**GIV – Interlis2 Tool**

|  |  |
| --- | --- |
| Dokument: | High Level Design (HLD) |
| Zusammenfassung: | Das Interlis2 Tool vom GIS Verbund Thurgau konvertiert Interlis2-Datendateien von verschiedenen Quell-Modellen in Ziel-Modellen.  Das Tool wird öffentlich als Open Source Projekt zur Verfügung gestellt. |

Change-Log:

|  |  |
| --- | --- |
| 16.09.2021 (USO) | Erstellen der Dokumentation. |
|  |  |

# Allgemein

## Zweck

Dieses Dokument beinhaltet die Anforderungen, Software-Architektur sowie das Software-Design für das Tool «GIV Interlis2 Tools».

## Gültigkeit

Dieses Dokument ist für die gesamte Lebensdauer des Projektes gültig und wird laufend erweitert und angepasst.

## Bezeichnungen und Namespace

Projekt Name: GIV.Interlis2.Tools

Produkt Name: GIV Interlis2 Tools

Assembly-Name: GIV Interlis2 Tools.exe

Namespace: GIV.Interlis2.Tools.\*

Applikations-Titel: GIV-TG – Interlis2 Tool

Deployment: EXE (portable)

Installationspfad: -

Startmenu: -

Konfigurationsdatei: -

## Abkürzungen

DSS Datenstruktur Siedlungsentwässerung

MEL Meliorationen

GEP Generelle Entwässerungsplanung

# Anforderungen (Übersicht)

In diesem Abschnitt werden die essenziellen Anforderungen dokumentiert.

|  |  |  |
| --- | --- | --- |
| ID: | Bezeichnung: | Datum/Owner: |
| RE001 | Konvertieren vom Modell DSS in das Modell GEP | 27.01.2021 TK Werke |
| RE002 | Konvertieren vom Modell DSS in das Modell MEL | 27.01.2021 TK Werke |
| RE003 | Aus dem Datenbestand DSS den Datenbestand MEL extrahieren | 27.01.2021 TK Werke |

# FU001: Modelle Konvertierungen

Diese Funktion wird für die Konvertierung von einem Quell-Modell zu einem Ziel-Modell verwendet.

## Abdeckung Anforderungen

RE001: Konvertieren vom Modell DSS in das Modell GEP

RE002: Konvertieren vom Modell DSS in das Modell MEL

## Grundsätzliche Funktionsweise

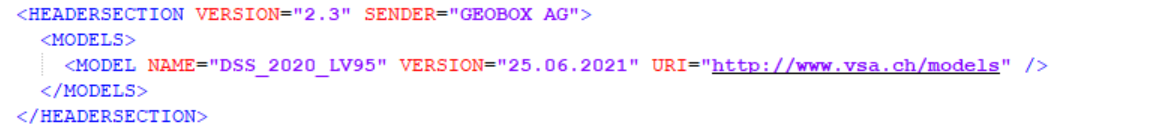
Die Konvertierung verwendet lediglich die Daten Dateien (XTF) und ignoriert die entsprechenden Modelle (ILI).

Es werden keine Prüfungen bezüglich Korrektheit oder Modell-Konformität gemacht.

Die Konvertierung basiert auf einer einfachen Text Ersetzung (String.Repalce)

## Model-Node

Im XTF in der «HEADERSECTION» ist die Angabe des «MODEL» gespeichert.



Diese Angaben wird überschrieben mit den Angaben des Ziel-Modelles, wenn der Model-Node für das Quell-Modell vorhanden ist.

Die Angaben für das Ziel-Modell sind im jeweiligen Kontroller hart codiert.

## Daten Nodes

Bei der Konvertierung werden lediglich die XML-Tags von DSS auf GEP oder MEL umgeschrieben.

Konkret wird der Namespace des XML-Tags umgeschrieben mit einer Textersetzung.

Beispiel:

String.Replace(«DSS\_2020\_LV95», «DSS\_2020\_LV95\_GEP»)

# FU002: Meliorationsdaten extrahieren

Diese Funktion filtert Datenbestände aus einer Quell-Datendatei und speichert die gefilterten Daten in eine neue Datendatei (Ziel-Datendatei).

## Abdeckung Anforderungen

RE003: Aus dem Datenbestand DSS den Datenbestand MEL extrahieren

## Grundsätzliche Funktionsweise

Die Auftrennung erfolgt mittels Filterangaben (Funktionen-Melioration) und den verbundenen Elementen.

Die Verbundenen Elementen werden mit den Referenzattributen entsprechend gesucht (TID/OID Werte).

Es werden keine Prüfungen (Interlis-Checks) gemacht, sondern auf der Basis der XTF-Dateien die Datensätze extrahiert und gespeichert.

## Referenzen

Die Referenzen werden nicht aus dem Interlis-Modell (ILI) gelesen, sondern sind im Kontroller fest eingestellt.

## Irrelevante Objektklassen

Datensätze der Irrelevanten Objektklassen werde ohne Prüfung entfernt, respektive nicht übertragen.

## Filterklassen und Filterattribute

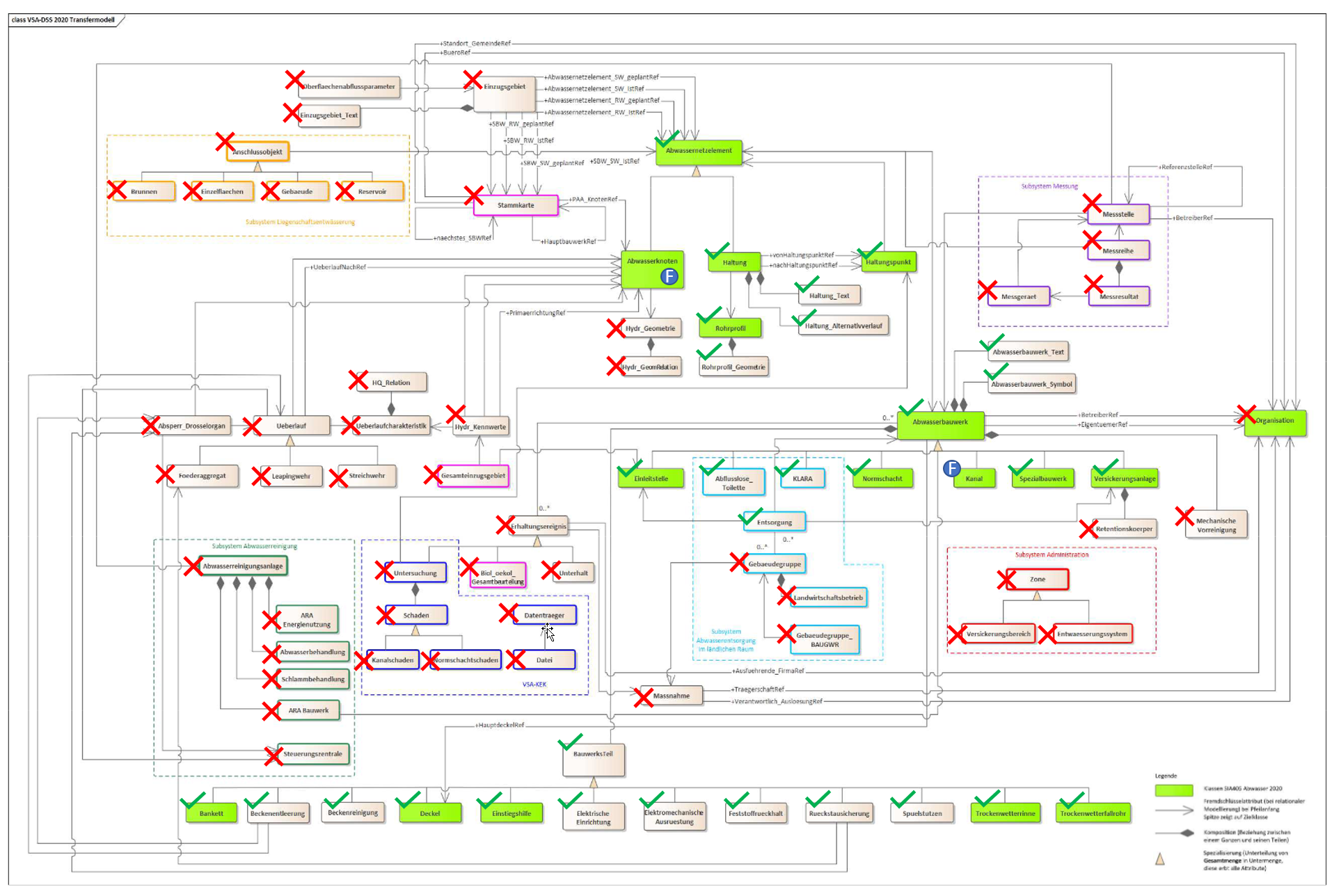
Die folgenden Klassen und Attribute werden als Filter verwendet:

*Klasse: Attribut:*

Kanal FunktionMelioration

Abwasserknoten Funktion\_Knoten\_Melioration

## Objektklassenübersicht



|  |  |  |
| --- | --- | --- |
| Symbol: | Bezeichnung: | Beschreibung: |
|  | Filterklasse | Klassen bei deren die Filter angewendet werden und die entsprechenden Datensätzen werden übertragen. |
| Häkchen mit einfarbiger Füllung | Referenzierte Klasse | Diese Klassen beinhaltet Daten, die in Abhängigkeit (Referenz) zu den Filterklassen beinhalten.  Hier werden die entsprechenden Referenzierten Datensätze übertragen. |
| Schließen mit einfarbiger Füllung | Irrelevante Klasse | Daten in diesen Klassen werden nicht übertragen. |

# SAD – Kommandozeilen Argumente

## Argumente

|  |  |  |
| --- | --- | --- |
| Argument: | Kurz: | Beschreibung: |
| --type | -t | Angabe der Anzuwendenden Funktion (siehe detaillierte Beschreibung) |
| --input | -i | Pfad und Dateiname der Eingabe (Input) Datei.  In der Regel das XTF mit den Quelldaten. |
| --output | -o | Pfad und Dateiname der Ausgabe (Output) Datei.  In der Regel das XTF für die Speicherung der Konvertierten Daten. |
| --log | -log | Pfad und Dateiname für die Protokoll Datei. |

## Argument «--type»

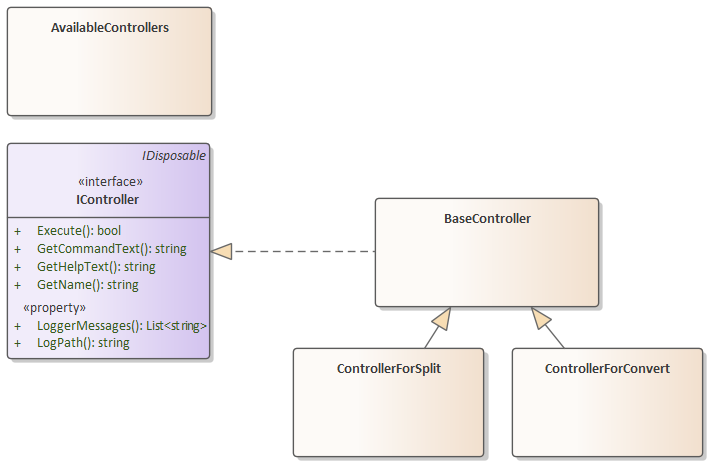
Das Argument «--type» wird für die Angabe der Funktion (Konvertierung oder Trennung) verwendet.

Für die Benutzenden muss dieser Wert in der Benutzeranleitung bekannt gemacht werden.

Der Wert wird in der Konkreten Implementation des Kontrollers als Konstante definiert und in den Verfügbaren Kontrollers (AvailableControllers) eingetragen.

# SAD – Controllers

Die «Controllers» definieren die verschiedenen Aufgaben innerhalb der Applikation.



## AvailableControllers

Jeder konkrete Implementation eines «Controllers» (:IController) muss in der Liste eingetragen werden.

Dies mit dem Funktionsname (FUNCTIONTYPE) der Gleichzeitig als Kommandozeilen Argument für den Aufruf verwendet wird.

## BaseController : IController

Stellt die Funktion über allen Kontroller sicher.

## Konkrete Implementationen

Die Konkreten Implementationen leiten vom BaseController ab und Implementieren so das Interface IController.

Für die Kontroller ist die Vorgabe, dass die Konstante «FUNCTIONTYPE» definiert wird, mit einem Eindeutigen Wert (UNIQUE) für die Erkennung des Kontrollers per Kommandozeilen Argument (--type).