReference	This Generic Reference provides the detailed transition status information. It may be represented by an arbitrary CRISMA JSON object that is defined for a specific Application CCIM and may be injected dynamically, e.g. by the actual simulation execution building block. It may also point to the raw transition status, e.g. a logfile, a link to a website with status messages, a WPS url, etc.	<i>Default:</i>

Worldstates

Type:

Class **BaseEntity**

Status:

Proposed. Version 1.0. Phase 1.0.

Package:

Core Control and Communication Information Models *Keywords:*

Detail: Created on 18.06.2013. Last modified on 04.07.2013.

GUID: {ED2315DD-4023-4bd5-B228-A431A73193AE}

This class represent a concrete worldstate. A snapshot of the World or World State consists of all data related to a specific crisis simulation experiment.

Attributes

Attribute	Notes	Constraints and tags
categories <u>Categories</u> Private Collection [1..*]	Categories of the worldstate, e.g. to relate the worldstate to a specific scenario or a crisis management phase.	<i>Default:</i>
creator <u>String</u> Private	User or process that created the worldstate.	<i>Default:</i>
created <u>timestamp</u> Private	Creation Timestamp of the worldstate.	<i>Default:</i>
originTransition <u>Transition</u> Private [0..1]	Reference to the transition that created the worldstate, e.g. a manipulation or a simulation.	<i>Default:</i>

Attribute	Notes	Constraints and tags
parentWorldstate <u>Worldstates</u> Private [0..1]	Reference to the parent worldstate in the worldstates tree.	<i>Default:</i>
childWorldstates <u>Worldstates</u> Private Collection [0..*]	Reference to all child worldstates of this worldstate.	<i>Default:</i>
worldstateData <u>DataItems</u> Private Collection [1..*]	An array of data items that represent the actual worldstate data.	<i>Default:</i>
iccData <u>DataItems</u> Private [1..3]	A three-dimensional array of data items that represents the Indicator, Criteria and Cost data that has been produced by a Indicator Function (in case of Indicatior data) or other services (in case of Cost and Criteria data).	<i>Default:</i>