Inhaltsverzeichnis

1. Modelltypen

INTERLIS-Modelle werden für verschiedene Zwecke eingesetzt. Entsprechend muss dies beim Modellieren berücksichtigt werden. Nicht in jedem Fall werden oder müssen sowohl ein Erfassungs- und eine Publikationsmodell geschrieben werden.

1.1. Erfassungsmodell

Erfassungsmodelle dienen der Erfassung und Nachführung von Daten in der Datenbank. Als Werkzeug zur Bewirtschaftung der Daten wird *QGIS* eingesetzt. Der Fokus bei der Modellierung liegt auf den Verzicht von Redundanzen, d.h. die Modelle sind normalisiert. So werden z.B. zwischen den einzelnen Klassen Beziehungen modelliert und Aufzähltypen modelliert, um allfällige Redundanzen zu vermeiden. Im Zusammenspiel mit *QGIS* entsteht somit eine «generische» Erfassungsfachschale.

1.2. Publikationsmodell

Aufgrund der Normalisierung eignen sich Daten im Erfassungsmodell nicht für die Publikation in einem Web GIS oder als «einfacher» Layer in *QGIS*. Die für den Benutzer wichtigen Informationen sind in einem Erfassungsmodell oftmals auf verschiedene Klassen resp. Datenbanktabellen verteilt. Für die reine Darstellung und das einfache Abfragen von Informationen ist ein simpleres Modell – das Publikationsmodell – zu erarbeiten. Hauptmerkmale eines Publikationsmodelles:

- Verzicht auf Beziehungen (Assoziationen) zwischen Klassen.
- Eine Geometriespalte pro Klasse.
- Der Inhalt kann zu jedem Zeitpunkt und automatisiert aus dem Erfassungsmodell und weiterer vorhandener Daten abgeleitet werden.

Es wird kein zusätzliches Datenabgabemodell angestrebt.

1.2.1. Aufzähltypen

Damit Aufzähltypen sinnvoll im Web GIS Client eingesetzt werden können, müssen zwei Bedingungen erfüllt sein: Jeder Aufzähltyp-Wert muss ein Metaattribut `@ili2db.dispName` (siehe dazu auch die offizielle Dokumentation von ili2pg) erhalten und es braucht in der Datentabelle eine zusätzliche Spalte, die den Wert des Metaattributes (den «schönen» Text) zum Aufzähltyp-Wert speichert.

Für die erste Bedingung muss mindestens UML-Editor grösser V3.9.0 verwendet werden. Diese Version erlaubt es für jedes Objekt beliebige Metaattribute zu erfassen.

Für die zweite Bedingung muss beim Erstellen des Schemas die Option `--createEnumTxtCol` verwendet werden. Diese erstellt für Aufzähltypattribute eine zusätzliche Spalte `<Attributname>_txt`. Werden z.B. Daten in ein solches Schema importiert, werden die `@ili2db.dispName`-Werte der Aufzähltypwerte gespeichert. Im häufigen Fall einer Publikation von Daten aus der Edit-DB in die Pub-DB muss man sich jedoch selber um das Speichern der `@ili2db.dispName`-Werte kümmern. Am einfachsten ist ein Update-Befehl mit einem SqlExecutor-Task nach dem Db2Db-Task. Der Update-Befehl macht ein Join mit der Aufzähltyp-Tabelle (`--createEnumTabs`).

1.3. Seitenwagenmodell

Ein Seitenwagenmodell ist strukturell ein Publikationsmodell, liegt aber in der Erfassungsdatenbank. Es wird für den CCC-Cache verwendet, d.h. die externe Fachanwendung schreibt via Data-Service in die Erfassungsdatenbank.

1.4. Validierungsmodell

Das Validierungsmodell dient zur Definition von zusätzlichen Prüfungen mit der Software *ilivalidator*. In diesem Modell müssen zuerst Views erstellt werden. In diesen Views können mit Constraints die zusätzlichen Prüfungen definiert werden. Ein Beispiel findet sich hier.

Weitergehende Informationen zu den Modelltypen gibt es in der Präsentation des AGI vom 29. August 2017 zum Thema «Arbeiten mit einem Datenmodell».

Zu einem Validierungsmodell gehört auch immer eine Konfigurations- und Metakonfigurationsdatei.

1.5. Extension-Functions-Modell

Zum jetzigen Zeitpunkt (2022-06-22) gibt es *ein* Modell (`SO_FunctionsExt`), in welchem zusätzliche Prüffunktionen deklariert werden. Die Funktionen selber werden in Java implementiert. Der Modellnamen dieses Modelles ist statisch, damit nicht bei jedem neuen Release sämtliche Validierungsmodelle, welche dieses Modell importieren, nachgeführt werden müssen. Dies bedingt jedoch eine strenge Stabilität der Funktionssignaturen.