WP 3 Project Deliverable D3.1

Specification of Geometrical Database

Project Number	IST-2000-29266
Project Title	Virtual Real Time Fire Emergency Simulator
Deliverable Type	Report
Deliverable Class	Public

Deliverable Number	D3.1
Title of Deliverable	Specification of geometrical database [WP3.1]
Nature of the Deliverable	Report
Contributing WPs	WP 2, WP 3
Contractual Date of Delivery	30. March 2002
Actual Date of Delivery	20. Sept 2002
URL	www.virtualfires.org
Authors	Gunther Lenz (SiTu), Thomas Reichl (SiTu)
Contact Details	Institute for Structural Analysis / SiTu Research Univ. Prof. Dipl.-Ing. Dr. techn. Gernot Beer Lessingstrasse 25/II 8010 Graz / Austria Tel.: +43 316 8736180 Fax: +43 316 8736185 Email: gernot.beer@ifb.tu-graz.ac.at

Abstract	WP3 Task 1 is responsible for the specification of the Geometrical-Database. The results of this workpackage are summarised in this report and define how the geometrical data. will be stored and transferred between CFD-calculation, Database and VR-System
Keywords	Specification, Database, Geometry

Contents

1 Overview

The database management for the VFS will provide the services for the storage and retrieval of the required data for the simulation environment. This contains data for CFD-Calculation, all data for the simulation in the VR-System, data for the definition of missions etc. WP3 Task 1 is responsible for the specification of the Database regarding geometrical information. The results of this workpackage are summarised in this report and define, how the geometrical information will be stored. The delay of delivering this report accrues, because report D2.4 [VF1] has to be taken into account, which was closed on 19th of August 2002. The specification of the geometrical (D3.1) and the CFD database (D3.2) are the basis for the implementation of the database. Some additional specifications about management of simulations will be defined separately and summarised in report D3.4.

2 Abbreviations

VFS ..... Virtual Fires Simulator

CFD ..... Computational Fluid Dynamics

VR ..... Virtual Reality

DM ..... Data Management

BLOB ..... Binary Large Object

OBJ ..... Wavefront Object

OFF ..... Object File Format

3 Requirements

3.1 Functional Requirements

The geometrical data file format supported by the VFS must allow a simple import of 3D models provided by an external modeler.

The geometrical data file format delivered from the database to the VR subsystem must be supported the VR systems used within VFS, i.e. Covise from Vircinity AG and the OpenSG library and their extensions provided by FIGD.

The geometrical database must support the planned relations between the objects of the simulator, which were defined in the analysis model of the database content.

3.2 Nonfunctional Requirements

The geometrical data file format must be well documented to allow modification within the VFS.

4 Analysis Model of the Content of the VFS Database

Based on the basic system specification in the D2.4 report [VF1], the suggestions at WP3 meeting at Stockholm and the following discussion at virtualfires.org the layout of the database of the VFS will be as follows.

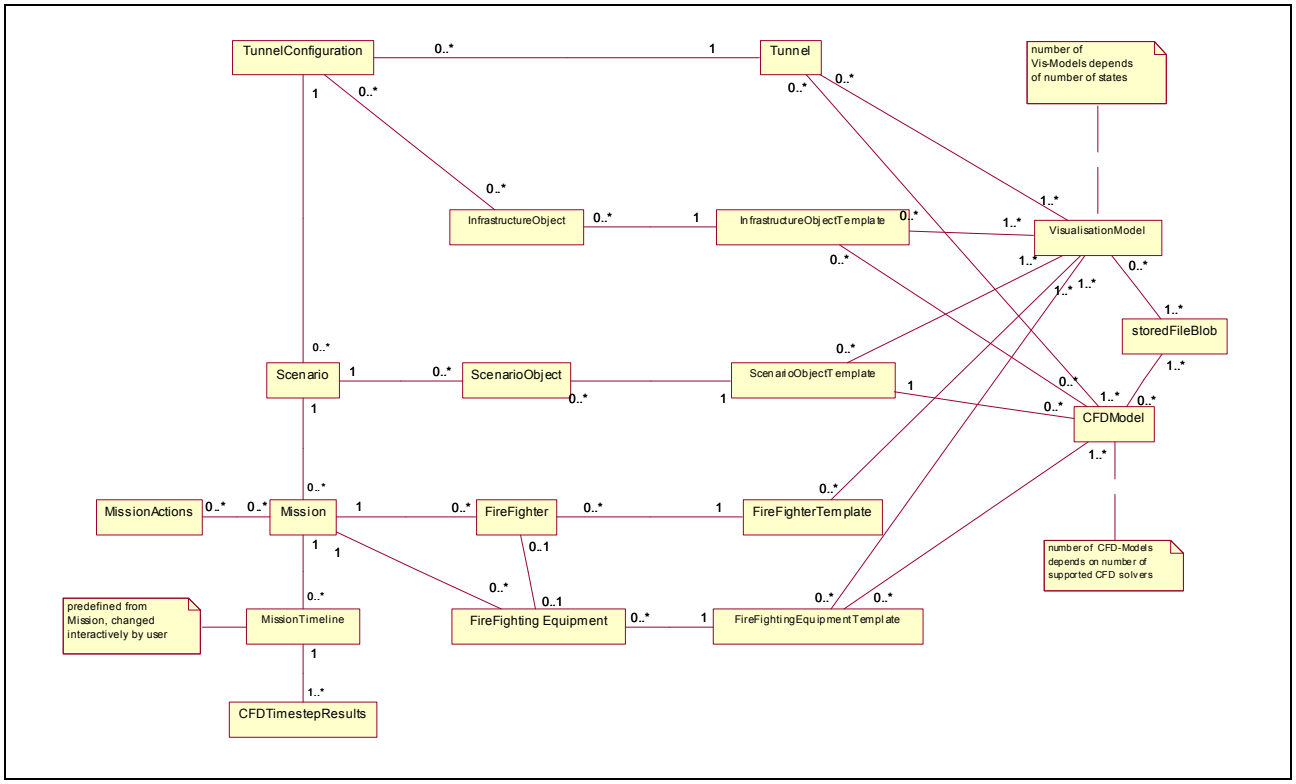

Figure 1: Analysis model of the VFS DB

This enables an user of the simulator to separate the configuration of simulation into the following parts:

- setup a tunnel
  This is the basic step for defining a tunnel for the use within VFS. It oncorporates the selection of one or more visualisation models that build the shape of the tunnel. Splitting the visualisation model from one into several sections allows to give a visual feedback of the degradation of the tunnel at section-level.

- configure a tunnel
  This defines the infrastructure, e.g. traffic signs, ventilation fans etc., within a tunnel. For a given tunnel there are multiple configurations possible.

- setup a scenario
  A scenario defines the arrangement of objects within a tunnel on which a mission should act. E.g. this may be the setup of a car-crash inside the tunnel. The scenario takes place within a certain tunnel configuration.

- define a mission
  Based on a given scenario a mission defines the timely order of actions and commands to the objects for the simulation. Within this part a firefighter mission leader would plan its firefighting mission or a tunnel operator would test his emergency plan.

4.1 Description of the Objects

Object	Description
Tunnel	A Tunnel object contains the references to the model of the tunnel for the visualisation and the model data for CFD-calculation, i.e. the computational domain. Additional infformation: name of the tunnel, short description of the tunnel.
TunnelConfiguration	A TunnelConfiguration defines a whole tunnel with its infrastructure, like ventilation fans, traffic signs,…
Scenario	A Scenario defines the setting on which missions take place, e.g. a car crash inside the tunnel. It contains the involved ScenarioObjects, the environmental conditions of the tunnel, the location and starting time of the fire.
Mission	A Mission contains the chronological list of planned actions and its dedicated resources of manpower and equipment.
InfrastructureObjectTemplate	All Objects installed in a tunnel that have bearing on the CFD-Calculation. E.g. Ventilation, fixed fire fighting equipment.
ScenarioObjectTemplate	The description of the state of the tunnel and its environment at the beginning of the simulation of a scenario.
FireFightingEquipmentTemplate	The template for a piece of firefighting equipment. It contains the models for visualisation, a list of changeable parameters and the description of its influence to the simulation.
FireFighterTemplate	The template for a firefighter. It contains the models for visualisation, a list of changeable parameters and the description of its influence to the simulation.
CFDModel	Contains the mesh and boundary condition data for each CFD-Solver. According to the WP 2.4 report there will be one for FLUENT and one for ICE.
VisualisationModel	Contains the model for VR-System that includes all objects necessary for the visualisation
InfrastructureObject	Objects that must be visualised but do not have any influence on the CFD-calculation. E.g. traffic signs, lights, construction equipment.
ScenarioObject	The description of the state of the tunnel and its environment at the beginning of the simulation of a scenario.
FireFightingEquipment	A piece of firefighting equipment and its actual parameters. Its definition is derived from the according FireFightingEquipmentTemplate.

FireFighter	A firefighter and its actual parameters. Its definition is derived from the according FireFighterTemplate.
MissionAction	A defined action on an object within a mission, e.g. "turn on fan xy at tz"
MissionTimeline	Describes the timely change of a firefighting missions, e.g. the way of the firefighters or the activation time and duration of the nozzles. Initially there is just one single timeline defined by the mission and additional ones are added by inserting branching points, which yields to a treelike structure.
CFDTimestepResults	The results for a given timestep on a timeline within a mission. Contains also the state and changed boundary condition values for the branching points.

5 Specification

5.1 Objects related to Geometrical Information

Withins this report only those database entries are concerned which are related to geometrical data.

This reduces the scope of the document to the following objects from the analysis model:

- VisualisationModel
- Tunnel
- TunnelConfiguration
- Scenario
- Mission
- InfrastructureObjectTemplate
- ScenarioObjectTemplate
- FireFighterTemplate
- FireFightingEquipmentTemplate
- storedFileBlob

5.2 Data Format for Geometrical Representation

Based on the functional requirements it was decided to use the Wavefront OBJ file format for the storage of all geometrical models.

5.2.1 Description of OBJ file format

Object files define the geometry and other properties for objects. Object files can be in ASCII format (.obj) or binary format (.mod). Release 3.0 supports both polygonal objects and free-form objects. Polygonal geometry uses points, lines, and faces to define objects while free-form geometry uses curves and surfaces. In the development of the first prototype VFS will use object-files in ASCII format to guarantee the best transfer portability. Specification of Wavefront Object files can be found in [OBJ1]. OBJ-files do not include textures. They are defined in the corresponding MTL-file described below.

5.2.2 Description of the MTL file format

MTL-file is used to define textures for OBJ-files. A MTL-file does not contain the texture itself but it defines the name and how the texture will be used.

5.2.3 Constraints for the use of OBJ within the simulator

To assure the correct location of textures all references to texturefiles or MTL-files must not contain any path information. I.e. all texture files must be located in the same directory as the model files.

5.3 State Transition due to Temperature Exposition

The objects within the simulation will change in the visual appearance due to their exposition to heat. To provide this functionality with little computational effort the the following approach to model this behavior was chosen.

5.3.1 Modeling of "Burned State"-Dependency

As it is neither possible to determine the exact level of burn-up of an object nor to describe its geometry as a function of degradation with low computational effort it was decided to base the state of the object simply on its reached maximum temperature and to provide different geometrical models for these states.

As shown in Figure 2 the state changes according to the exceeding of given threshold values for the exposed temperature.

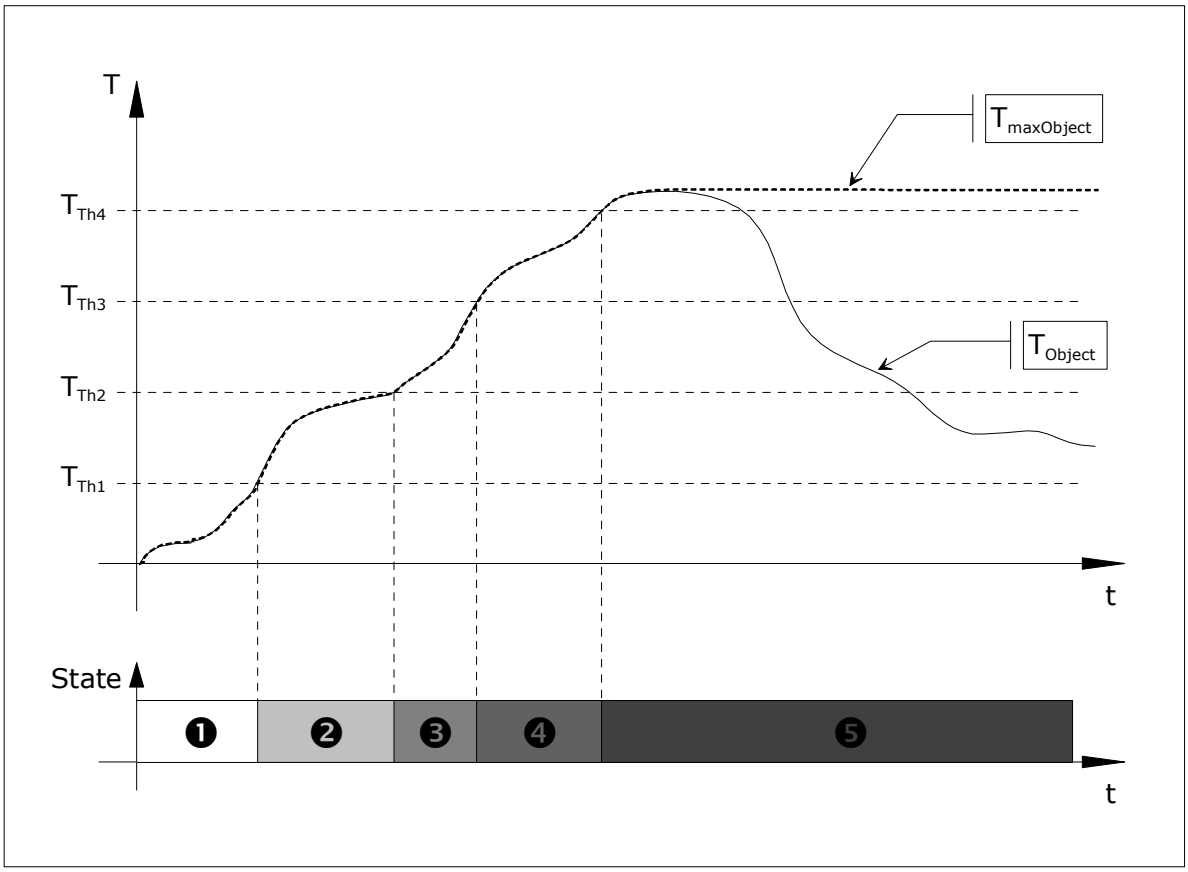

Figure 2: Interconnection between state and reached temperature

5.3.2 Data for Modeling the State Transition

As the states are irreversible (a burnt car won't become good again from simply cooling it down) the transitions form a directed graph which simply degrades to a forward linked list in this case.

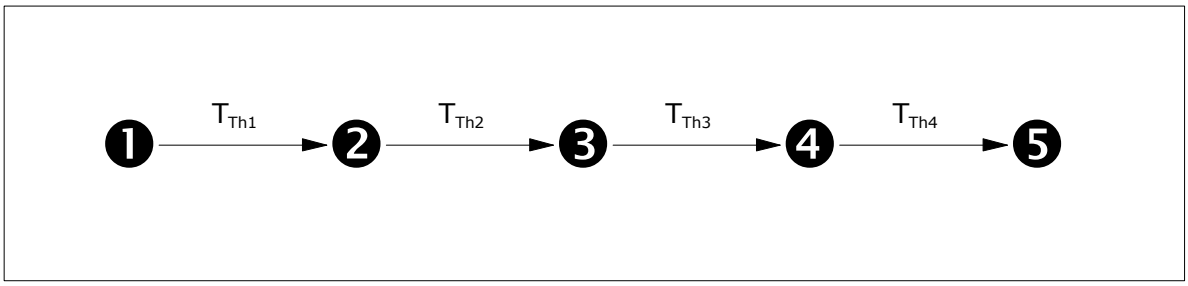

Figure 3: State graph

This can be modeled as an entry who contains the reference ID of his successor, like shown in Figure 4.

Additionally each state description includes the required threshold temperature and reference to the associated visualisation model, that provides the visual feedback for this state.

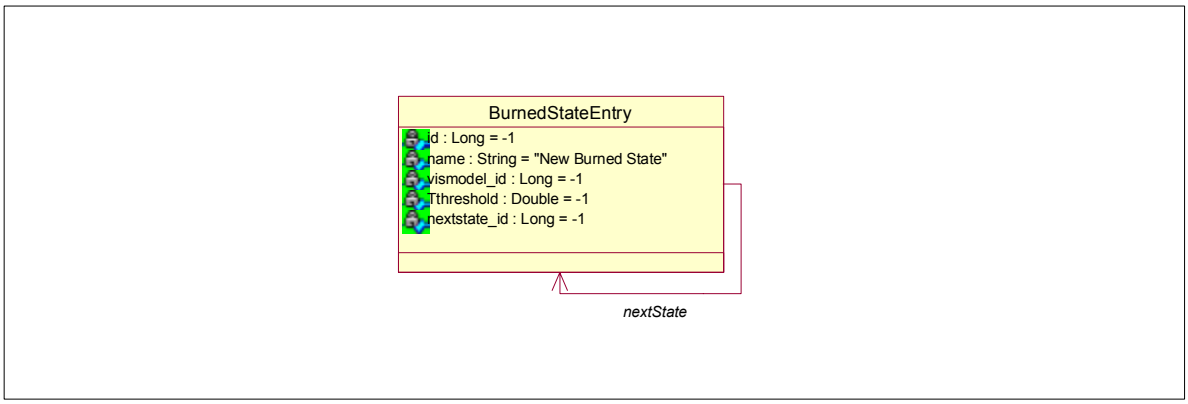

Figure 4: Model for the burned state description

5.4 Design Model for the Geometrical Database

To realize the planned relations between the geometrical objects within the VFS the following design model was developed.

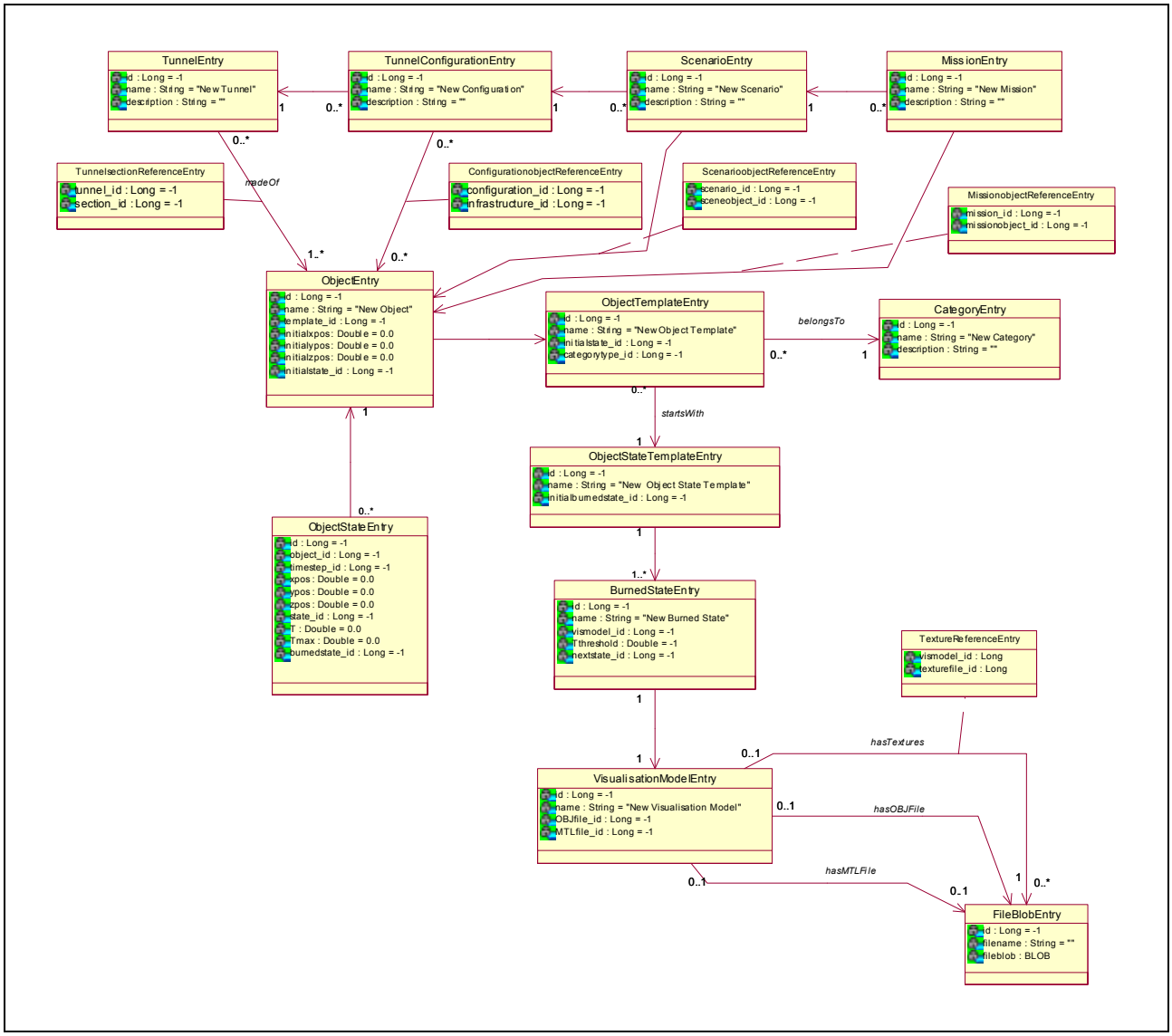

Figure 5: Design model for the database content (geometry related part)

5.5 Mapping from Analysis Model to Design Model

The connection between the objects described in the analysis model and those within the design model is given in the table below.

Analysis Model	*Design Model*
VisualisationModel	As the VisualisationModel encapsulates the set of OBJ-, MTL- and texturefiles its mapping corresponds to multiple entries: For a VisualisationModel a corresponding VisualisationModelEntry is created. For every file that belongs to this model a FileBlobEntry is created that stores the content of the file as binary data. The id of the OBJ- and MTL- FileBlobEntries are stored within the VisualisationModelEntry For every texturefile a TexturefileReferenceEntry is generated which holds the id of the textures FileBlobEntry and the id of the VisualisationModelEntry.
Tunnel	A Tunnel is represented by a TunnelEntry. For each section of a tunnel a new TunnelsectionReferenceEntry is created and the IDs of the tunnel and the associated object are stored.
TunnelConfiguration	A TunnelConfiguration is represented by a TunnelConfigurationEntry. For each associated infrastructure object a new ConfigurationobjectReferenceEntry is created and the associated IDs of the configuration and the object are stored.
Scenario	A Scenario is represented by a ScenarioEntry. For each placed object within a scenario a new ScenarioobjectReferenceEntry is created and the associated IDs of the scenario and the object are stored.
Mission	A Mission is represented by a MissionEntry. For each object involved in a mission a new MissionobjectReferenceEntry is created and the associated IDs of the mission and the object are stored.
InfrastructureObjectTemplate	All these map to a ObjectTemplateEntry. The distinction to which kind of template object an ObjectTemplateEntry belongs is made by its category_id.
ScenarioObjectTemplate	
FireFighterTemplate	
FireFightingEquipmentTemplate	

InfrastructureObject	An InfrastructureObject maps to an ObjectEntry, whose template belongs to the category "Infrastructure".
ScenarioObject	An ScenarioObject maps to an ObjectEntry, whose template belongs to the category "Scenario".
FireFightingEquipment	An FireFightingEquipment maps to an ObjectEntry, whose template belongs to the category "Firefighting Equipment".
FireFighter	An FireFighter maps to an ObjectEntry, whose template belongs to the category "Firefighter".
storedFileBlob	A storedFileBlob maps directly to a FileBlobEntry, who contains the filename and the content of a file as binary data.

5.6 Data Dictionary for the Geometrical Database

5.6.1 BurnedStateEntry

This entry describes the state of an object according to the maximum temperature it has reached.

5.6.1.1 Attributes

Name	Type	Description
id	Long	The ID of the entry for referencing it within the database. The value of the id is unique within the table.
name	String	A userfriendly name of the burned state of an object. The name is unique within the table to allow exact querying by name.
vismodel_id	Long	The ID of the VisualisationModelEntry that belongs to this state.
Tthreshold	Double	The threshold temperature for changing to the next state. Unit: °C This value must only be considered when the nextstate_id has a value other than -1.
nextstate_id	Long	The current state of the object transits to this state, when the temperature of the object reaches the threshold level. Has a value of -1, when the current state is the final one.

5.6.2 CategoryEntry

This entry describes the main category to which a template object belongs.

5.6.2.1 Attributes

Name	Type	Description
id	Long	The ID of the entry for referencing it within the database. The value of the id is unique within the table.
name	String	A userfriendly name of the category. The name is unique within the table to allow exact querying by name.
description	String	A short description of the purpose of this category. E.g. "Into this category belong all

		objects that are part of the infrastructure of a tunnel, like traffic signs, emergency phones etc."

5.6.3 ConfigurationobjectReferenceEntry

This entry is for realizing the one-to-many relationship between the tunnelconfiguration and its infrastructure objects.

5.6.3.1 Attributes

Name	Type	Description
configuration_id	Long	The ID of the configuration to which the object referenced via infrastructure_id belongs.
infrastructure_id	Long	The ID of the infrastructure object that belongs to the configuration referenced via configuration_id.

5.6.4 FileBlobEntry

This entry describes a stored binary file within the database.

5.6.4.1 Attributes

Name	Type	Description
id	Long	The ID of the entry for referencing it within the database. The value of the id is unique within the table.
filename	String	The filename of the stored file. There is no additional path information included.
fileblob	BLOB	The content of the stored file as a single binary object.

5.6.5 MissionEntry

This entry contains the necessary information about a mission.

5.6.5.1 Attributes

Name	Type	Description

id	Long	The ID of the entry for referencing it within the database. The value of the id is unique within the table.
name	String	A userfriendly name of the mission. The name is unique within the table to allow exact querying by name.
description	String	A short description of the purpose of the mission. E.g. "Cool down the fire."

5.6.6 MissionobjectReferenceEntry

This entry is for realizing the one-to-many relationship between the mission and its incorporated objects.

5.6.6.1 Attributes

Name	Type	Description
mission_id	Long	The ID of the mission to which the object referenced via missionobject_id belongs.
missionobject_id	Long	The ID of the object that belongs to the mission referenced via mission_id.

5.6.7 ObjectEntry

An actual object from a simulation.

5.6.7.1 Attributes

Name	Type	Description
id	Long	The ID of the entry for referencing it within the database. The value of the id is unique within the table.
name	String	A userfriendly name of the object. The name is unique within the table to allow exact querying by name.
template_id	Long	The ID of the template from which the object is derived.
initialxpos	Double	Starting position of the object along the global x-axis.

initialypos	Double	Starting position of the object along the global y-axis.
initialzpos	Double	Starting position of the object along the global z-axis.
initialstate_id	Long	Initial state of the object.

5.6.8 ObjectStateEntry

This entry contains the information about the state of an object at a specific point in time within a simulation.

5.6.8.1 Attributes

Name	Type	Description
id	Long	The ID of the entry for referencing it within the database. The value of the id is unique within the table.
object_id	Long	The ID of the object to whom this state belongs.
timestep_id	Long	The ID of the timestep to whom the state of the object corresponds.
xpos	Double	Current position along the global x-axis.
ypos	Double	Current position along the global y-axis.
zpos	Double	Current position along the global z-axis.
state_id	Long	ID of the current state of the object.
T	Double	Current temperature value of the object.
Tmax	Double	Already reached maximum temperature value of the object.
burnedstate_id	Long	ID of the current burned state of the object.

5.6.9 ObjectStateTemplateEntry

This entry describes a state that an object may adopt during the simulation. It also serves as the entry point to the chain of burned states for this object state.

5.6.9.1 Attributes

Name	Type	Description
id	Long	
name	String	A userfriendly name of the template state. The name is unique within the table to allow exact querying by name.
initialburnedstate_id	Long	The ID of the starting burned state of the object. Normally points to the "unburnt" state.

5.6.10 ObjectTemplateEntry

This entry describes a template for a simulation object.

5.6.10.1 Attributes

Name	Type	Description
id	Long	The ID of the entry for referencing it within the database. The value of the id is unique within the table.
name	String	A userfriendly name of the object template. The name is unique within the table to allow exact querying by name.
initialstate_id	Long	The ID of the starting state for this template.
categorytype_id	Long	The ID of the category to which this template belongs.

5.6.11 ScenarioEntry

This entry describes a scenario within a tunnel.

5.6.11.1 Attributes

Name	Type	Description
id	Long	The ID of the entry for referencing it within the database. The value of the id is unique within the table.
name	String	A userfriendly name of the scenario. The name is unique within the table to allow exact

		is unique within the table to allow exact querying by name.
description	String	A short description of the scenario.

5.6.12 ScenarioobjectReferenceEntry

This entry is for realizing the one-to-many relationship between the scenario and its contained objects.

5.6.12.1 Attributes

Name	Type	Description
scenario_id	Long	The ID of the scenario to which the object referenced via sceneobject_id belongs.
sceneobject_id	Long	The ID of the scenario object that belongs to the scenario referenced via scenario_id.

5.6.13 TextureReferenceEntry

This entry is for realizing the one-to-many relationship between the visualisation model and its texture files.

5.6.13.1 Attributes

Name	Type	Description
vismodel_id	Long	This is the ID of the VisualisationModel to whom the texture belongs.
texturefile_id	Long	The ID of the texture file.

5.6.14 TunnelConfigurationEntry

This entry describes the configuration of a tunnel.

5.6.14.1 Attributes

Name	Type	Description
id	Long	The ID of the entry for referencing it within the database. The value of the id is unique within the table.
name	String	A userfriendly name of the configuration. The name is unique within the table to allow exact

		querying by name.
description	String	A short description of the configuration.

5.6.15 TunnelEntry

This entry contains all the information about a tunnel that is needed for a simulation of it within VFS.

5.6.15.1 Attributes

Name	Type	Description
id	Long	The ID of the entry for referencing it within the database. The value of the id is unique within the table.
name	String	A userfriendly name of the tunnel. The name is unique within the table to allow exact querying by name.
description	String	A short description of the purpose of the template.

5.6.16 TunnelsectionReferenceEntry

This entry is for realizing the one-to-many relationship between the tunnel and its sections.

5.6.16.1 Attributes

Name	Type	Description
tunnel_id	Long	The ID of the tunnel to which the object referenced via section_id belongs.
section_id	Long	The ID of the section object that belongs to the tunnel referenced via tunnel_id.

5.6.17 VisualisationModelEntry

This entry describes a model for the visualisation including its textures.

5.6.17.1 Attributes

Name	Type	Description

id	Long	The ID of the entry for referencing it within the database. The value of the id is unique within the table.
name	String	A userfriendly name of the visualisation model. The name is unique within the table to allow exact querying by name.
OBJfile_id	Long	The ID of the OBJ file that belongs to this model.
MTLfile_id	Long	The ID of the MTL file that belongs to this model. -1 if there is none.

6 Final remarks

The definition of the interfaces between CDF-Calculation, DM and VR-System are subjected to change within the prototype development and will be summarised in the report for D3.4.

These specifications will be revised and adapted during the development of the VFS.

7 Literature/Links

[VF1] WP 2.2+2.4 Project Deliverable: Selection of developer tools, Specification of planned system capabilities (software)

[VF2] WP 2.6 Project Deliverable: Review of existing VR systems, Adaptability to VIRTUALFIRES

[OBJ1] ftp://avalon.viewpoint.com/avalon/format_specs/obj_spec_format