



Hochschule Neubrandenburg
University of Applied Sciences

Hochschule Neubrandenburg
Studiengang Geoinformatik

Entwicklung eines Konvertierungstools zwischen der ALKIS-Schnittstelle
NAS und der thüringenspezifischen Schnittstelle ASI/ASO

Bachelorarbeit

zum Erlangen des akademischen Grades

„Bachelor of Engineering“

vorgelegt von

Michael Wolf

thesis 2012-0632-2

Betreuer: Prof. Dr. Dipl.-Ing. Andreas Wehrenpfennig

Dipl.-Ing. Frank Fuchs

2012

Zusammenfassung:

In dieser Arbeit wird auf die Problematik der Datenformate bei der Umstellung, des Freistaates Thüringen, nach ALKIS eingegangen. Es werden zuerst die nötigen Grundlagen des AAA-Models dargelegt und die beiden Datenformate beschrieben. Dabei werden die Besonderheiten der Formatstrukturen und deren Aufbau im Freistaat Thüringen dargestellt. Anschließend wird die Schnittstelle zwischen den beiden Formaten erklärt. Zusätzlich wird ein Vergleich zwischen Bestandsdaten und Fortführungsdaten beschrieben. Abschließend wird ein Lösungsansatz in Form eines Programms, mit der Programmiersprache JAVA, für die Konvertierung der Daten und dem Vergleich entwickelt.

Abstract:

In this work addresses the problem of data formats in the transition for the state of Thuringia after ALKIS. First there are the necessary basis of the AAA models illustrated and described the two data formats. Account the characteristics of the format structures and their development in Thuringia. Then, the interface between the two formats is explained. In addition, a comparison of inventory data and continuation data is described. Finally, a solution in the form of a program with the Java programming language, for the conversion of data and the comparison is developed.

Inhaltsverzeichnis	Seite
Inhaltsverzeichnis	I-III
Abbildungsverzeichnis	IV-V
Tabellenverzeichnis	VI
Abkürzungsverzeichnis	VII-VIII
1. Einleitung	1
1.1 Aufgabenstellung	2
1.2 Aufbau der Arbeit	2
2. Das AAA Modell	2
2.1 Grundlagen von ALKIS, ATKIS und AFIS	2
2.1.1 Das AFIS-ALKIS-ATKIS-Anwendungsschema	4
2.1.2 Das AFIS-ALKIS-ATKIS-Basisschema	5
2.1.3 Das Fachschema	6
2.1.4 Attribute	7
2.1.5 Relationen	7
2.1.6 Koordinatenreferenzsysteme	8
2.2 NAS	8
3. ASI/ASO Format	10
4. XML Format	17
4.1 Allgemeiner Aufbau	18
4.2 Beispiel eines Vermessungspunkt in ALKIS	19

5.	Konvertierung der Fachinformationen	22
5.1	Objektbereiche, Objektartengruppen und Objektarten für die Aufgabe	23
5.2	Positionen der ASI/ASO Fachinformationen in XML Dokument	23
5.3	Rückmigration der Fachinformationen	24
6.	Vergleich zwischen Bestands- und Fortführungsdaten	25
7.	Lösungsansatz	25
7.1	Phase der Analyse	26
7.1.1	Anforderungen „XML2ASI“	28
7.1.2	Anforderungen „XML-ASO“	29
7.2.	Phase des Entwurfs	29
7.2.1	Aufbau des Programms	29
7.2.2	Entwurf des Programms	32
7.2.2.1	„XML2ASI“ Entwurf	32
7.2.2.2	„XML-ASO“ Entwurf	33
7.3	Phase der Implementierung	34
7.3.1	„XML2ASI“ Feinentwurf	35
7.3.2	„XML-ASO“ Feinentwurf	44
7.4	Phase der Integration	53
7.4.1	Verbinden der Komponenten	53
7.4.2	Test des Gesamtsystem	53
7.5	Phase der Installation	53
7.6	Phase der Wartung	54
8.	Ergebnis	54

9. Fazit	55
Literaturverzeichnis	56
Anlagen	58
Eidesstattliche Erklärung	66

Abbildungsverzeichnis

		Seite
Abbildung 1	Die Rolle des Anwendungsschema	5
Abbildung 2	XML-basierende Kodierungsregeln gemäß ISO 19118	9
Abbildung 3	ASI/ASO Format Attribute	10
Abbildung 4	ASI Export mit ALK	27
Abbildung 5	ASI Export über ALKIS	28
Abbildung 6	Umsetzung des Systems	31
Abbildung 7	Komponenten XML2ASI	33
Abbildung 8	Komponenten XML-ASO	34
Abbildung 9	XML-Modul (parser)	36
Abbildung 10	XML-Modul (REO Klassen filtern)	37
Abbildung 11	XML-Modul (ordnen der REOs)	38
Abbildung 12	XML-Modul (Kartendarstellung)	39
Abbildung 13	XML-Modul (Vergleich Kartendarstellung)	40
Abbildung 14	XML-Modul (Koordinaten)	40
Abbildung 15	Methoden zur Umwandlung mit Steuerdatei	42
Abbildung 16	Methoden zur Umwandlung ohne Steuerdatei	42
Abbildung 17	GUI XML2ASO	44
Abbildung 18	ASO Modul	45
Abbildung 19	PKZ Vergleich	46
Abbildung 20	utm_aso- und utm_bestand Modell füllen	48
Abbildung 21	ASI/ASO Liste färben	49

Abbildung 22	Bestandsliste färben	50
Abbildung 23	Tabelle färben	51
Abbildung 24	GUI XML-ASO	52

Tabellenverzeichnis

		Seite
Tabelle 1	Produktgruppen des Referenzmodells	3
Tabelle 2	PAT Schlüsselzahlen	12
Tabelle 3	PST Schlüsselzahlen	12
Tabelle 4	VAT Schlüsselzahlen	13
Tabelle 5	PGW Schlüssel	14
Tabelle 6	OSK- und VAT- Schlüssel	15
Tabelle 7	LST Verschlüsselung	16
Tabelle 8	HER Verschlüsselung	16
Tabelle 9	LGS Verschlüsselung	17
Tabelle 10	ZUSO mit zwei REO	17
Tabelle 11	ZUSO mit drei REO	18
Tabelle 12	ZUSO mit einem REO	18
Tabelle 13	Inhalt der XML-Datei (ZUSO)	18
Tabelle 14	Inhalt der XML-Datei (REO)	19
Tabelle 15	tags der Objekte, Attribute, Werte eines ZUSO	20
Tabelle 16	tags der Objekte, Attribute, Werte des ersten REO	21
Tabelle 17	tags der Objekte, Attribute, Werte des zweiten REO	22
Tabelle 18	Objektarten für Aufgabe	23
Tabelle 19	Position der ASI/ASO Werte in XML-Datei	24

Abkürzungsverzeichnis

AAA	AFIS-ATKIS-ALKIS
AFIS	Amtliches Festpunktinformationssystem
AKZ	Aktenkennzeichen
ALB	Automatisiertes Liegenschaftsbuch
ALK	Automatisierte Liegenschaftskarte
ALKIS	Amtliches Liegenschaftskatasterinformationssystem
APK	Auskunft- und Präsentationskomponente
ASCII	American Standard Code for Information Interchange
ASI	ASCII Input
ASO	ASCII Output
ATKIS	Amtliches Topographisch-Kartographisches Informationssystem
CRS	Koordinatenreferenzsystem
DHK	Datenhaltungskomponente
DLKM	Digitales Liegenschaftskataster Modell
EQK	Erhebungs- und Qualifizierungskomponente
ETRS89	Europäisches Terrestrisches Referenzsystem 1989
GeoDB	Geographische Datenbank
GK	Gauß-Krüger Koordinaten
GML	Geography Markup Language
HER	Herkunft
HST	Höhensystem
LGS	Lagegenauigkeitsstufe
LST	Lagestatus
NAS	normbasierte Datenaustauschnittstelle
NBZ	Nummerierungsbezirk
NREO	nicht raumbezogenes Elementarobjekt
OCG	Open Geospatial Consortium
OSK	Objektschlüssel
OSKA	Objektschlüsselkatalog
PAT	Punktart
PD83	Potsdamer Datum 1983
PGW	Punktgewicht
PKZ	Punktkenzeichen
PMO	Punktmengenobjekte
PNR	Punktnummer
PST	Punktstatus
PunktDB	Punktdatenbank
REO	Raumbezogene Elementarobjekte
TLVermGeo	Thüringer Landesamt für Vermessung und Geoinformation
UML	Unified Modeling Language
UTC	Universal Time Coordinated
UTM	Universal Transverse Mercator
VAT	Vermarktungsart

XML Extensible Markup Language
ZUSO Zusammengesetzte Objekte

1. Einleitung

Grund und Boden hat für Menschen schon immer einen hohen Wert. Er ist ein sicheres Vermögen und Prestige. In der Bundesrepublik Deutschland wird das Eigentum an Grund und Boden sehr gepflegt und geregelt um Streitigkeiten zu verhindern, Ansprüche klar nachzuvollziehen und Planungen durchzuführen.

Der Nachweis von Flächen bzw. Eigentum ist ein schwieriges und aufwendiges Verfahren, in der Aufgabe von Behörden. Erschwerend beim Nachweis steht der Föderalismus in Deutschland. Die Vermessung ist Ländersache und dadurch verfügt jedes Bundesland über eigene Gesetze.

Die Bundesländer haben also je eine eigene Datenführung und Datenerhebung. Um ein einheitliches Grundgerüst zu schaffen, wurde von der AdV (Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschlands) auf der Grundlage von Normen und Standards ein Modell entwickelt. Dieses Modell soll nicht nur den Vermessungsnachweis einheitlich gestalten, sondern auch eine Vereinfachung und Qualitätssteigerung mit sich bringen. Durch die technische Entwicklung sowohl im IT- Bereich als auch im Vermessungsbereich ist eine Umsetzung des Modells erst möglich.

Im Freistaat Thüringen ist zurzeit die Konzeption und Koordinierung eines neuen Systems, Namens AAA Modell (ATKIS-ALKIS-AFIS Modell), in Arbeit. Die amtliche Datenhaltung wird gegenwärtig mit Hilfe von ALK (Automatisierte Liegenschaftskarte), ALB (Automatisiertes Liegenschaftsbuch) und ATKIS (Amtliches Topographisch-Kartographisches Informationssystem) realisiert. Die Umstellung nach AFIS (Amtliches Festpunktinformationssystem), ALKIS (Amtliches Liegenschaftskatasterinformationssystem) und ATKIS (Amtliches Topographisch-Kartographisches Informationssystem) stellt die zuständige Behörde vor einige Aufgaben.

In dieser Arbeit wird ein Abschnitt der Problematik ALKIS-Umstellung des Freistaates Thüringen in seiner Weiterentwicklung behandelt. Das TLVermGeo (Thüringer Landesamt für Vermessung und Geoinformation) beginnt im Jahr 2012 mit der Einführung von ALKIS. Die Einführung erfolgt stufenweise und betrifft in der ersten Stufe den Bereich Datenführung. Der Bereich Datenerhebung soll erst nach Abschluss der ersten Stufe erfolgen. Aus der zeitlichen Differenz ergibt sich naturgemäß ein Nebeneinander von neuen (Datenführung) und alten Technologien (Datenerhebung). Dieses Nebeneinander erfordert neben Anpassungen in den Arbeitsabläufen auch die Modifizierungen in den Schnittstellen.

Der Datenaustausch zwischen der Datenführung (Katasterbereiche) und der Datenerhebung (Vermessungsstellen, Katasterbereiche, ÖbVI) erfolgt derzeit über die thüringenspezifische

Schnittstelle ASI/ASO. Diese Schnittstelle wird von ALKIS nicht unterstützt, da sich grundlegende Datenstrukturen verändert haben. Für einen gewissen Übergangszeitraum ist es notwendig das ASI/ASO Format weiter zu führen.

Es ist erforderlich, unter Berücksichtigung bestimmter Standardeinstellungen ein Konvertierungstool zwischen der ALKIS-Schnittstelle NAS (XML Format) in die ASI/ASO Schnittstelle zu schaffen. Dieses Tool soll weiterhin auf dem „Rückweg“ von der Vermessungsstelle zum Katasterbereich einen Vergleich zwischen den Bestandsdaten und den Fortführungsdaten vornehmen und das Ergebnis visualisieren und übersichtlich protokollieren.

1.1 Aufgabenstellung

Das Format ASI/ASO wird von den Komponenten des ALKIS nicht ausgegeben. Die Vermessungsstellen, Katasterbereiche und ÖbVI sind momentan nicht in der Lage XML-Dateien (Extensible Markup Language) zu verarbeiten und zu kontrollieren.

So ergibt sich eine Diskrepanz zwischen Datenführung in den Datenbanken und der Datenerhebung im Außen- bzw. Innendienst.

In dieser Arbeit wird für die obige Aufgabe eine mögliche Lösung aufgezeigt. Durch die Entwicklung eines Computerprogramms wird eine Schnittstelle zwischen den Formaten XML und ASI/ASO geschaffen und eine Möglichkeit die Bestandsdaten mit den Fortführungsdaten zu kontrollieren.

1.2 Aufbau der Arbeit

Im ersten Abschnitt werden die Grundlagen zum AAA Modell erläutert und wie es aufgebaut ist.

Die Analyse der beiden Formate ist im zweiten Abschnitt beschrieben. Es wird der Aufbau der Formate XML und ASI/ASO und die Bedeutung der einzelnen Informationen dargelegt. Ein weiterer Teil dieses Abschnittes sind die Regeln der Konvertierung des XML Formates, sowie der Vergleich zwischen Bestandsdaten und Fortführungsdaten.

Der dritte Abschnitt behandelt einen möglichen Lösungsweg der Aufgabe in Form eines Computerprogramms.

2. Das AAA Modell

2.1 Grundlagen von ALKIS, ATKIS und AFIS

In den Bundesländern haben Vermessungs- und Katasterstellen die Aufgabe raumbezogene Basisdaten für Verwaltung, Wirtschaft und private Nutzer zu liefern. Mit fortschreitender

technischer Entwicklung ist es möglich, die Informationen in digitaler Form zu liefern. Ein erster Ansatz war, die Daten des Liegenschaftskatasters in der ALK und dem ALB deutschlandweit einheitlich zu erfassen und zur Verfügung zu stellen. Die Daten der Topographischen Landesaufnahme wurden im ATKIS digital realisiert. Aus den Daten der ALK, ALB und ATKIS entstehen Geobasisdaten für andere FIS [GeoInfoDok].

In ALKIS sind die liegenschaftsbeschreibenden Daten des ALB und die Daten der ALK in einem Datenmodell beschrieben. Die Neukonzeption von ALKIS im Vergleich zum ALB und zur ALK umfasst ein neues, zwischen ALKIS und ATKIS abgestimmtes Datenmodell. Des Weiteren sind durch AFIS zusätzlich die Punkte der Grundlagenvermessung integriert. Durch dieses Konzept soll es möglich sein, die Punktdaten der Bundesländer einheitlich zu beschreiben. AFIS, ATKIS und ALKIS sorgen für eine deutliche Qualitätsverbesserung und einheitliche Nachvollziehbarkeit der Geobasisdaten.

Jeder Teil des AAA-Modells enthält unterschiedliche Bereiche und Beschreibungen der einzelnen Geobasisdaten. Die Quellen für das Datenmodell können aus analogen oder digitalen Auszügen stammen.

In nachfolgender Tabelle sind die einzelnen Produktgruppen von AFIS, ALKIS und ATKIS dargestellt

AFIS	enthält beschreibende und darstellende Daten	AFIS-Bestandsdaten
		digitale AFIS-Auszüge
		analoge AFIS-Auszüge
ALKIS	enthält liegenschaftsbeschreibende und -darstellende Daten	ALKIS-Bestandsdaten
		digitale ALKIS-Auszüge
		analoge ALKIS-Auszüge
ATKIS	beschreibt Landschaft mit unterschiedlichen Anwendungsziele	digitales Landschaftsmodell
		digitales Geländemodell
		digitale topographische Karten
		analoge Auszüge
		digitale Bildmodelle

Tabelle 1: Produktgruppen des Referenzmodells (Quelle: In Anlehnung an GeoInfoDok, Seite 8)

Um die Qualitätssteigerung und Vereinfachung für die Vermessungsverwaltung zu erzielen wurde von der AdV das AAA- Anwendungsschema entwickelt. Es besteht aus dem fachneutralen AAA- Basisschema und den drei Fachschemata AFIS, ALKIS und ATKIS. Die Schemata sind in UML beschrieben und somit entsteht eine standardisierte Dokumentation. Die Beschreibung mit UML hat folgende Vorteile:

- Eine fachliche und transparente Struktur, welche in den vorherigen Verfahrenslösungen der Vermessungsverwaltung nur unzureichend vorhanden war.

- Die Informationen in den Komponenten können ohne länderspezifische Besonderheiten allgemein beschrieben werden.
- Ohne Veränderung im Datenmodell können Weiterentwicklungen der GIS-Technologie integriert werden
- Das Modell ist ausbaubar und beliebig erweiterbar.
- Die Geschäftsprozesse im amtlichen Vermessungswesen können vollständig dargestellt werden.
- Die Qualitätssicherung für Produkte ist durch eine standardisierte Prüfung gegenüber den Vorgaben des Datenmodells möglich.

(Quelle: Wissenschaftlicher Beitrag für den Aufbau einer Geodateninfrastruktur zur Lösung von Aufgaben des E-Government)

2.1.1 Das AFIS-ALKIS-ATKIS-Anwendungsschema

Das AAA- Anwendungsschema ist Modular aufgebaut. Es hat neben dem fachneutralen Basis-schemata und den fachspezifischen Fachschemata AFIS, ALKIS und ATKIS das Modul NAS (normbasierte Datenaustauschnittstelle). In der NAS wurden die GIS-Standards des OCG (Open Geospatial Consortium) beachtet. Es ermöglicht somit, in Online Portalen zu arbeiten. Die Modellierung des Anwendungsschemas erfolgt durch eine Vielzahl an Normen und Standards. Neben den Normen und Standards ist die Modellierung mit einem einheitlichen Objektartenkatalog an einem Objekt orientiert. Das Anwendungsschema hat den Zweck, ein gemeinsames und einheitliches Verständnis der Daten zu definieren und Dateninhalte zu dokumentieren [GeoInfoDok].

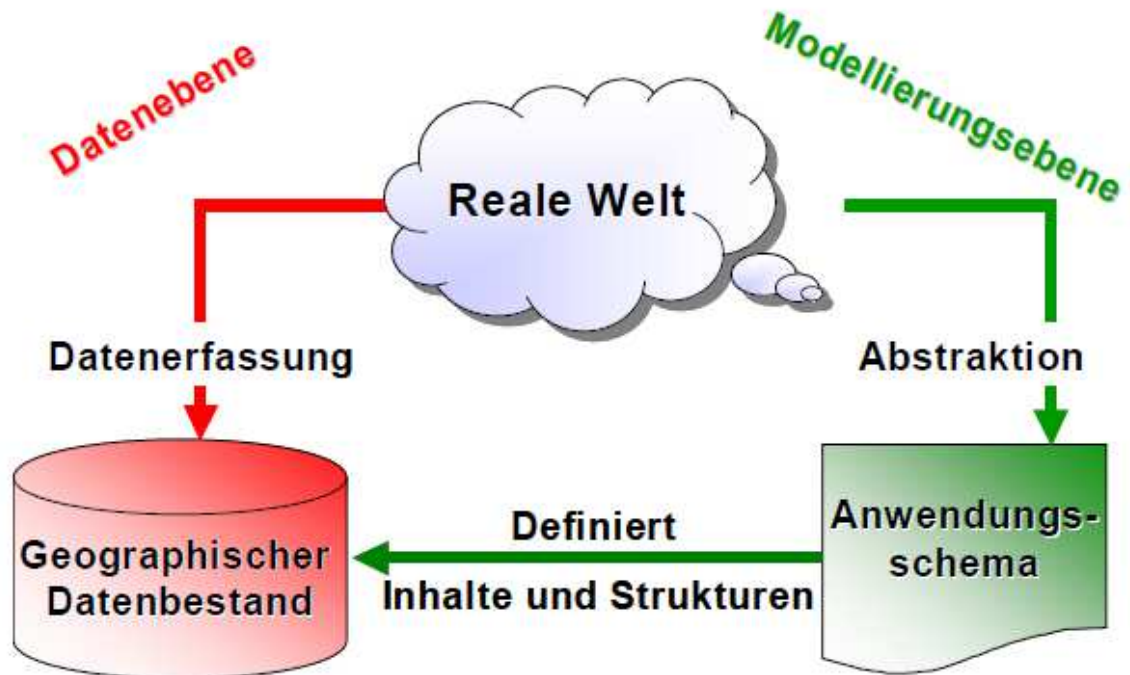


Abbildung 1: Die Rolle des Anwendungsschema (Quelle: GeoInfoDok Abbildung 2)

2.1.2 Das AFIS-ALKIS-ATKIS-Basisschema

Das Basisschema ist das Kernelement des AAA Modells. In ihm sind die fachneutralen Beschreibungen der Grundelemente in Klassen mit ihren Objekten und deren Datenaustausch dargestellt. Dies ist für ein einheitliches Datenmodell notwendig und somit für die einheitlichen Geobasisdaten. Das Datenmodell Basisschema ist auf ISO- und OGC-Standards aufgebaut. Mit diesen Normen und Standards werden allgemeingültige Regeln zur Bildung von Objekten definiert. Damit besitzt das Basisschema grundlegende Eigenschaften, die von den drei Fachschemata fachübergreifend und grenzübergreifend angewendet werden können.

Im Basisschema sind zehn Pakete enthalten, aber nicht alle sind von Bedeutung für die Vermessungspunkte. Das Paket der Basisklasse und die Pakete der Geometrie sind wichtig und werden verwendet. Damit die Klassen AAA-Basisklassen, AAA-Unabhängige Geometrie, AAA-Gemeinsame Geometrie und AAA-Spatial Schema einheitlich sind, bestehen gewisse Systematiken in ihrer Bezeichnung.

1. Klassen mit grundsätzlicher Bedeutung für AFIS, ALKIS und ATKIS erhalten das Präfix AA
2. Klassen, die aus den ISO TS_*Component-Klasse ('simple topology') abgeleitet wurden, erhalten das Präfix TA
3. Klassen mit gemeinsam genutzter Geometrie erhalten das Präfix AG

4. Klassen mit unabhängigen Geometrie erhalten das Präfix AU

(Quelle: GeoInfoDok, Version 6.0.1)

Mit Hilfe dieser Klassen können grundlegende und allgemeine Elemente von Objektarten definiert werden.

Das Basisschema trennt die AFIS, ALKIS- und ATKIS-Objekte in Objekte mit geometrischer Ausprägung sowie Objekte ohne Raumbezug (z.B. Personen). Um die Erstellung der Fachschemata zu unterstützen, werden im gemeinsamen AAA-Basisschema fünf generelle Arten von Objektausprägungen vordefiniert [GeoInfoDok]:

1. Raumbezogene Elementarobjekte (REO)

Diese werden gebildet, wenn zu fachlichen Eigenschaften auch geometrische oder topologische Eigenschaften nachgewiesen werden sollen.

2. Raumbezogene Elementarobjekte 3D (REO_3D)

Die geometrischen Eigenschaften enthalten drei Dimensionen

3. Nicht raumbezogene Elementarobjekte (NREO)

Diese sind zu bilden, wenn nur fachliche Eigenschaften nachgewiesen werden sollen.

4. Zusammengesetzte Objekte (ZUSO)

Sie werden gebildet, wenn der Zusammenhang von mehreren semantisch zusammengehörenden raumbezogener Elementarobjekten hergestellt werden soll. Ein AA_ZUSO muss mindestens ein Objekt als Bestandteil haben.

5. Punktmengenobjekte (PMO)

Das AA_PMO ist ein Objekt, welches eine große Anzahl geometrischer Orte mit jeweils gleichen Attributarten vereint. In manchen Situationen ist es besser ein AA_PMO zu bilden, als einzelne AA_REOs. [GeoInfoDok].

2.1.3 Das AAA Fachschema

Das Fachschema führt das Basisschema weiter und definiert die fachspezifischen Anforderungen. Durch Vererbung aus dem Basisschema entstehen fachliche Inhalte. Damit können Erweiterungen in das Modell integriert werden. Mit diesen Erweiterungen werden Fachobjekte gebildet, welche einen Vermessungspunkt länderspezifisch beschreiben können. Durch

das Basisschema wird eine einheitliche Grundlage gegeben. Die Fachobjekte werden im Objektartenkatalog aufgeführt. Im Katalog unterteilen sich die Objekte in Objektartenbereiche, Objektartengruppen und Objektart mit Attribut-, Werte-, Relationsarten und Operationen [ALKIS-Objektartenkatalog]. Die Fachobjekte sind die Grundlage für die Migration der bisherigen Vermessungsnachweise.

Die Vorgaben für die Objektarten sind von der AdV vorgeschrieben.

2.1.4 Attribute

Die Objekte können Attribute besitzen. Diese sind selbstbezogene Eigenschaften der Objekte. Attribute sind über einen Namen und eine Wertart definiert. Wertarten können Zahlen, Zeichenketten, Datums- und Zeitangaben sein. Es ist auch möglich komplexe Datentypen, wie Geometrie oder Qualitätsmerkmale, zu erstellen. Die Attribute können multipel und Zeichenketten beliebig lang sein.

Datumsangaben werden nach den Festlegungen der ISO 8601 modelliert. Die Zeitgenauigkeit ist die volle Sekunde und die Zeitzone ist immer UTC (Universal Time Coordinated) [GeoInfoDok].

2.1.5 Relationen

Die beschriebenen Objekte in Fachschemata können untereinander in Beziehungen stehen, somit können sie auch fremdbezogene Eigenschaften besitzen. Fachobjekte können verschiedene Arten von Beziehungen zueinander eingehen. Diese werden in den fachspezifischen Sub-schemata definiert [GeoInfoDok]. Im gemeinsamen Basisschema sind einige Beziehungen zwischen Objekten fest vorgegeben.

1. Relationen zur Bildung von ZUSO

2. Unterführungsrelation

Es ist die Relation einer relativen Lage einzelner Objekte im Verhältnis zu anderen Objekten.

3. Kartengeometrie

Gibt die Relation an von welchen Objekten die Kartengeometrieobjekte abgeleitet wurden.

4. Generalisierung

Die Ableitung von 3D-Fachobjekten mit einer detaillierteren 3D Geometrie.

5. Fachdatenverbindung

Relation für ein Objekt, das auf ein fremdes Fachdatenobjekt verweisen soll

6. Darstellungsrelation

Der Nachweis zwischen Präsentationsobjekten und anderen Objekten wird über diese Relation beschrieben [GeoInfoDok].

2.1.6 Koordinatenreferenzsysteme

Das CRS (coordinate reference system) ist nach ISO 19111 ein mit der Erde verbundenes Bezugssystem zur modellhaften Beschreibung der räumlichen Punktlage. Es erfolgt eine Zweiteilung in das Koordinatensystem und das Datum [ETRS89/UTM - Der Bezugssystemwechsel und die Auswirkung auf die Geodatenutzung].

Es ist möglich in AFIS-ALKIS-ATKIS für jede Geometrie das zugehörige CRS anzugeben bzw. zu speichern. Nach ISO19111 können zusammengesetzte CRS eingeführt werden. Bei der Objektart „Punktort“ sind aber zusammengesetzte CRS nicht erlaubt. Da ALK- und ATKIS keine Angaben über das relevante CRS führen, ist es die Aufgabe des Migrationskonzeptes entsprechende Festlegungen zu treffen [GeoInfoDok].

In Thüringen werden die ALK Daten im Amtlichem CRS ETRS89/UTM (Europäisches Terrestrisches Referenzsystem 1989/ Universal Transverse Mercator) und zusätzlich im CRS PD83/GK (Potsdamer Datum 1983/ Gauß-Krüger Koordinaten) geführt. Mit der Umstellung zum AAA Modell soll in Zukunft nur das CRS ETRS89/UTM verwendet werden. Für Grenzuntersuchungen ist es nötig PD83/GK weiterhin zu führen.

2.2 NAS

Um im gemeinsamen AAA-Anwendungsschema modellierte Geobasisdaten und deren Fachinformationen auszutauschen, dient die NAS (normbasierte Austauschschnittstelle) als Hilfsmittel. Bei den Fachinformationen kann es sich um gespeicherte Datenbestände, einschließlich Zusatzdaten handeln oder um daraus abgeleitete Sichten auf diese Datenbestände. Es ist nicht möglich Datenbestände auszutauschen, bei denen der Objektbezug völlig verloren geht oder Daten, die nach einem anderen Basisschema zu definieren sind [GeoInfoDok]. Der Einsatz der NAS hat folgende Anwendungsschwerpunkte:

- Originalität der Daten
- volle Auswertbarkeit
- differenzierte Fortführbarkeit

Die NAS basiert auf den entwickelten XML-Standards des W3C (World Wide Web Consortium). Diese Standards sind XML, XML Namespaces, XML Schema, XLink, XPointer und XPath.

Die Beschreibung der Objektarten wird durch GML (Geography Markup Language) realisiert. (Wissenschaftlicher Beitrag für den Aufbau einer Geodateninfrastruktur zur Lösung von Aufgaben des E-Government, Seite 68). Es wird somit das UML-Modell des Anwendungsschemas in eine XML-Schemadatei abgebildet. Durch die XML-Schemadatei wird eine XML-Datei erstellt.

Der durchzuführende Kodierungs- und Dekodierungsprozess ist in der ISO 19118 beschrieben und sieht in allgemeiner Form folgendermaßen aus [GeoInfoDok]:

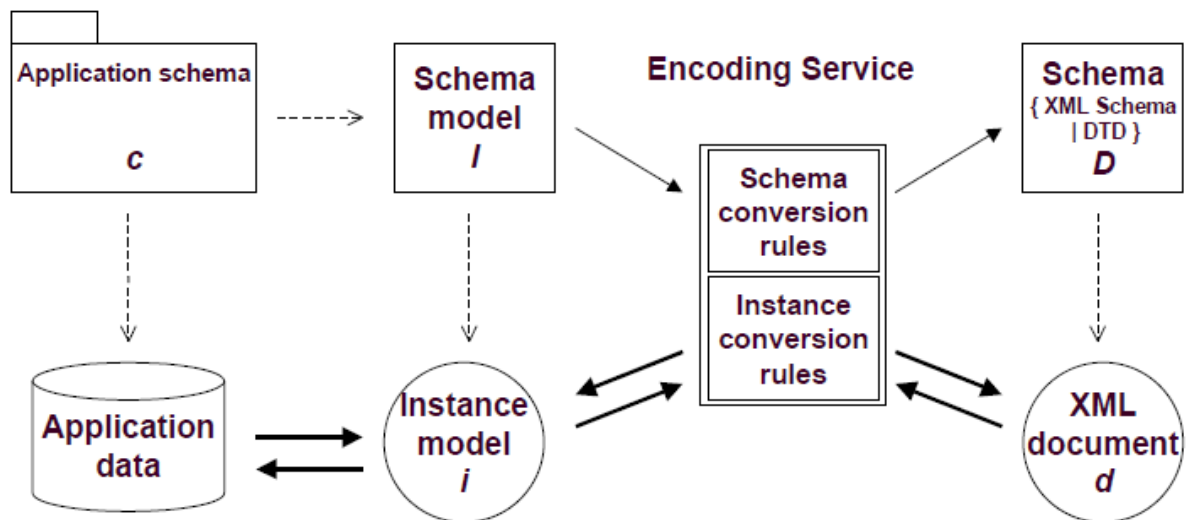


Abbildung 2: XML-basierende Kodierungsregeln gemäß ISO 19118 (Quelle: GeoInfoDok Abbildung 42)

Dieser Prozess hat folgende Rahmenbedingungen:

1. Es existiert ein formal (z.B. in UML) beschriebenes Anwendungsschema.
2. Auf der Basis von Umwandlungsregeln (Schema Conversion Rules) und ggf. Steueranweisungen werden die Informationen des UML-Anwendungsschemas in eine XML-Schemadatei überführt.
3. In gleicher Weise werden die auf dem Anwendungsschema beruhenden Anwendungsdaten (Objekt-Instanzen) mit Hilfe von Umwandlungsregeln (Instance Conversion Rules) in eine XML-Datei überführt, die in ihrem Aufbau den Definitionen der XML-Schemadatei entspricht. (Quelle: GeoInfoDok, Seite 83)

Die Ausgabe von Bestandsdaten ist somit eine XML-Datei. Die Vermessungspunkte in dieser XML-Datei müssen für die Weiterverarbeitung im Freistaat Thüringen in das Format ASI/ASO umgewandelt werden.

3. ASI/ASO Format

In diesem Teil der Arbeit wird das ASI/ASO Format der Bestands- und Fortführungsdaten beschrieben, um die nötigen Informationen in der XML-Datei zu filtern. Dies ist nötig, um im späteren Verlauf der Arbeit die Umwandlungsregeln für die Formate zu definieren.

Die Bezeichnung ASI/ASO steht für ASCII (American Standard Code for Information Interchange) Input und ASCII Output. Das Input-Format (ASI) ist das Eingangsformat für den Außendienst als Bestandsdaten. Das Output-Format (ASO) wird vom Außendienst herausgegeben als Fortführungsdaten. Diese ASCII Formate enthalten alle Fachinformationen eines Vermessungspunktes in zwei Zeilen. Die Fachinformationen bestehen aus einem Attribut und dem dazugehörigen Wert. Zur Beschreibung eines Vermessungspunktes werden die Fachinformationen kodiert. Jede Kodierung hat eine feste Position und Anzahl an Zeichen. Die nicht besetzte Positionen und Freiräume sind mit Leerzeichen aufzufüllen. Das Format ist 1997 speziell in Thüringen, mit der Umstellung eines neuen Außendienstprogramms, eingeführt worden. Nach der Einführung unterlag das ASI/ASO Format noch mehreren Veränderungen. In folgender Abbildung sind die einzelnen Attribute und ihre Position der aktuellen Version aufgeführt. Eine Beschreibung mit der Erweiterten Backus-Naur-Form ist in Anlage zwei.

1. Zeile										
Position	1	9	10	11-13	16-22	25-28	30-32	39-50	57-66	71-78
Bezeichner	Zeilenkennung	PGW	PAT	VAT	NBZ	PNR	LST	Rechtswert	Hochwert	Höhe
2. Zeile										
Position	1	2	6	8-15	19	20	25-31	42-60		
Bezeichner	Zeilenkennung	PST	HER	AKZ	HST	LGS	OSK	Datum		

Abbildung 3: ASI/ASO Format Attribute (Quelle: eigene Darstellung)

Die einzelnen Attribute und kodierten Werte geben Aufschluss über den Vermessungspunkt.

PKZ

Die PNR (Punktnummer), PAT (Punktart) und der NBZ (Nummerierungsbezirk) ergeben das PKZ (Punktkennzeichen). Diese Kennung ist ein Ordnungsmerkmal der Punkte und muss einmalig sein. [ThürVV-Lika].

NBZ

Der NBZ ist eine achtstellige Zahl, welche das Kilometerquadrat des PD83/GK darstellt. Die Zahl setzt sich aus den vollen Kilometerwerten (Rechts- und Hochwert) für die südwestliche Ecke des Kilometerquadrats zusammen. Folgende Regeln für die Kennzeichnung sind zu beachten:

1. Stelle: Nummer des Meridianstreifensystems
2. Stelle: 100 km des Rechtswertes
3. Stelle: 1000 km des Hochwertes
4. Stelle: 100 km des Hochwertes
5. Stelle: 10 km des Rechtswertes
6. Stelle: 1 km des Rechtswertes
7. Stelle: 10 km des Hochwertes
8. Stelle: 1 km des Hochwertes

Durch die Einführung von ALKIS erfolgt eine Umstellung der Nummerierung auf das Kilometerquadrat in ETRS89/UTM. Im Beschluss 16/3 vom 04.04.2012 ist die neue Formatbeschreibung erklärt und hat folgenden Aufbau:

1. bis 2. Stelle: Zonennummer
3. Stelle: 100 km des Ostwertes
4. Stelle: 10 km des Ostwertes
5. Stelle: 1 km des Ostwertes
6. Stelle: 1000 km des Nordwertes
7. Stelle: 100 km des Nordwertes
8. Stelle: 10 km des Nordwertes
9. Stelle: 1 km des Nordwertes

Bei der Einführung des ASI/ASO Formates wurde das PKZ mit einem siebenstelligen NBZ geführt. Um die ASI/ASO Formatbeschreibung mit der Einführung des ALKIS nicht zu ändern, werden alle neuen PKZ auf den alten siebenstelligen NBZ zurückgeführt. Wie sich das PKZ zusammensetzt wird unter dem Punkt 5.2 Rückmigration der Fachinformationen erklärt.

PAT

Die PAT (Punktart) ist die numerische Verschlüsselung von Vermessungspunkten nach ihrer Fachbedeutung. Die einstellige Zahl hat folgende Schlüssel:

Schlüsselzahl	Bezeichner
0	Trigonometrischer Punkt (TP)
1	Aufnahmepunkt (AP) Sonstiger Vermessungspunkt (SVP) Katasterfestpunkt (KFP) Sicherungspunkt zu KFP Punkt eines historischen Netzes Alter sonstiger Vermessungspunkt
2	Grenzpunkt (GP)
3	Gebäudepunkt (GbP)
4	Topographischer Punkt (TopP)

Tabelle 2: PAT Schlüsselzahlen (Quelle: in Anlehnung an ThürVV-Lika, Anlage 22)

PNR

Die PNR (Punktnummer) ist eine fortlaufende Nummer innerhalb des NBZ. Diese Nummer ist quadrantenweise ganzzahlig zu führen. Innerhalb eines NBZ darf keine doppelte Punktnummer vergeben werden.

PST

Der PST (Punktstatus) verschlüsselt die Historie des Vermessungspunktes. Es gibt folgende Schlüssel:

Schlüssel	Bezeichner
0	untergegangener Punkt mit Nachweis
1	aktueller Punkt des amtlichen Nachweises

Tabelle 3: PST Schlüsselzahlen (Quelle: in Anlehnung an ThürVV-Lika, Anlage 22)

VAT

Die VAT (Vermarkungsart) ist eine dreistellige Zahl und gibt an wie der Punkt vermarktet ist. Die zulässigen Vermarkungen sind in nachfolgender Tabelle aufgeführt.

Schlüssel	Bezeichner
000	Nach Quelle nicht zu spezifizieren
010	Stein
015	Landesgrenzstein
020	Rohr
035	Glasrohr
040	Bolzen
050	Schutzkasten
055	Drainrohr
060	Meißelzeichen
065	Nagel
070	Punkt dauerhaft und gut erkennbar festgelegt
075	Klebmarke
080	Hohlziegel
090	Ohne Marke
095	Abmarkung zurückgestellt
190	Steinplatte, unterirdisch
280	Marke besonderer Ausführung

Tabelle 4: VAT Schlüsselzahlen (Quelle: in Anlehnung an ThürVV-Lika, Anlage 22)

PGW

Das PGW (Punktgewicht) ist ein Faktor für die Bedeutung (Wichtung) der Koordinaten des Vermessungspunktes und wird einstellig verschlüsselt. Je höher das Punktgewicht, desto genauer die Lage.

In nach folgender Tabelle sind die Kodierungen der Schlüssel aufgelistet und wie diese entstehen.

Schlüssel	Bezeichner
0	PGW nicht bekannt
1	Aus Digitalisierung/Homogenisierung entstandene Koordinaten (Maßstab der Liegenschaftskarte < 1: 2000)
2	Aus Digitalisierung/Homogenisierung entstandene Koordinaten (Maßstab der Liegenschaftskarte < 1: 1000 bis einschl. 1: 2000)
3	Aus Digitalisierung/Homogenisierung entstandene Koordinaten (Maßstab der Liegenschaftskarte < 1: 1000) und Befliegungsergebnisse
4	Koordinaten der Topographie, alle Punkte der PAR 4
5	Koordinaten aus alten Vermessungen berechnet
6	Koordinaten aus Transformationen und aus Vermessungen mit einer Lagegenauigkeit $d_x > 3\text{cm}$ und $d_y \leq 3\text{cm}$
7	Koordinaten von verhandelten und festgestellten Grenz- und Gebäudepunkten mit einer Lagegenauigkeit $d_x \leq 3\text{cm}$ und $d_y \leq 3\text{cm}$
8	Koordinaten von KFP (Katasterfestpunkt) und deren Sicherungspunkten
9	Koordinaten von TP (Trigonometrischer Punkt)

Tabelle 5: PGW Schlüssel (Quelle: in Anlehnung an ThürVV-Lika, Anlage 22)

OSK

Der OSK (Objektschlüssel) ist eine Verschlüsselung der Punktobjekte für die Foliennummer und die Schlüsselzahl der jeweiligen Objektart. Zudem verschlüsselt der OSKA (Objektschlüsselkatalog) Grundrissinformationen und ist an den Objektschlüsselkatalog der AdV angelehnt [Gespräch Dipl.-Ing. (FH) Matthias Franke, 11.7.2012]. Der OSK steht somit in der Relation mit der PAT und der VAT.

OSK	Bezeichner	VAT
0510121	Aufnahmepunkt	020, 040, 060, 065, 190, 280
0510122	Sonstiger Vermessungspunkt	000, 010, 020, 040, 055, 060, 065, 075, 080, 090, 190
0510123	Katasterfestpunkt	000, 010, 020, 040, 055, 060, 065, 075, 080, 190, 280
0510124	Punkt eines historischen Netzes	000, 010, 020, 040, 050, 060, 065, 075, 080, 090, 190
0510125	Sicherungspunkt zu KFP	000, 010, 020, 040, 050, 060, 065, 075, 080, 090, 190
0520101	Grenzstein, Schlagvermarkung mit Kunststoffkopf	000, 010, 090
0520982	Grenzstein, Zentrum ist die Mitte der Oberkante des Steines	000, 010, 090
0520983	Grenzstein, Zentrum ist die Ecke des Steines	000, 010, 090
0520102	Unterirdischer Grenzpunkt	000, 010, 090
0520103	Grenzhügel	000, 070, 090
0520104	Grenzkreuz	000, 060, 090
0520105	Grenzbaum (Nadelbaum)	070, 090
0520106	Grenzbaum (Laubbaum)	070, 090
0520108	Sonstige Grenzmarke	000, 020, 035, 040, 050, 055, 065, 075, 080, 090, 190
0520984	Zaunpfosten	000, 070, 090
0520111	Grenzmarke Landesgrenze	000, 010, 015, 020, 040, 055, 070, 090
0520112	Grenzsäule Landesgrenze	000, 015, 090
0520118	Abgemarkter Grenzpunkt	000, 090
0520119	Nicht abgemarkter Grenzpunkt	070, 090, 095
0530151	Gebäudepunkt	070, 090
0540152	Topographischer Punkt	070, 090

Tabelle 6: OSK- und VAT- Schlüssel (Quelle: in Anlehnung ThürVV-Lika, Anlage 22)

Datum

Das Datum ist der Zeitpunkt, wann der Punkt das letzte Mal fortgeführt wurde. Folgende Formattierung für das Datum ist zulässig: tt.mm.jjjj hh:mm:ss

AKZ

Das AKZ (Aktenkennzeichen) ist ein Hinweis auf die Entstehung oder eine maßgebende Änderung des Punktes. Die achtstellige Kennung soll die entsprechende Antragsnummer gemäß der Antragsverwaltung führen [ThürVV-Lika].

LST

Im dreistelligen LST (Lagestatus) ist das Lagebezugs- und Koordinatensystem verschlüsselt. Folgende Verschlüsselungen sind zulässig:

Schlüssel	Bezeichner
120	GK-Koordinaten 3° Meridianstreifen Bessel_Ellipsoid im System PD83 (RW, HW)
489	UTM-Koordinaten im System ETRS89 (E,N)

Tabelle 7: LST Verschlüsselung (Quelle: in Anlehnung an ThürVV-Lika, Anlage 22)

HER

Die HER (Herkunft) ist eine Angabe zur Datenerhebung und Bestimmung der Koordinaten des Punktes.

Schlüssel	Bezeichner
0	Nach Quellenlage nicht spezifiziert
1	aus GNSS-Messung
2	aus Liegenschaftsvermessung ermittelt
3	aus Koordinatentransformation
4	aus Polygonierungsmessung
5	aus Berechnung oder Abstandsbedingung
6	aus Fernerkundungsdaten
7	aus Netzvermessung
8	aus Katasterkarte digitalisiert
9	aus sonstiger Vermessung

Tabelle 8: HER Verschlüsselung (Quelle: in Anlehnung an ThürVV-Lika, Anlage 22)

LGS

Die LGS (Lagegenauigkeitsstufe) gibt die zulässige tatsächliche Abweichung der Koordinaten an.

Schlüssel	Bezeichner
3	< 2cm
4	< 3 cm
5	< 4 cm
6	< 8 cm
7	< 14 cm
8	< 42 cm

Tabelle 9: LGS Verschlüsselung (Quelle: in Anlehnung an ThürVV-Lika, Anlage 22)

HST

Das HST (Höhensystem) wird für DHHN 92 mit 6 verschlüsselt. Die Höhe wird im Liegenschaftskataster nicht geführt.

4. XML Format

Dieser Teil der Arbeit stellt den Aufbau der Bestandsdaten im XML Format da. Die XML-Dateien müssen die Geobasisdaten wiedergeben, können aber durch die Bundesländer mit zusätzlichen Fachinformationen erweitert werden (siehe Kapitel 2.1.3). Der Aufbau des XML Formates ist durch die NAS geregelt. Ein Vermessungspunkt der Bestandsdaten ist ein ZUSO (zusammengesetzte Objekte). Dieses ZUSO benötigt, um die Geobasisdaten des Vermessungspunkt wiederzugeben, mindestens ein oder mehrere REO (raumbezogenes Elementarobjekt) und kann zusätzlich ein oder mehrere NREO (nicht raumbezogenes Elementarobjekt) besitzen. Eine Besonderheit in Thüringen ist das Führen zweier CRS. Hierdurch entstehen mehrere Koordinatenangaben und somit mehrere REO. Ein ZUSO kann folgende REO Kombinationen besitzen:

REO	CRS	Koordinaten
1	ETRS89/UTM	transformiert aus PD83/GK
2	PD83/GK	gemessen bzw. gerechnet

Tabelle 10: ZUSO mit zwei REO (Quelle: eigene Darstellung)

REO	CRS	Koordinaten
1	ETRS89/UTM	transformiert aus PD83/GK
2	ETRS89/UTM	gemessen bzw. gerechnet
3	PD83/GK	gemessen bzw. gerechnet

Tabelle 11: ZUSO mit drei REO (Quelle: eigene Darstellung)

REO	CRS	Koordinaten
1	ETRS89/UTM	digitalisiert aus Karten

Tabelle 12: ZUSO mit einem REO (Quelle: eigene Darstellung)

4.1 Allgemeiner Aufbau

Das ZUSO enthält grundlegende Informationen, wie die Angabe zur Objektart, die Identifikationsnummer des Objektes, das Lebenszeitintervall, Modellart, Anlass, zuständige Stelle, Markung und Zeitpunkt der Entstehung. Damit es möglich ist, mit Informationen des vorherigen Systems zu arbeiten, werden diese im ZUSO mitgeführt. Zu diesen Informationen gehören eine Fachdatenverbindung für das Aktenkennzeichen, die Punktkennung und sonstige Eigenschaften, welche das alte Punktkennzeichen von PD83/GK enthält.

Inhalt	Beschreibung
Objektart	Name des ZUSO
Identifikation	Kennung des Objektes
Lebenszeitintervall	Entstehungs- bzw. Endzeitpunkt des Objektes
Modellart	gibt an zu welchen Teil des 3A Modells das Objekt gehört (ALKIS/AFIS/ATKIS)
Anlass	gibt den Zweck, Benutzung und Fortführung der Informationen an
Fachdatenverbindung	Beziehung zu Außenstehenden Fachdaten
Punktkennung	Punktkennzeichen aus vorherigen System
zuständige Stelle	Bezeichnung des Bundeslandes und Amtes
Markung	Abmarkungsart bzw. Vermarkungsart
relative Höhe	Angabe bei Unterirdischer Vermarkungsart
sonstige Eigenschaften	weitere Informationen aus ALK-Punktdatei
Zeitpunkt der Entstehung	Datum der letzten Fortführung

Tabelle 13: Inhalt der XML-Datei (ZUSO) (Quelle: eigene Darstellung)

Zur Beschreibung des REO sind grundlegenden Informationen des Objektes, wie Objektart, Identifikation, Lebenszeitintervall, Modellart und Anlass, nötig. Die Objektart z.B. gibt Aufschluss

über die Klasse der Geometrie, die aus dem Basisschema abgeleitet wurde. Neben diesen Informationen enthält das REO die Lage, CRS und die Darstellungsart in der Liegenschaftskarte.

Inhalt	Beschreibung
Objektart	Name des REO
Identifikation	Kennung des Objektes
Lebenszeitintervall	Entstehungs- bzw. Endzeitpunkt des Objektes
Modellart	gibt an in welchem Teil des 3A Modells die Informationen sind
Anlass	gibt den Zweck, Benutzung und Fortführung der Informationen an
Relation zur Bildung ZUSO	Verknüpfung zu einem REO
Position	CRS
Kartendarstellung	Anzeige in AAA-Komponenten
Koordinatenstatus	gibt an ob Koordinaten amtlich sind oder anderen Status haben
Hinweis	nötig für die Koordinatenhomogenisierung
Qualitätsangaben	Enthält Qualitätsaussagen zu Lage

Tabelle 14: Inhalt der XML-Datei (REO) (Quelle: eigene Darstellung)

Die Inhalte sind in tags aufgeführt. Ein tag kann ein Attribut mit Namen und/ oder einer Wertart besitzen. Das REO hat den Präfix AU, AG oder TA und ist vom Objekt *Punktort* abgeleitet. Das ZUSO ist ein Fachobjekt. In den Tabellen 11, 12 und 13 ist an einem Vermessungspunkt der Aufbau dargestellt.

4.2 Beispiel eines Vermessungspunkt in ALKIS

Anhand eines Beispiels wird ein Vermessungspunkt in seiner zukünftigen Form beschrieben. Als erstes ist das ZUSO zu betrachten. Das Beispiel ist von der Objektart ein Grenzpunkt, aus dem Objektbereich Flurstücke, Lage, Punkte und besteht aus mindestens einem REO. Die Wertarten sind festgelegte Kodierungen, um die Informationen des Vermessungspunktes zu beschreiben.

Inhalt	tag	Attribut	Wertart
Objektart	AX_Grenzpunkt	DETHL51P0000069J	
Identifikation	gml:identifizier	http://www.adv-online.de	urn:adv:oid:DETHL51P0000069J
Lebenszeitintervall	lebenszeitintervall		
	AA_lebenszeitintervall		
	beginnt		2011-11-10T12:02:03Z
Modellart	modellart		
	AA_Modellart		
	advStandardModell		DLKM
Anlass	anlass		000000
Fachdatenverbindung	zeigtAufExternes		
	AA_Fachdatenverbindung		
	art		de.thueringen.tlvermgeo:antragskennzeichen
	fachdatenobjekt		
	AA_Fachdatenobjekt		
	name		51000112
Punktkennung	punktkennung		327675673020642
zuständige Stelle	zustaendigeStelle		
	land		16
	stelle		0050
Markung	abmarkung_Marke/Vermarkung_Marke		9500
sonstige Eigenschaften	sonstigeEigenschaften		APN=44564772220642
Zeitpunkt der Entstehung	zeitpunktDerEntstehung		1999-04-30, 10:10:42

Tabelle 15: tags der Objekte, Attribute, Werte eines ZUSO (Quelle: eigene Darstellung)

Das erste REO ist vom Objekt *Punktort* mit dem Präfix TA. Der *Punktort* ist Bestandteil des gemeinsamen Geometrie-Topologithemas und dient zur Darstellung in der Liegenschaftskarte [Dokumentation zur Modellierung der Geoinformationen, Seite 44]. Das REO steht in Beziehung zu dem ZUSO über einen Attributnamen im tag *istTeilVon*.

Inhalt	tag	Attribut	Wert
Objektart	AX_PunktortTA	DETHL51P00000hD6	
Identifikation	gml:identifizier	http://www.adv-online.de	urn:adv:oid:DETHL51P00000 fmd
Lebenszeitintervall	lebenszeitintervall		
	AA_Lebenszeitintervall		
	beginnt		2011-11-10T12:02:03Z
Modellart	modellart		
	AA_Modellart		
	advStandardModell		DLKM
Anlass	anlass		000000
Relation zur Bildung ZUSO	istTeilVon	DETHL51P0000069J	
CRS	Point	urn:adv:crs:ETRS89_UT M32	
	pos		447881.000 5672704.446
Kartendarstellung	kartendarstellung		true
Koordinatenstatus	koordinatenstatus		2000
Hinweis	hinweis		1011
Qualitätsangaben	AX_DQPunktort		
	herkunft		
	LI_Lineage		
	source		
	LI_Source		
	description		
	AX_Datenerhebung_Punktort		1900
	genauigkeitsstufe		2300

Tabelle 16: tags der Objekte, Attribute, Werte des ersten REO (Quelle: eigene Darstellung)

Das zweite REO dieses Vermessungspunktes ist auch ein *Punktort*, bezieht sich aber auf ein anderes CRS. Dieses CRS ist nicht Bestandteil des Flurstückthemas und kommt nicht zur Darstellung in der Liegenschaftskarte vor. Bei dem CRS ETRS89 können mehrere REO vorhanden sein, mit unterschiedlichen Koordinaten. Eine Koordinate ist zur Darstellung in der Liegenschaftskarte gedacht und die andere ist die gemessene bzw. gerechnete oder aus PD83 transformierte Koordinate. Die Darstellung ist im tag *kartendarstellung* durch die Wertarten „true“ (wahr) oder „false“ (falsch) abgebildet.

Inhalt	tag	Attribut	Wert
Objektart	AX_PunktortAU	DETHL51P00000fmd	
Identifikation	gml:identifizier	http://www.adv-online.de	urn:adv:oid:DETHL51P00000fmd
Lebenszeitintervall	lebenszeitintervall		
	AA_Lebenszeitintervall		
	beginnt		2011-11-10T12:02:03Z
Modellart	modellart		
	AA_Modellart		
	advStandardModell		DLKM
Anlass	anlass		000000
Relation zur Bildung ZUSO	istTeilVon	DETHL51P0000069J	
CRS	Point		
	pos	urn:adv:crs:DE_PD-83_3GK4	447881.000 5672704.446
Kartendarstellung	kartendarstellung		false
Koordinatenstatus	koordinatenstatus		2000
Hinweis	hinweis		1010
Qualitätsangaben	AX_DQPunktort		
	herkunft		
	LI_Lineage		
	source		
	LI_Source		
	description		
	AX_Datenerhebung_Punktort		1900
	genauigkeitsstufe		2100

Tabelle 17: tags der Objekte, Attribute, Werte des zweiten REO (Quelle: eigene Darstellung)

An dem Beispiel ist zu erkennen, dass das ZUSO aus zwei REO besteht. Im Ersten wird das amtliche CRS ETRS89 mit UTM Koordinaten geführt. Im Zweiten REO ist das CRS PD83 mit GK Koordinaten angegeben. Weiterhin ist zu erkennen, dass sich die Kodierungen sehr vom ASI/ASO Format unterscheiden. Um diese Unterschiede zu beseitigen, muss eine Konvertierung nach bestimmten Regeln erfolgen.

5. Konvertierung der Fachinformationen

Nach dem der Aufbau sowohl der ASI/ASO, als auch der XML-Datei bekannt ist, wird nun die Umwandlung der einzelnen Werte im nächsten Teil der Arbeit dargestellt. Die Migration von vorherigen Nachweisen in ALKIS ist in einem Stufenkonzept vorgesehen. Diese Konzepte sind länderspezifisch auszuführen. Die Rückmigration in die Schnittstellen der bisherigen Systeme ist für einen Übergangszeitraum möglich [GeoInfoDok]. Der Freistaat Thüringen hat für die Migration mehrere Tabellen aus der Umstellung nach ALKIS von Nordrhein Westfalen für den

eigenen Bedarf angepasst. Diese Tabellen basieren auf dem ALKIS-Objektkatalog der GeoInfoDok 6.0.1.

5.1 Objektartenbereiche, Objektartengruppen und Objektarten für die Aufgabe

Zunächst ist zu klären welche Objektartenbereiche, Objektartengruppen und Objektarten für die Aufgabe von Bedeutung sind. Es sind die Objekte bedeutend, bei denen gemessene, gerechnete und teilweise auch digitalisierte (Randpunkte der Katasterbereiche) Koordinaten aus der Punktdatensatz der ALK in ALKIS überführt wurden.

Objektartenbereich	Objektgruppe	Objektart
Flurstücke, Lage, Punkte	Angaben zum Flurstück Angaben zur Netzpunkt Angaben zum Punktort	AX_Grenzpunkt AX_Aufnahmepunkt AX_Sicherungspunkt AX_SonstigerVermessungspunkt AX_PunktortAG AX_PunktortAU AX_PunktortTA
Gebäude	Angaben zum Gebäude	AX_BesondererGebaedepunkt
Bauwerke und Einrichtungen in Siedlungsflächen	Angaben zu Bauwerken	AX_BesondererBauwerkspunkt

Tabelle 18: Objektarten für Aufgabe (Quelle: in Anlehnung GeoInfoDok ALKIS-Objektkatalog)

5.2 Positionen der ASI/ASO Fachinformationen in XML Dokument

Mit Hilfe der Migrationstabellen ist es möglich die Positionen der Fachinformationen in der XML-Datei herauszufiltern. In Tabelle 16 werden die Positionen der einzelnen ASI/ASO Fachinformationen in der XML-Datei dargestellt. Ein Teil der Informationen ist im ZUSO gespeichert und der andere Teil im REO.

ASI/ASO	XML (ZUSO)	XML (REO)
PGW		genauigkeitsstufe
PAR	AX_Objektart	
VAT	abmarkung_Marke/vermarkung_Marke	
NBZ	punktkennung	
PNR	punktkennung	
LST		Point bzw. pos
Koordinaten		pos
HER		AX_Datenerhebung_Punktort
AKZ	name	
HST		
LGS		genauigkeitsstufe
OSK	abmarkung_Marke/vermarkung_Marke	
Datum	zeitpunktDerEntstehung	

Tabelle 19: Position der ASI/ASO Werte in XML-Datei (Quelle: eigene Darstellung)

5.3 Rückmigration der Fachinformationen

Die Umwandlung der Werte von ALKIS zu ALK ist in den Migrationstabellen zum Teil festgelegt. Im Rahmen dieser Arbeit mussten Parameter, wie die Umwandlung des Punktgewichtes in Abhängigkeit des CRS und die Zuordnung des OSK, erst ermittelt werden. In der folgenden Aufzählung sind die Besonderheiten der Rückmigration aufgelistet.

1. Die Konvertierung des Punktgewichtes ist vom CRS abhängig.
2. Die Position des CRS in der XML-Datei ist unterschiedlich.
3. Die Abmarkung/ Vermarkung ist für einige Objektarten vorgeschrieben. Zum Beispiel hat der *AX_BesondererGebauedepunkt* immer die Vermarkung 070.
4. Das Format des Datums ist unterschiedlich und muss somit erst angepasst werden.
5. Da ein ZUSO mehrere REO besitzen kann, enthält das ZUSO mehrere Koordinaten in unterschiedlichen sowie gleichen CRS. Beim ASI/ASO Format sollen, falls vorhanden, die gemessenen bzw. gerechneten Koordinaten angegeben sein. Somit ist das ETRS89/UTM REO zu verwenden, wo der tag *kartendarstellung* den Wert „false“ besitzt. Sollte nur ein ETRS89/UTM

REO vorhanden sein, werden dessen Koordinaten verwendet. Eine weitere Möglichkeit ist, das REO mit der höchsten Genauigkeitsstufe zu ermitteln und diese Koordinaten zu verwenden.

6. Für das Aktenkennzeichen sind in der ASI/ASO-Datei acht Stellen reserviert. In der XML-Datei können bis zu zehn Stellen verwendet werden. Es ist nötig, die Stellen anzupassen und ggf. Zeichen bei der Umwandlung zu entfernen.

Zusätzlich ist zu beachten, dass die Informationen NBZ und PNR nicht umzuwandeln, sondern aus dem Wert des tags *punktkennung* zu ermitteln sind. Der NBZ setzt sich zusammen aus der 3., 4., 5., 7., 8., 9., 11. Stelle des Wertes. Die PNR steht an 12., 13., 14., 15. Stelle des Wertes.

Für das HST wird immer der Wert 6 vergeben.

Das tag *relativeHoehe* beeinflusst den VAT Schlüssel.

6. Vergleich zwischen Bestands- und Fortführungsdaten

Die konvertierten Bestandsdaten werden von den Vermessungsstellen, Katasterbereichen und ÖbVI verarbeitet und als ASO zurückgegeben. Die ASO-Datei soll nun mit den Bestandsdaten (XML-Datei) verglichen und gegen geprüft werden. Dieser Vergleich ist der zweite Abschnitt der Aufgabe und wird in diesem Teil der Arbeit näher erläutert

Als erstes ist zu klären, welches Format beim Vergleich eingehalten werden soll. Die Entscheidung fiel auf das ASI/ASO Format, da es bei den Bearbeitern bekannt ist und die geringere Umstellung veranlasst. Ein weiterer Vorteil für die Verwendung des ASI/ASO Formates ist, dass die Umwandlung von ASI/ASO- in XML-Format eine zusätzliche Fehlerquelle in sich birgt. Die Umwandlung der Bestandsdaten von XML- in ASI/ASO-Format für den Vergleich erfolgt im selben Verfahren wie bei der Rückmigration.

Der zweite Punkt ist die Wichtung von Bestandsdaten gegenüber Fortführungsdaten. Die Fortführung hat das höhere Gewicht, weil sie die Aktualisierung der Bestandsdaten ist.

Beim Vergleich beider Daten soll angezeigt werden, wo Veränderungen und Neupunkte auftreten. Jede Änderung soll zusätzlich protokolliert und gespeichert werden.

7. Lösungsansatz

Nach Erläuterung aller grundlegenden Verhältnisse, Strukturen und Aufgabenteile, wird im nächsten Teil der Arbeit eine mögliche Lösung der Aufgabe mit Hilfe einer Software dargelegt. Die Programmiersprache soll JAVA sein, da im TLVermGeo mit dieser Sprache gearbeitet wird und damit eine Weiterführung und Wartung des Programms realisierbar ist. Die Entwicklung

des Programms orientiert sich am Wasserfallmodell und die weiteren Gliederungspunkte sind nach dessen Aufbau festgelegt.

7.1 Phase der Analyse

Die Analyse erfasst den gegenwärtigen und zukünftigen Stand der Aufgabe und beschreibt die Anforderungen an die Software. Als Ergebnis dieser Phase entstehen die Systemspezifikationen.

Der aktuelle Stand zum Erstellen von ASI/ASO-Dateien ist durch das ALK Führungssystem und dem DAVID System gegeben. Das ALK Führungssystem, als amtliche Datenhaltung, besteht aus einer GeoDB (Grafikdatenbank) und einer PunktDB (Punktdatenbank). In der ersten Datenbank ist die Grafik abgelegt und in der zweiten Datenbank sind die Vermessungspunkte gespeichert. Jeder der acht Katasterbereiche des Freistaates Thüringen unterhält je eine GeoDB und PunktDB.

Die antragsbezogene Datenverarbeitung geschieht mit Hilfe von DAVID. DAVID ist das Basissystem zur Erstellung und Fortführung der ALK-Daten [www.thueringen.de/de/tlvermgeo]. Das amtliche Bezugssystem ist ETRS89/UTM, aber die ALK ist nur in PD83/GK darstellbar. Im DAVID kann mit Hilfe eines Umrings der auszugebende Bereich festgelegt werden. DAVID stellt eine Exportfunktion zur Verfügung. Mit dieser Exportfunktion werden die Vermessungspunkte, wie in der ASI/ASO Formatbeschreibung festgelegt, gespeichert. Die ausgegebene ASI-Datei kann im Feldsystem weiter verarbeitet werden und nach Bearbeitung und Kontrolle als ASO-Datei an DAVID zurückgegeben werden.

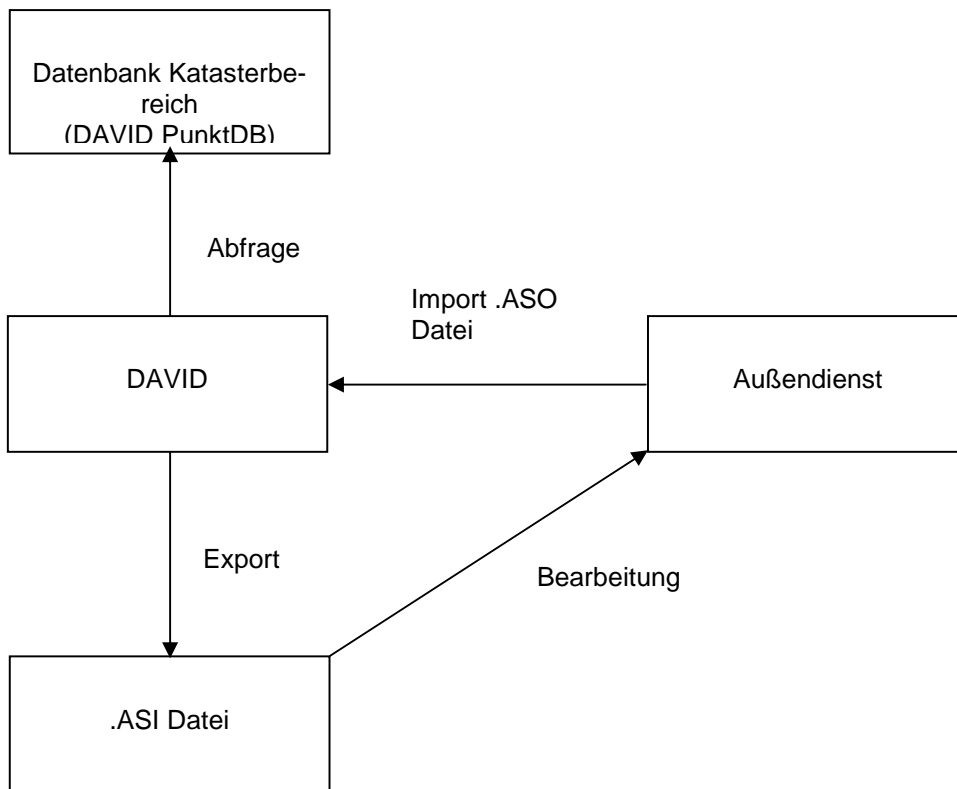


Abbildung 4: ASI Export mit ALK (Quelle: eigene Darstellung)

Durch die Umstellung zum AAA-Modell ist zukünftig kein exportieren von ASI-Dateien aus ALKIS und kein importieren von ASO-Dateien nach ALKIS möglich. ALKIS, welches ein Teil des AAA Modells ist, wird in einzelnen Komponenten für den Anwender unterteilt. Die DHK (Datenhaltungskomponente) bildet das Basisschema ab und ist die Datenbank. Die EQK (Erhebungs- und Qualifizierungskomponente) dient zum Erfassen und Prüfen von Daten. Die APK (Auskunft- und Präsentationskomponente) ist zur Präsentation und Auskunft der Informationen gedacht.

In Thüringen führt jeder Katasterbereich eine eigene DHK. Diese acht Datenerhaltungskomponenten aktualisieren eine zentrale DHK. Der Bestandsdatenauszug, als amtlichen Nachweis in XML Format, erfolgt über ein Webportal mit Hilfe der APK und der Zentralen DHK.

Die XML-Datei soll von den Vermessungsstellen, Katasterbereichen und ÖbVIs verarbeitet und kontrolliert und als XML-Datei wieder an die DHK zurückgeben werden.

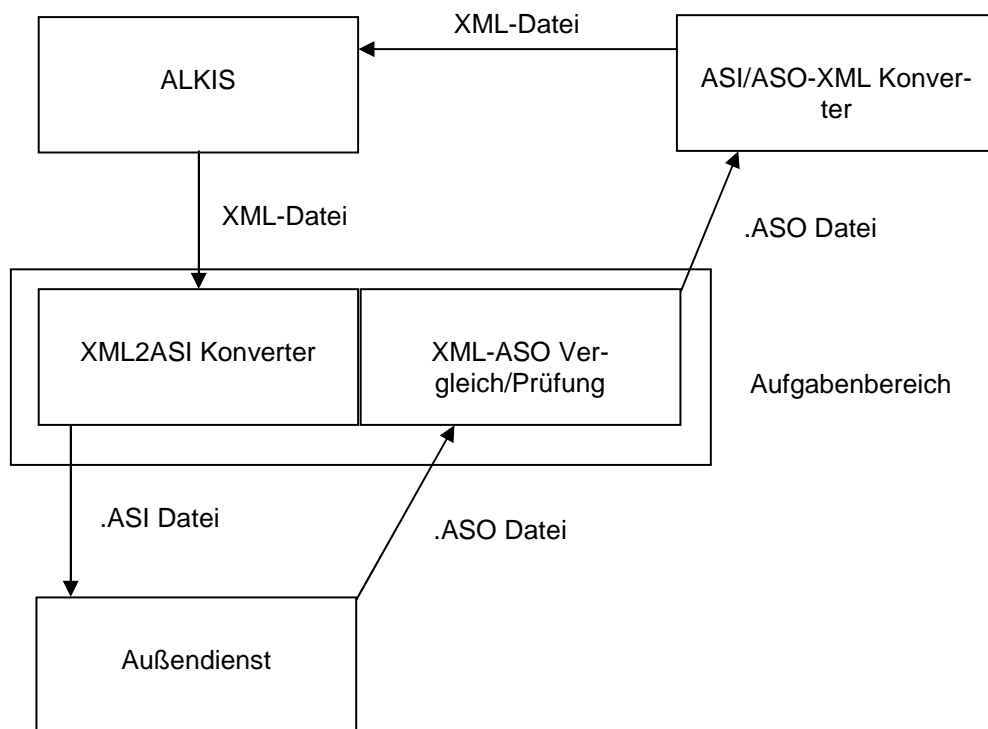


Abbildung 5: ASI Export über ALKIS (Quelle: eigene Darstellung)

Der Soll-Zustand soll ein Programm sein, welche die ALKIS Bestandsdaten aus der NAS einliest und in ein ASI Format umwandelt. Nach der Verarbeitung der ASI-Datei von Vermessungsstellen soll das Programm das abgegebene ASO Format mit den passenden Bestandsdaten vergleichen und eine ASO ausgeben.

Der Ist-Zustand bedeutet, dass die Bestandsdaten in der Datenhaltung im beschriebenen XML Format vorliegen und für die Datenführung ASI benötigt wird. In der Umkehrung sind die Fortführungsdaten in ASO Format, aber es wird XML benötigt.

Die Zusammenhänge und Struktur der Formate wurden bereits dargelegt (siehe Kapitel 3. und 4.)

Das Programm „MW-Konverter“ besteht aus zwei Unterprogrammen. In dem ersten, „XML2ASI“, findet das Umwandeln von XML in ASI statt und im zweiten, „XML-ASO“, der Vergleich zwischen XML und ASO-Dateien. Beide Unterprogramme besitzen unterschiedliche Anforderungen und somit verschiedene Kriterien.

7.1.1 Anforderungen „XML2ASI“

Als erstes ist es nötig, die Bestandsdaten im XML Format einzulesen. Die eingelesenen Fachinformationen müssen nach der ASI/ASO Formatbeschreibung umgewandelt und im ASI Format gespeichert werden. Zu beachten ist, dass sich das Format verändern kann

(siehe Kapitel 4). Der zweite Punkt ist eine übersichtliche GUI, welche die Fachinformationen anzeigt. Bei dieser Anzeige soll es möglich sein, die einzelnen Werte zu editieren. Der letzte Punkt der Anforderungen ist das Festlegen der Regeln zur Rückmigration. Diese Regeln beschreiben welche Kodierung die Werte in der XML-Datei in ASI/ASO Format annehmen. Ein nachträgliches Ändern der Werte Zuweisung ist nötig, da sich beide Formate in der Entwicklung befinden und somit ändern können.

Bei diesem Unterprogramm ist eine Protokollausgabe nicht nötig, weil kein Vergleich zwischen unterschiedlichen Datensätzen vorkommt.

7.1.3 Anforderungen „XML-ASO“

Dieses Unterprogramm liest sowohl Bestandsdaten im XML Format als auch Fortführungsdaten im ASO Format ein, mit dem Ziel beide Dateien zu vergleichen. Nach dem Vergleich sollen die Vermessungspunkte in einer übersichtlichen GUI dargestellt und ein Protokoll ausgegeben werden. Das Protokoll ist nötig, um eine Archivierung des Vergleiches zu gewährleisten, ohne eine Datenbank zu entwickeln. Die Darstellung der Fachinformationen unterliegt der ASI/ASO Formatbeschreibung (siehe Kapitel 6). Um Veränderungen und Neupunkte nach dem Vergleich zu erkennen, sollen diese in der GUI markiert werden.

Des Weiteren sollen die Fortführungsdaten editierbar sein. Das Editieren der Werte zieht eine Konsistenzprüfung nach sich, damit unzulässige Kodierungen der Werte verhindert werden können. Das Speichern der Fortführungsdaten erfolgt als ASO-Datei.

7.2 Phase des Entwurfes

In der Phase des Entwurfes entsteht eine Beschreibung der Software aus der Sicht des Entwicklers und wie die dargelegten Anforderungen erreicht werden können. Um dies zu realisieren, wird ein Entwurf der Umsetzung des Programms erstellt. Das Ergebnis dieser Phase ist eine Entwurfsspezifikation.

7.2.1 Aufbau des Programms

Die zwei Unterprogramme „XML2ASI“ und „XML-ASO“ werden in Komponenten unterteilt. In den Komponenten sind Klassen mit Methoden enthalten, welche die einzelnen Anforderungen realisieren. Einige Komponenten, wie das Einlesen der XML-Datei und das Umwandeln der Informationen in ASI/ASO Format, werden für beide Unterprogramme benötigt und dementsprechend entwickelt.

Die GUI Komponenten rufen die jeweils einzelnen Methoden auf, welche die Aufgaben erfüllen und zeigen das Ergebnis in einem JAVA Fenster (JFrame).

Da ein Vermessungspunkt jeweils eine Koordinate in den zwei CRS besitzen kann, ist eine Trennung und separate Bearbeitung für ETRS89/UTM und PD83/GK nötig.

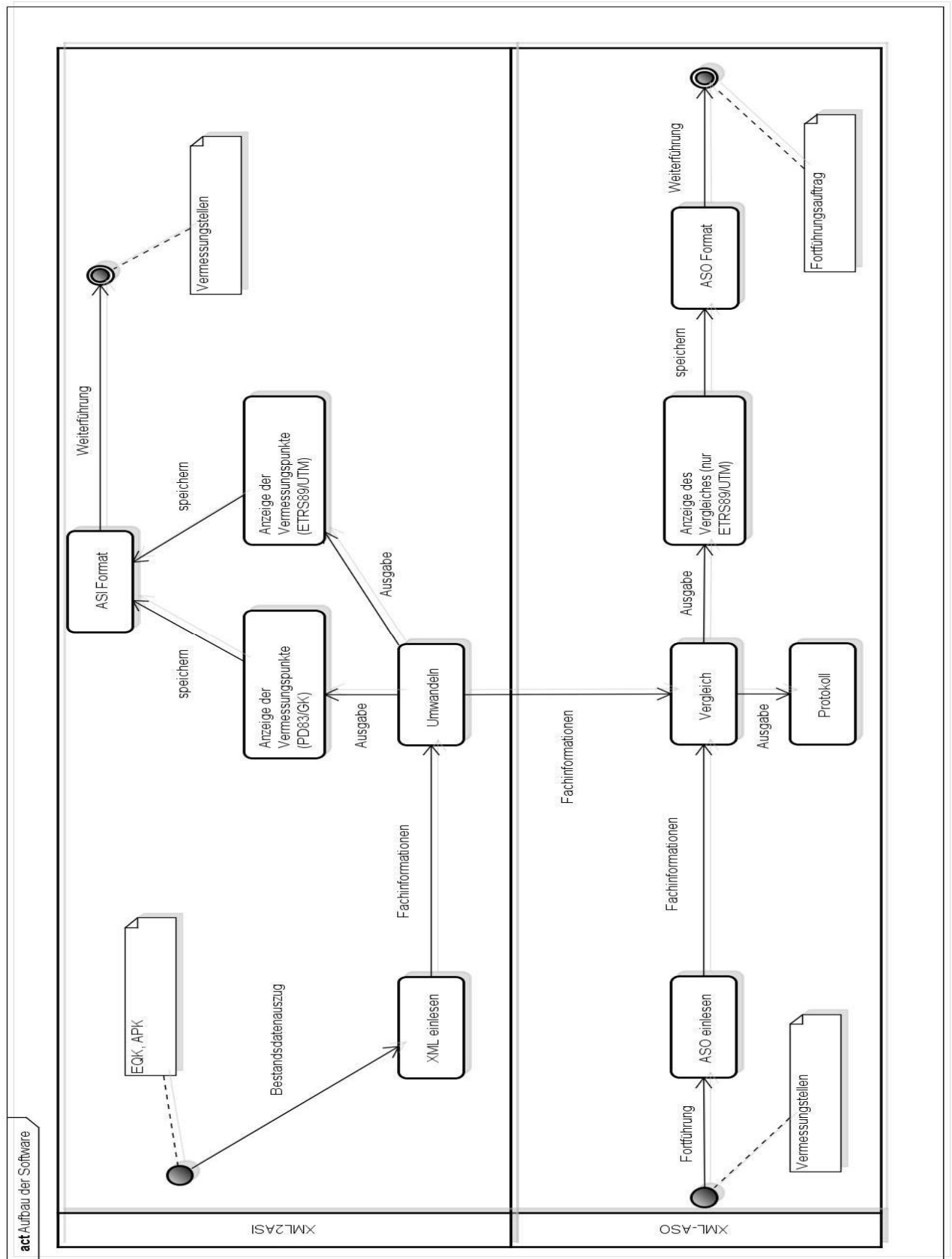


Abbildung 6: Umsetzung des Systems (Quelle: eigene Darstellung)

7.2.2 Entwurf des Programms

Die Trennung der zwei CRS erfolgt durch das Erstellen eines Objektes. Das Objekt speichert die Vermessungspunkte mit ETRS89/UTM bzw. PD83/GK und kann durch die jeweilige *get* Methode abgerufen werden. Die ASI/ASO Attribute werden in Form von Listen in ein Objekt abgelegt, da sie immer wieder verwendet werden.

7.2.2.1 „XML2ASI“ Entwurf

Für das Unterprogramm „XML2ASI“ sind folgende Komponenten nötig:

- XML-Modul (XML einlesen)
- Werte-Modul (Werte umwandeln)
- GUI XML2ASI

Die Komponente zum XML-Datei einlesen enthält eine Methode, welche die benötigten Objektarten, Attribute und Werte (siehe Tabelle 18, 19) herausfiltert. Die gefilterten Informationen werden in ETRS89 und PD83 Vermessungspunkte getrennt und in eine Liste zum Weiterverarbeiten gespeichert.

In der Komponente *Werte-Modul* ist für jedes Attribut der ASI/ASO-Datei eine eigene Methode vorhanden. Der Vorteil daran ist, dass bei Änderungen der ASI/ASO Formatbeschreibung gezielt die Methode angepasst werden kann.

Die GUI Komponente besteht aus zwei Textfeldern (JTextArea) und einer Bedienerleiste zum einlesen und speichern der umgewandelten Informationen. In dem ersten Textfeld stehen die Vermessungspunkte mit UTM Koordinaten und im zweiten Textfeld die mit GK Koordinaten.

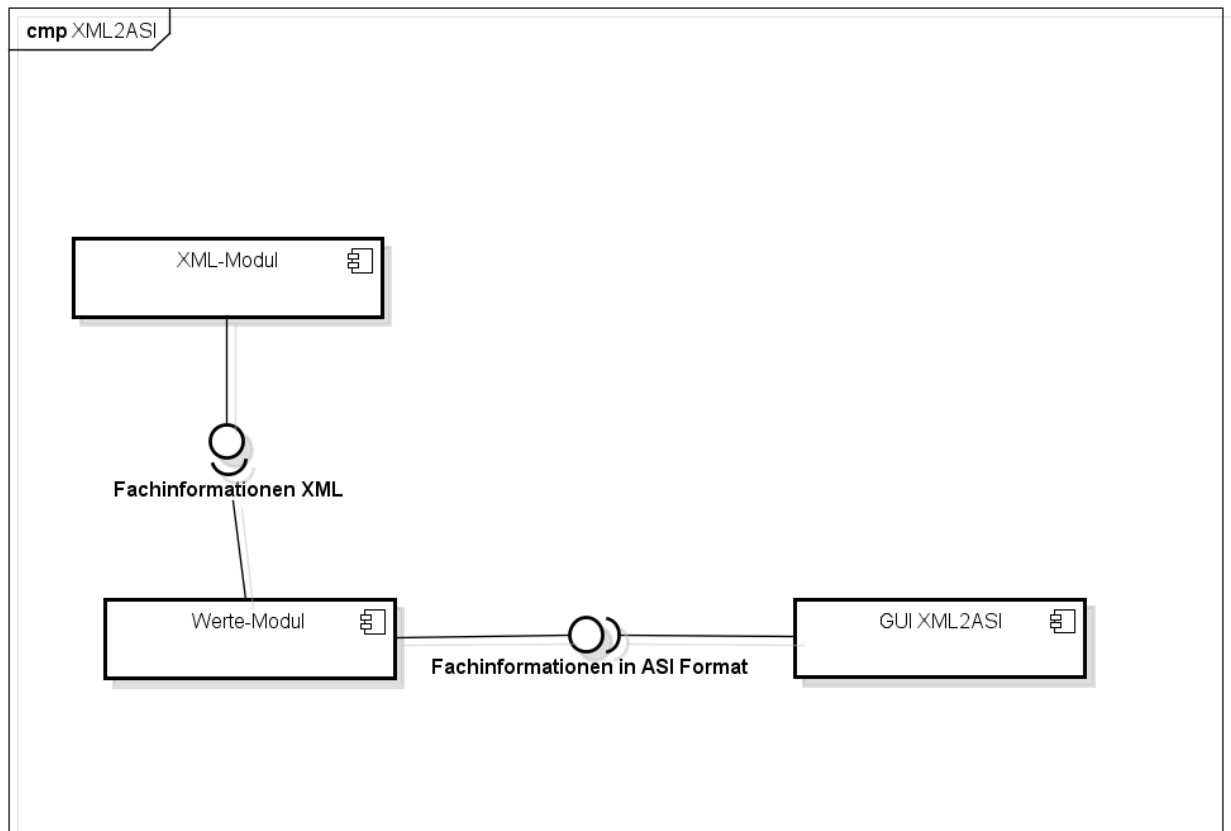


Abbildung 7: Komponenten XML2ASI (Quelle: eigene Darstellung)

7.2.2.2 „XML-ASO“ Entwurf

Das Unterprogramm XML-ASO soll folgende Komponenten haben:

- XML-Modul (XML einlesen)
- ASO-Modul (ASO einlesen)
- Werte-Modul (Werte umwandeln)
- Vergleichs-Modul
- GUI XML-ASO

Die Komponenten zum XML- Datei einlesen und Werte umwandeln sind vom Unterprogramm „XML2ASI“ und werden wieder verwendet. Die ASO-Datei soll zeilenweise eingelesen und in eine Liste gespeichert werden. Hierbei ist die Trennung von den unterschiedlichen CRS zu beachten.

Bei der Komponente *Vergleichs-Modul* werden die Punktkennungen von der ASO mit der XML-Datei verglichen und das Ergebnis in eine Liste gespeichert.

Die GUI wird aus zwei Listen (JList), einer Tabelle mit den Fachinformationen des Vermessungspunktes und einer Bedienerleiste bestehen. Die GUI besitzt zudem eine Methode um csv (Comma Seperated Value) Dateien auszugeben, welche das Protokoll darstellen. Die Anzeige von Veränderungen erfolgt über das Einfärben der Tabellenzeilen und Listeneinträge.

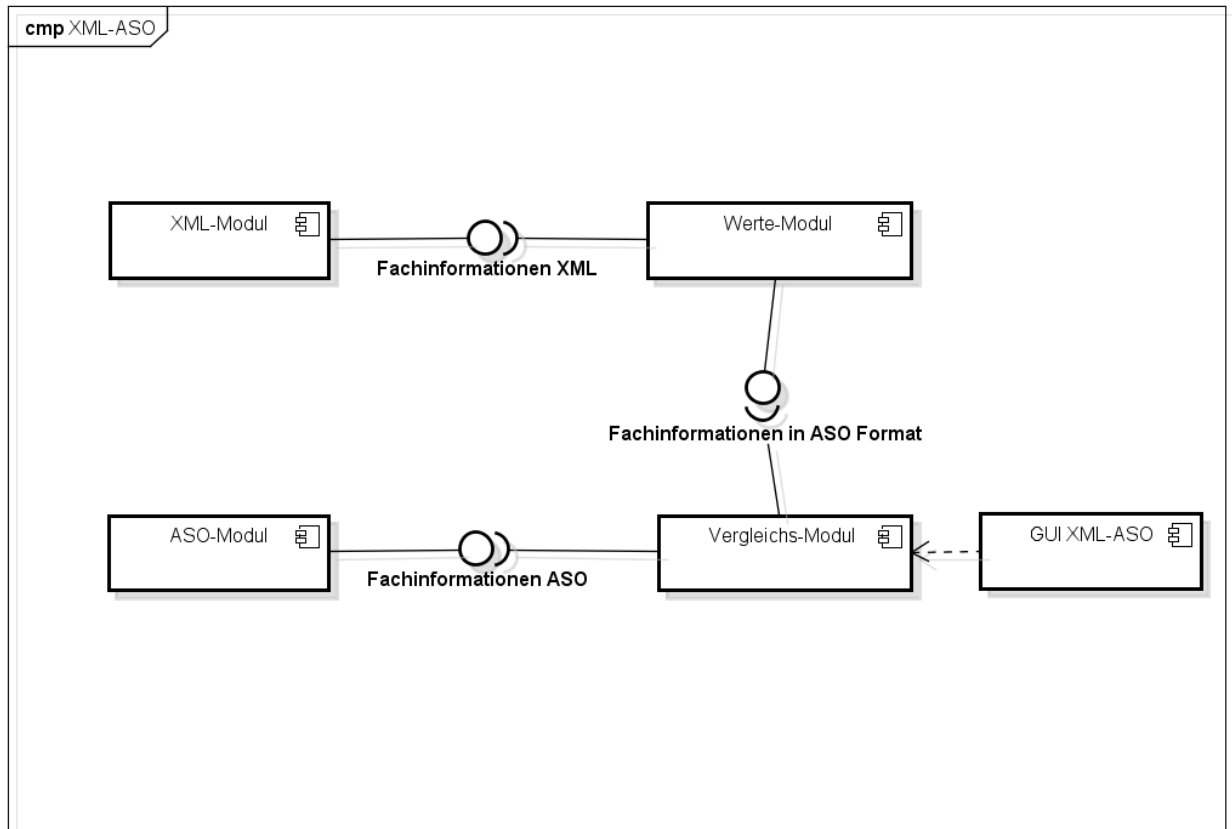


Abbildung 8: Komponenten XML-ASO (Quelle: eigene Darstellung)

7.3 Phase der Implementierung

Die Phase der Implementierung realisiert den Entwurf im Detail und setzt dessen Semantik um. Es handelt sich um einen Feinentwurf, mit ersten Tests der einzelnen Programmkomponenten und Fehlerbeseitigung. Das Resultat ist der Quellcode von einzelnen Komponenten und deren Testen auf die Lauffähigkeit.

7.3.1 „XML2ASI“ Feinentwurf

XML-Modul

Die Komponente zum XML einlesen ist in der Klasse *XML_einlesen* und enthält eine Methode, welche auf StAX basiert. Es gibt verschiedene Methoden XML Dokumente einzulesen, z.B. DOM, SAX und StAX. StAX durchläuft das XML Dokument sequentiell und das Anwendungsprogramm ruft iterativ den Parser auf. Es handelt sich also um Pull-Parsing. Mit Hilfe von StAX werden die benötigten Objektarten (siehe Tabelle 15) anhand des Starttags gesucht und in einen StringBuffer gespeichert. Zusätzlich sucht die Methode die Attribute und Wertarten heraus (siehe Tabelle 16) und speichert sie in den Buffer. Der Buffer wird mit Hilfe eines Trennzeichens in ein Datenarray umgewandelt, um mit regulären Ausdrücken weiter arbeiten zu können.

Der nächste Schritt in der Methode ist das Filtern der REO. Jede REO Klasse (siehe 2.1.2) wird in einem Punktortarray gespeichert. Eine Suche nach Punktnummern in dem Datenarray filtert die ZUSO heraus, welche nur ein REO besitzen und somit grafische Punkte sind. Das Datenarray mit den ZUSOs und das Punktortarray mit den REOs werden miteinander über das Attribut *istTeilVon* und den Wert des Identifikators verglichen. Während des Vergleiches wird eine Trennung der Vermessungspunkte nach ihren CRS vorgenommen. Das Ergebnis sind zwei Arrays, getrennt nach ihren CRS mit den ZUSO und passenden REOs.

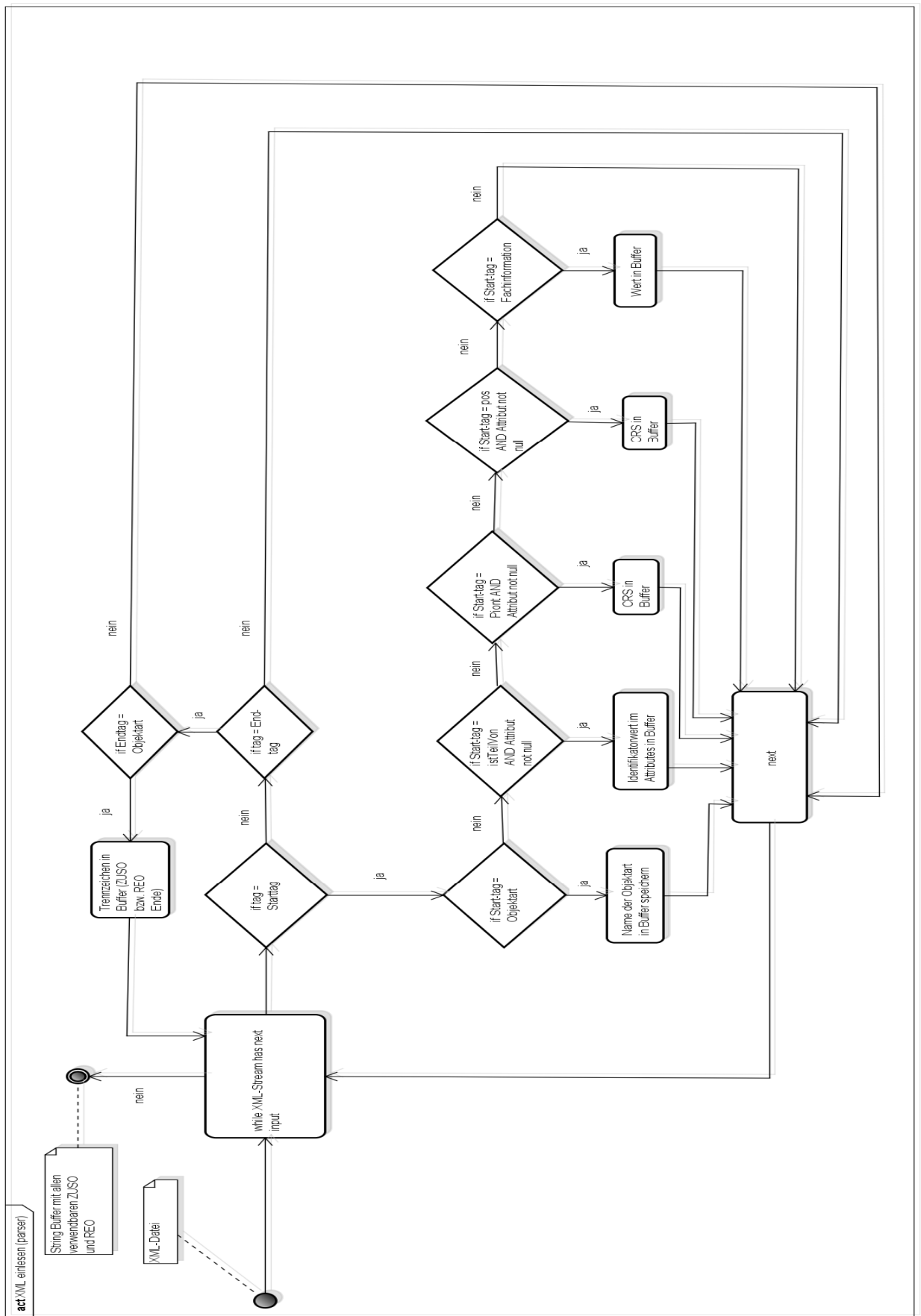


Abbildung 9: XML-Modul (parser) (Quelle: eigene Darstellung)

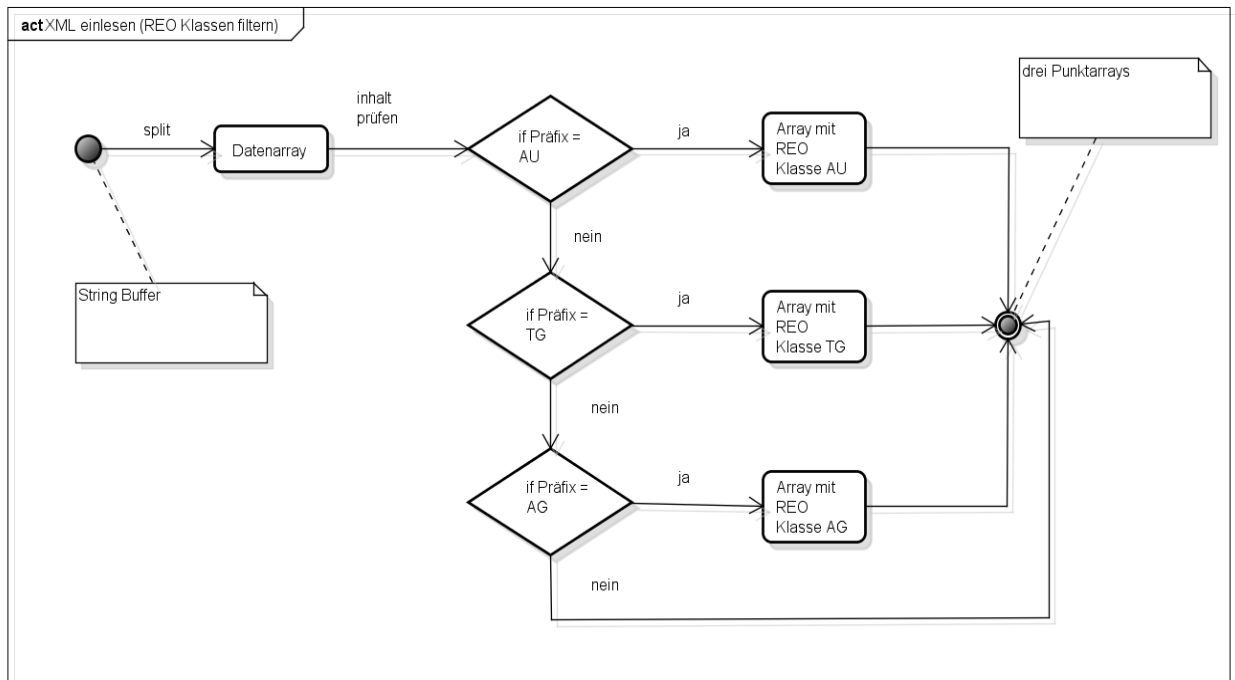


Abbildung 10: XML-Modul (REO Klassen filtern) (Quelle: eigene Darstellung)

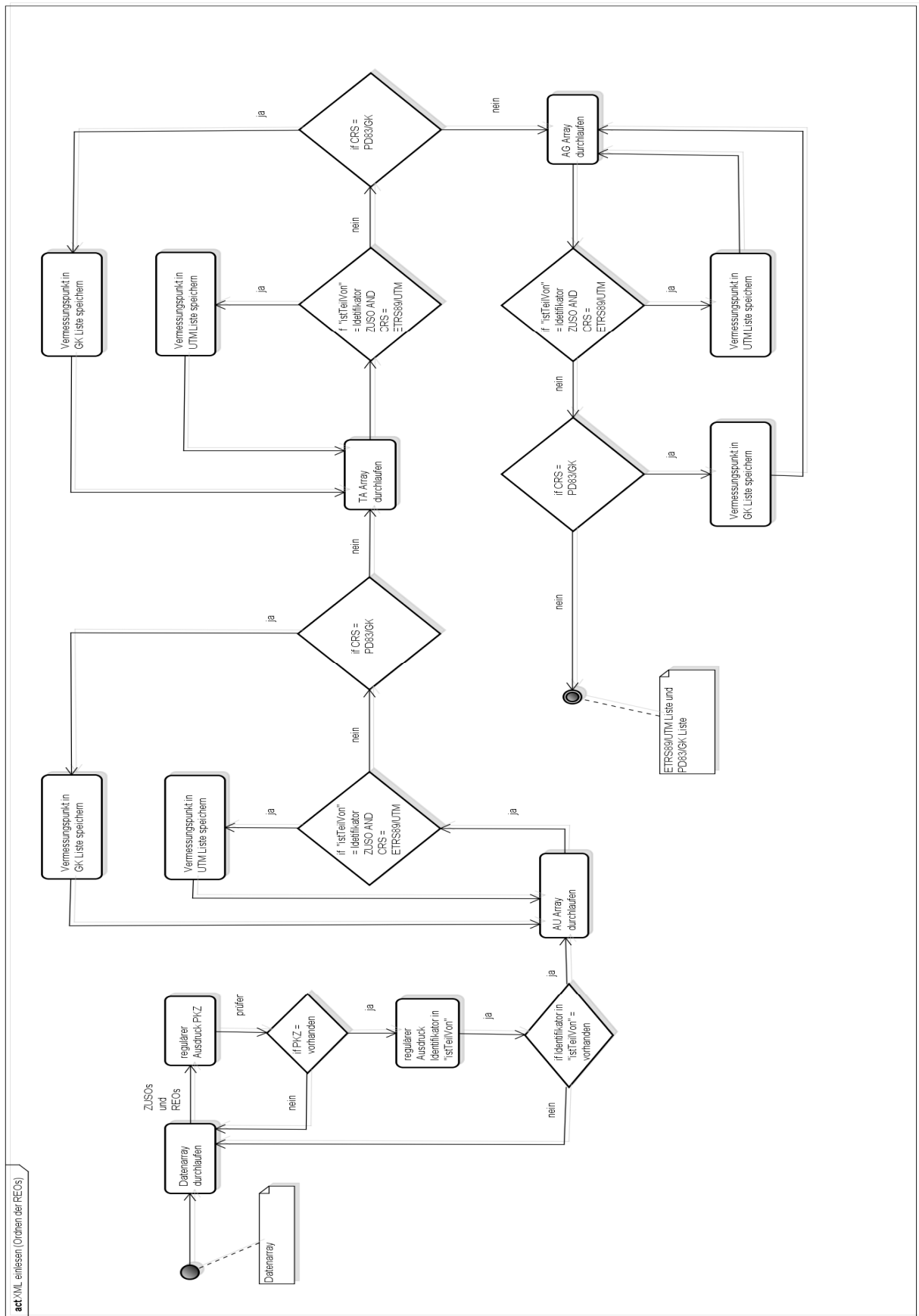


Abbildung 11: XML-Modul (ordnen der REOs) (Quelle: eigene Darstellung)

Der Letzte Schritt ist das Herauslesen der gemessenen bzw. gerechneten Koordinaten des Vermessungspunktes. Es gibt zwei Möglichkeiten dies zu realisieren. Die Erste ist über die Darstellung in der Liegenschaftskarte mit dem tag *kartendarstellung* und die Zweite ist über die Genauigkeitsstufe im REO gegeben.

Bei der Kartendarstellung hat das REO mit dem CRS ETRS89/UTM den Wert „false“ im tag *kartendarstellung* immer die gemessenen bzw. gerechneten Koordinaten. Falls nur ein REO mit dem CRS ETRS89/UTM vorhanden ist, werden dessen Koordinaten, unabhängig von dem Wert in der Kartendarstellung verwendet (siehe Tabelle 10, 11, 12). Die zweite Variante verwendet das REO, bei denen die Koordinaten die höchste Genauigkeit aufweisen.

Die Wahl fiel auf die Darstellung in der Liegenschaftskarte, weil es von der Performance das schnellere Verfahren ist. Hierfür muss zuerst ermittelt werden, welche ZUSO besitzen nur ein REO mit den ETRS89/UTM Koordinaten und somit der Kartendarstellung „true“. Mit Hilfe von regulären Ausdrücken werden die Identifikatoren des REO mit ETRS89/UTM in eine „true“- und „false“-Liste getrennt. Als zweites findet ein Vergleich zwischen beiden Listen statt. Falls die Identifikatoren in beiden Listen vorhanden sind, werden diese von der „true“ Liste subtrahiert. Als Ergebnis bleiben nur die REO übrig, welche in der Kartendarstellung den Wert „true“ haben. Anschließend kann mit regulären Ausdrücken jeder Vermessungspunkt auf die Kartendarstellung „false“ durchsucht werden. Sollte der Wert „false“ nicht vorhanden sein, werden die Koordinaten aus dem REO in der subtrahierten „true“-Liste verwendet.

Die Vermessungspunkte mit den passenden Koordinaten werden in ein Objekt als String Array gespeichert, welches sie in ETRS89/UTM und PD83/GK unterteilt.

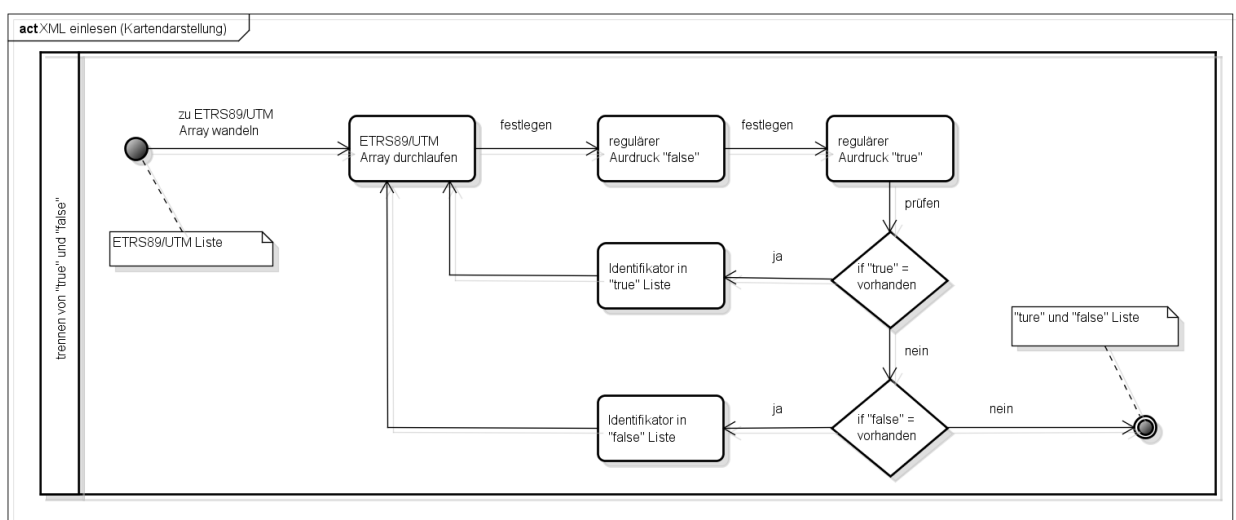


Abbildung 12: XML-Modul (Kartendarstellung) (Quelle: eigene Darstellung)

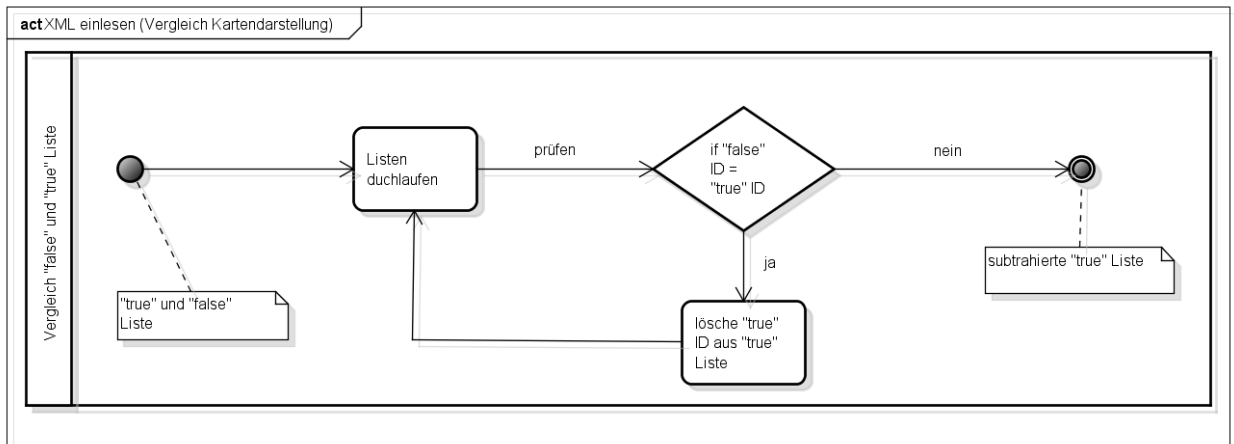


Abbildung 13: XML-Modul (Vergleich Kartendarstellung) (Quelle: eigene Darstellung)

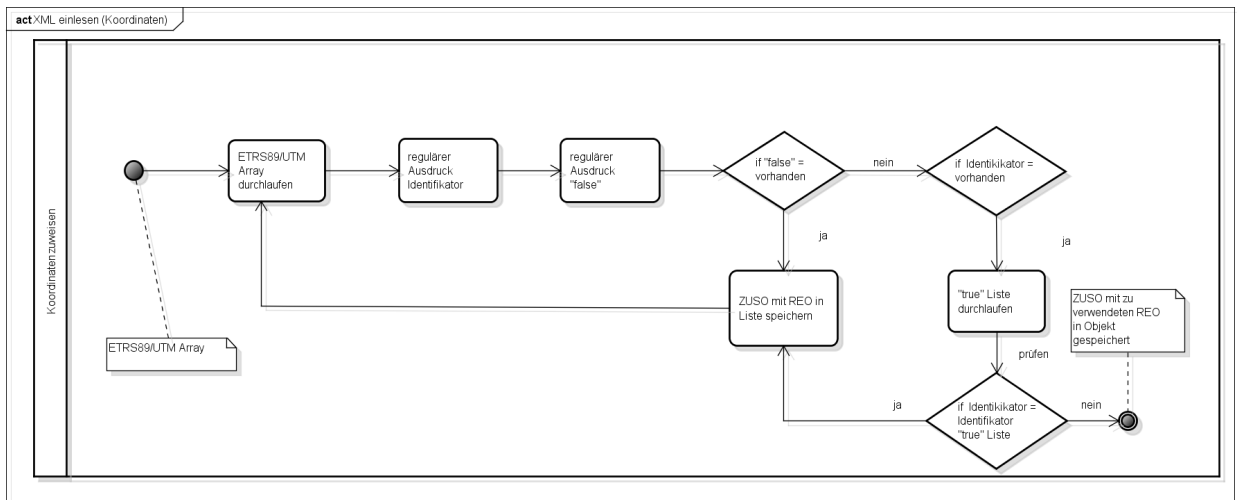


Abbildung 14: XML-Modul (Koordinaten) (Quelle: eigene Darstellung)

Der Test der Komponente erfolgte mit verschiedenen Bestandsdatenauszügen aus der EQK. Um das Fehlverhalten zu erschließen, wurden die XML-Dateien um tags erweitert, zusätzliche Wertarten eingefügt und sowohl ältere als auch neuere Bestandsdaten verwendet. Die Komponente hat jede Datei erfolgreich geladen und fehlerhafte, wie z.B. nicht wohlgeformte, verworfen.

Werte-Modul

Die Komponente, um die Werte umzuwandeln ist in der Klasse *Werte_umwandeln* enthalten und besitzt verschiedene Methoden und Steuerdateien.

Die Rückmigration der einzelnen Kodierungen von XML Format zu ASI/ASO Format wird in Steuerdateien festgelegt. Diese Auslagerung hat den Vorteil, dass eine einfache Anpassung der

Werte möglich ist, ohne den Quellcode zu ändern. Für folgende ASI/ASO Attribute sind Steuerdateien nötig:

PAT, LST, LGS, VAT, HER, PGW-PD83/GK, PGW-ETRS89/UTM, OSK-Grenzpunkt

XML Wert	ASI/ASO Wert
key (Schlüssel)	value (Wert)

Tabelle 20: Aufbau Steuerdatei (Quelle: eigene Darstellung)

Die restlichen ASI/ASO Attribute werden direkt aus der XML-Datei gelesen (für Positionen siehe Tabelle 19).

Der Übergabeparameter für die Methoden der einzelnen Attribute des ASI/ASO Formates ist das ETRS89/UTM-PD83/GK Objekt und gegebenenfalls die passende Steuerdateien. Die Methoden verwenden reguläre Ausdrücke um die gewünschte Position im String Array des Objektes zu finden und dann in eine Liste zu speichern. Sollte zu dem Attribut eine Steuerdatei vorhanden sein, wird diese gelesen und der zugeteilte Wert in die Liste gespeichert.

Die Methoden OSK-, VAT-, AKZ-umwandeln und Datum haben Besonderheiten. Der OSK für die ZUSO *AX_BesondererGebauepunkt* und *AX_BesondererBauwerkspunkt* ist unabhängig von der VAT immer 0530151. Die gleiche Bedingung hat der *AX_Sonstigevermessungspunkt* mit dem OSK 0510122, der *AX_Aufnahmepunkt* mit dem OSK 0510121 und der *AX_Sicherungspunkt* mit dem OSK 0510125. Eine ähnliche Rückmigration gilt bei den VAT von einigen ZUSOs. Die VAT beim *AX_BesondererGebauepunkt* und *AX_BesondererBauwerkspunkt* ist immer 070. Das Format des Datums in der XML-Datei entspricht nicht dem Format in der ASI/ASO-Datei und muss umgewandelt werden. Als erstes sollte mit Hilfe von einem regulären Ausdruck das Datum im Objekt ETRS89/UTM-PD83/GK gesucht werden. Die gefundenen Datumsangaben speichert die Methode in eine Liste. Als Zweites kann das JAVA Paket, mit dem Namen *SimpleDateFormat*, das gewünschte Datumsformat aus der Liste formatieren. Beim AKZ werden die Stellen auf acht angepasst, bei zu wenig Stellen durch auffüllen mit Leerzeichen und bei zu vielen Stellen durch Abschneiden der ersten beiden Positionen des Wertes.

Das Ergebnis der Methoden wird in ein Objekt ETRS89/UTM-PD83/GK gespeichert, welches eine Trennung der CRS ermöglicht.

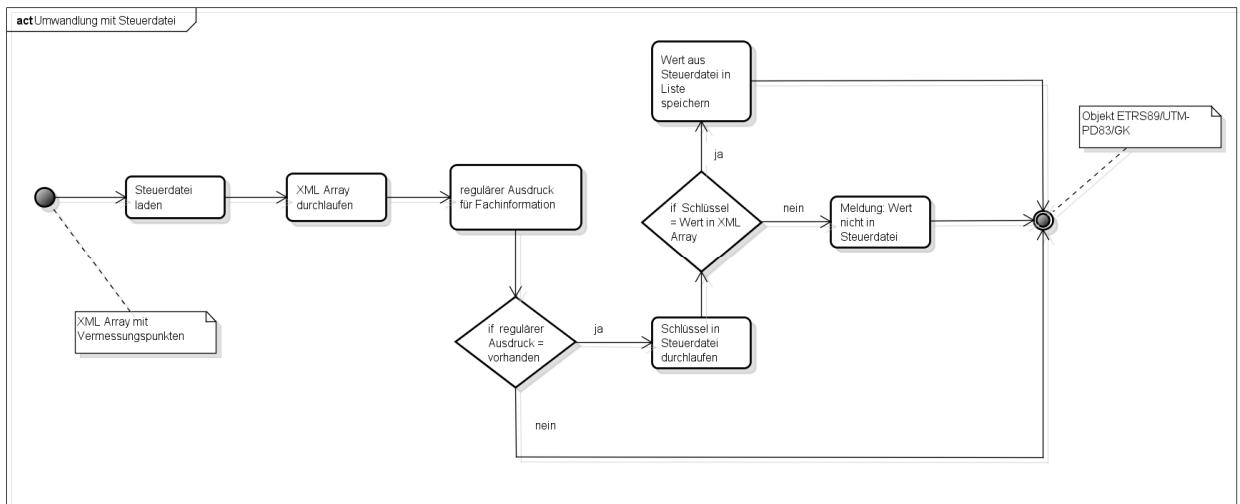


Abbildung 15: Methoden zur Umwandlung mit Steuerdatei (Quelle: eigene Darstellung)

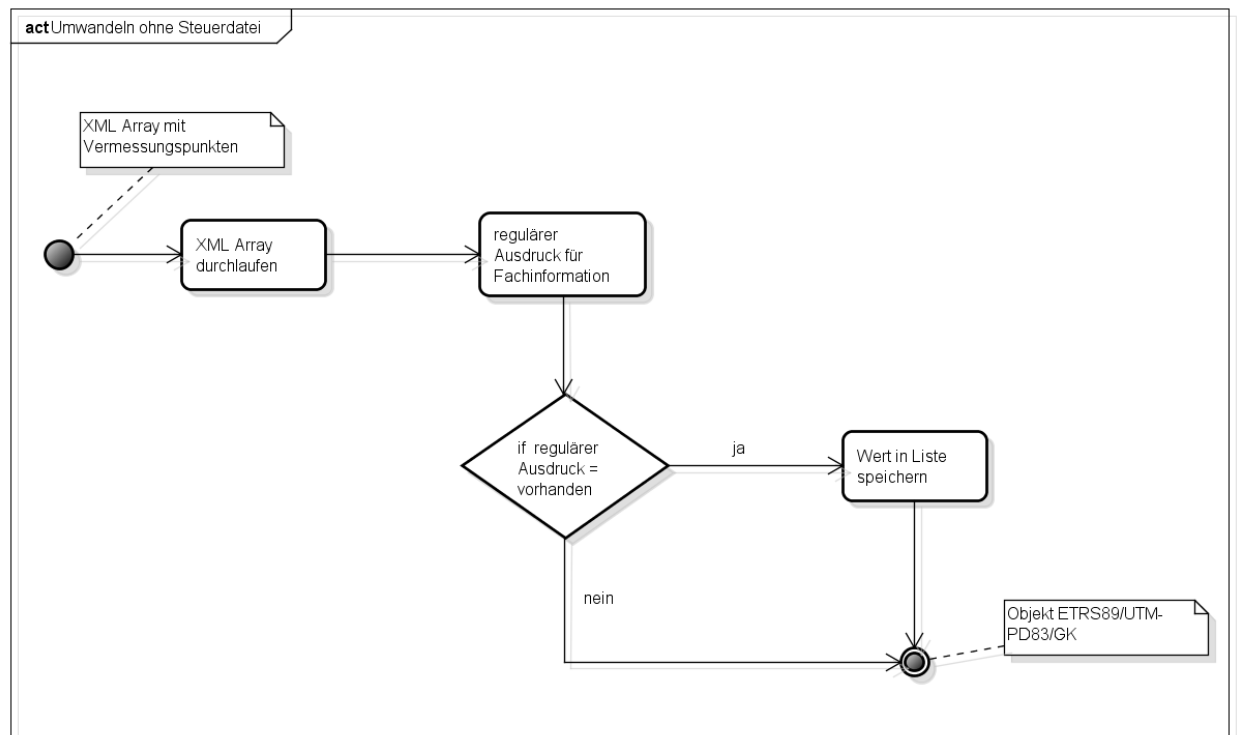


Abbildung 16: Methoden zur Umwandlung ohne Steuerdatei (Quelle: eigene Darstellung)

Der Test dieser Komponente erfolgte mit denselben Testdateien die beim Einlesen der Komponente XML einlesen angewandt wurden. Einen Fehler ergab sich beim Auslesen des AKZ. Das AKZ unterliegt keiner festen Konvention und kann relativ beliebig von den Katasterämtern festgelegt werden. Ältere AKZ bestehen aus Zahlen, Buchstaben und Sonderzeichen. Ein Anpassen des regulären Ausdrucks fing diesen Fehler ab.

GUI XML2ASI

Die GUI für die Programmkomponente wird mit dem JAVA Swing Packet realisiert. Die GUI ist als Konstruktor konzipiert. Der Konstruktor enthält zwei *JTextAreas*, je eine für Vermessungspunkte mit dem CRS ETRS89/UTM und PD83/GK. Damit eine Bedienung möglich ist, enthält der Konstruktor eine Menüleiste und eine Toolbar. Beide besitzen dieselben Bedienelemente zum Laden einer XML-Datei und zum Speichern in das ASI/ASO Format.

Das Bedienelement *offnen* ist ein JAVA ActionEvent und dient zum Laden der XML-Datei. Über das JAVA Packet *JFileChooser* wird zuerst ein Fenster zum Datei auswählen geöffnet. Als Zweites folgt der Aufruf der Klasse *XML_einlesen* und anschließend der Klasse *Werte_umwandeln*. Der letzte Schritt ist die Ausgabe der umgewandelten Fachinformationen in den jeweiligen *JTextAreas*.

Das Bedienelement *speichern* schreibt je nach Wahl den Inhalt der *JTextArea* von ETRS89/UTM oder PD83/GK an einem gewählten Speicherort. Der Speicherort wird über *JFileChosser* gewählt und kontrolliert, ob die Endung ASO vom Nutzer vergeben wurde. Sollte das nicht passiert sein, wird die Endung an den Dateinamen angehängt.

Beim Testen der GUI ergab sich ein Mangel. Das Laden von Byte intensiven Dateien dauert eine gewisse Zeitspanne. Da die menschliche Aufmerksamkeitsspanne drei Sekunden beträgt und alles nach dieser Zeit als Stillstand wahrgenommen wird, war ein Ladebalken nötig. Um den Ladebalken zu realisieren wird das Laden der XML-Datei in einen eigenen Prozess (thread) ausgelagert. Solange der Prozess bearbeitet wird, ist der Ladebalken animiert. Nach Beendigung des Prozess wird die Animation gestoppt.

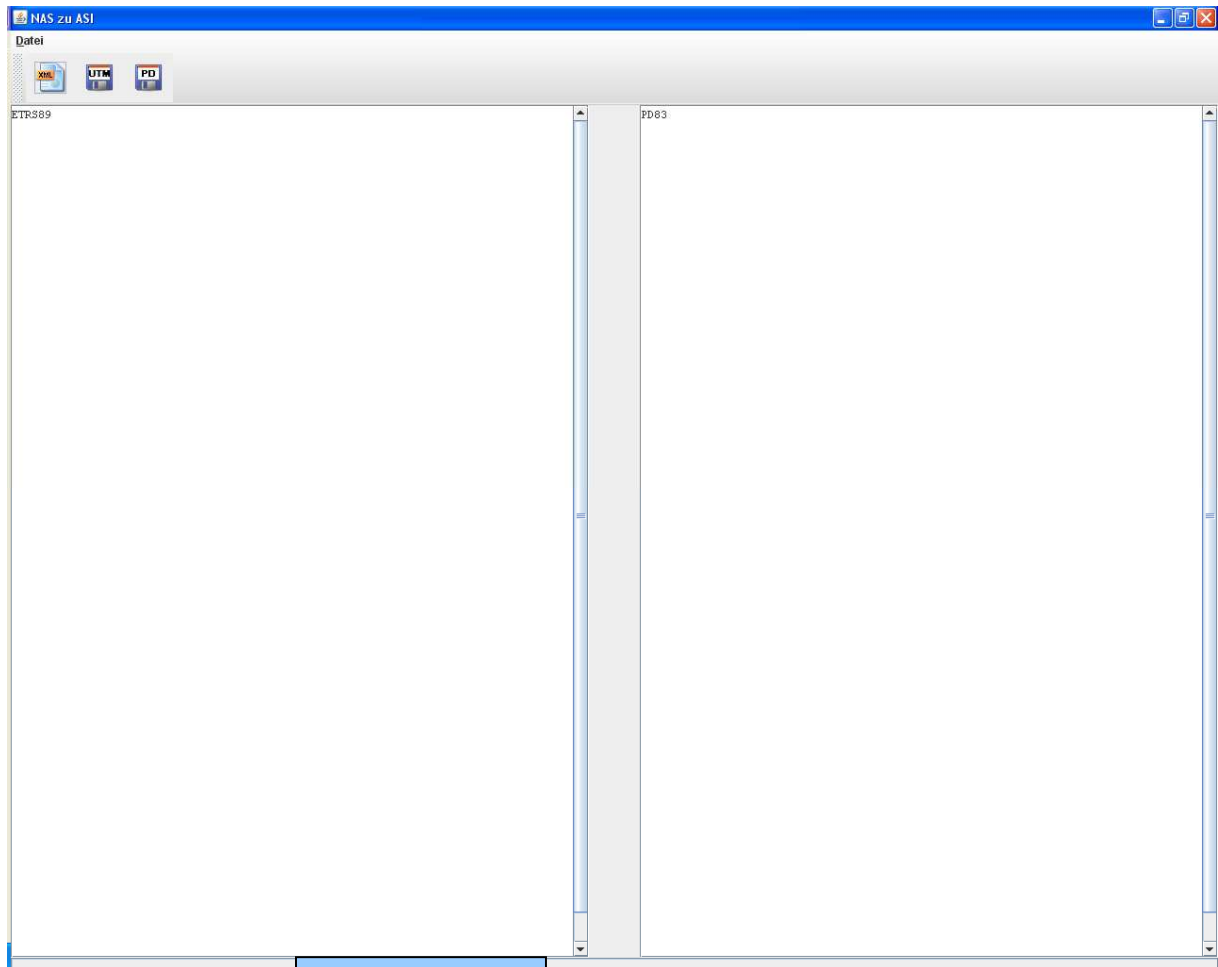


Abbildung 17: GUI XML2ASO (Quelle: eigene Darstellung)

7.3.2 „XML-ASO“ Feinentwurf

ASO-Modul

Das einlesen der ASO-Datei erfolgt durch die Klasse *datei_einlesen* und beinhaltet zwei Methoden. Die erste Methode prüft die Konsistenz der ASO-Datei, indem die Stellen in der Zeile gezählt und mit der zulässigen Zeilenlänge verglichen werden. Die zweite Methode liest die ASO mit Hilfe eines *buffered reader* zeilenweise ein. Dabei werden die Vermessungspunkte über den LST Wert in die zwei CRS getrennt und in jeweils das Objekt ETRS89/UTM-PD83/GK gespeichert.

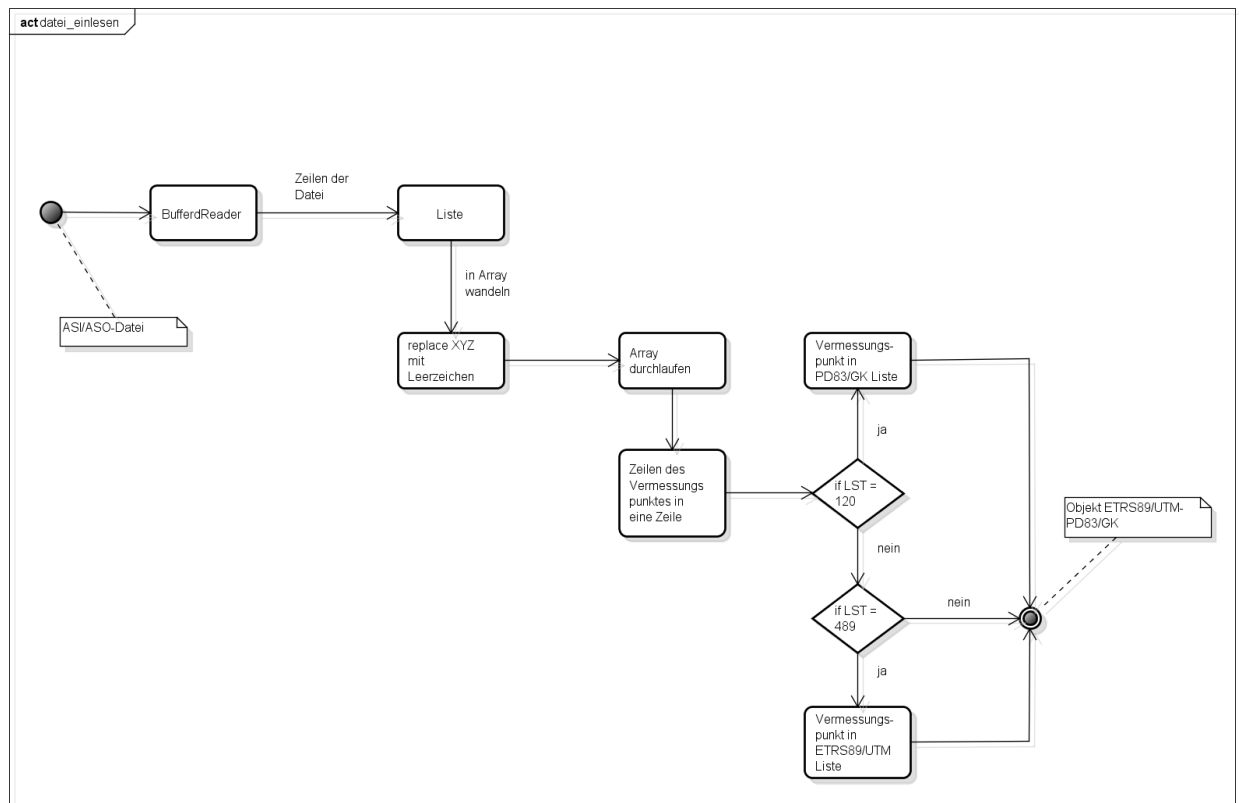


Abbildung 18: ASO Modul (Quelle: eigene Darstellung)

In der Testphase wurden mehrere ASO-Dateien eingelesen und beschlossen eine Konsistenzprüfung einzuführen, welche bei der kleinsten Abweichung im ASO Format das Einlesen unterbindet. Der einfachste Weg war die Prüfung der Zeilenlänge im ASO Format. Dadurch ist gewährleistet, dass alle Fachinformationen des Vermessungspunktes an der vorgeschriebenen Stelle sind.

Vergleichs-Modul

Der Vergleich von Punktkennungen ist in der Klasse „Vergleiche“ enthalten. Diese Klasse beinhaltet drei Methoden.

Die Methode *asx_zeilen* trennt die erste und zweite Zeile des ASO Formates im ETRS89/UTM-PD83/GK Objekt voneinander in einzelne Arrays. Diese Methode ist für die Darstellung in der GUI nötig (siehe Punkt GUI XML-ASO).

Die *xml_asx_punkte* Methode entfernt durch *hashset* alle doppelten Vermessungspunkte im Objekt ETRS89/UTM-PD83/GK. Diese Methode ist nötig, da sich in der Testphase gezeigt hat, dass es ein mehrfaches Speichern von Vermessungspunkten ergeben könnte.

In der letzten Methode, *xml_pkn_vergleich* wird mit Hilfe eines regulären Ausdrucks die Punktkennung in der XML-Datei gesucht und mit der Punktkennung der ASO-Datei (PNR+NBZ)

verglichen. Sind die beiden Punktkennungen gleich, werden die XML- und ASO Vermessungspunkte getrennt in das Objekt *Objekte_vergleich*, gespeichert.

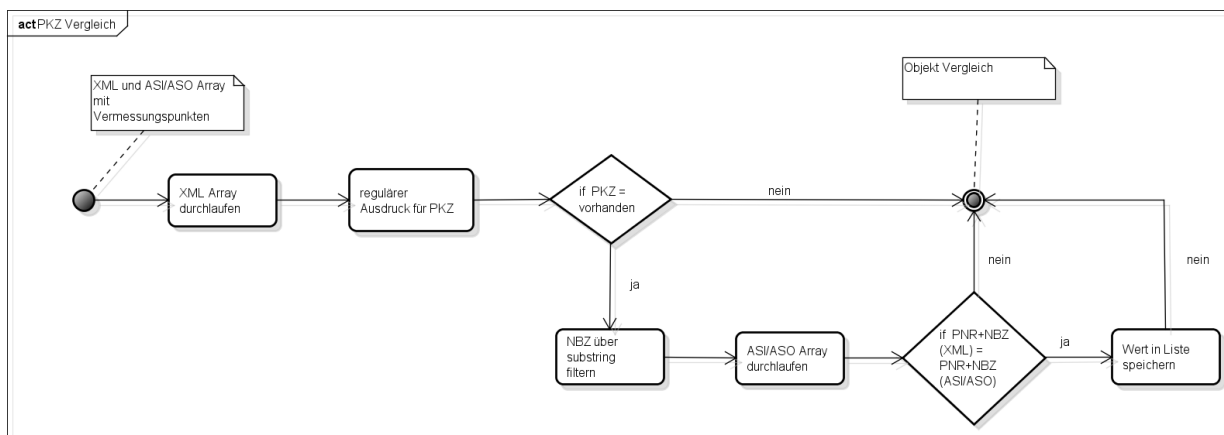


Abbildung 19: PKZ Vergleich (Quelle: eigene Darstellung)

GUI XML-ASO

Um die Anforderungen der GUI zu erfüllen, besteht sie aus mehreren Methoden, Modulen, einem Konstruktor und basiert auf dem JAVA Swing Packet. Der Konstruktor besteht aus einer Toolbar, Menüleiste, Tabelle, zwei Listen, ein Ladebalken und einem Textfeld. Mit der Toolbar bzw. Menüleiste sind die Operationen des Programms möglich:

- XML und ASO einlesen
- Vergleich der geladenen Dateien
- Protokollausgabe als CSV-Datei
- alle Werte des Bestandes in Fortführung übertragen
- einzelnen Punkt des Bestandes in Fortführung übertragen
- einzelne Werte des Bestandes in Fortführung übertragen
- Speichern als .ASO Datei
- die Listenanzeige löschen

Das Bedienelement *offnen* ist ein JAVA ActionEvent und dient zum Laden der XML- und ASO-Datei. Über das JAVA Packet *JFileChooser* wird ein Fenster zum Datei auswählen geöffnet. Anschließend wird die Dateiendung geprüft. Besitzt die geladene Datei die Endung .XML wird die Klasse *XML_einlesen* aufgerufen. Sollte die Endung .ASO oder .ASI sein, wird die Klasse

datei_einlesen aufgerufen. Der Inhalt ist somit jeweils in einem eigen ETRS89/UTM-PD83/GK Objekt (*xml*, *asx*) abgelegt. Der Inhalt der Objekte kann durch das Einlesen weiterer Dateien erweitert werden. Dadurch ist es möglich mehrere ASI/ASO-Dateien und XML-Dateien zu laden, ohne eine spezielle Reihenfolge einzuhalten.

Mit Hilfe des Bedienelementes *vergleich* wird der Inhalt der beiden Objekte (*xml*, *asx*) verarbeitet und das Ergebnis in den zwei JListen dargestellt. Zunächst erfolgt ein Vergleich des PKZ in beiden ETRS89/UTM-PD83/GK Objekten *xml* und *asx*, durch den Aufruf der Klasse *Vergleiche*. Das Ergebnis wird im Objekt *Objekte_vergleich* gespeichert. Es enthält alle Vermessungspunkte die sowohl in der XML-Datei als auch in der ASI/ASO-Datei vorhanden sind. Aufgrund dessen wird vermieden, dass die Bestandsdatenliste alle Vermessungspunkte anzeigt, wenn nur einige gebraucht werden. Als nächstes müssen die Fachinformationen aus der ASI/ASO-Datei gefiltert werden. Da sich die ASI/ASO-Fachinformationen auf zwei Zeilen mit fester Position verteilen, wird der Inhalt des Objektes *Objekt_vergleich*, mit Hilfe der Methode *asx_zeilen* aus der Klasse *Vergleiche*, in zwei Array zerlegt. Eines für die erste Zeile und eines für die zweite Zeile. Nun werden beide Arrays durchlaufen und die Fachinformationen per JAVA String Methode *substring* in ein Objekt *Objekt_asx_werte* (*list_werte_aso*) gespeichert. Das Trennen der Zeilen hat den Vorteil, dass bei Veränderungen in der ASI/ASO Formatbeschreibung eine übersichtliche Anpassung beim *substring* vorgenommen werden kann, ohne die Stringlänge abzuzählen. Zu beachten ist nur, dass JAVA bei der Position null beginnt und die ASI/ASO Formatbeschreibung bei eins. In *list_werte_aso* sind nun die PNR und der NBZ der ASI/ASO-Datei enthalten, welche in ein Listenmodell mit der Bezeichnung *utm_aso* hinzugefügt werden.

Die Fachinformationen der XML-Datei im Objekt *Objekt_vergleich* werden mit der Klasse *Werte_umwandeln* in das ASI/ASO Format konvertiert und in ein Objekt *Objekt_asx_werte* (*list_werte_xml*) gespeichert. Der Inhalt von *list_werte_xml* ist unter anderen die PNR und der NBZ der XML-Datei, welche in das Listenmodell *utm_bestand* hinzugefügt werden.

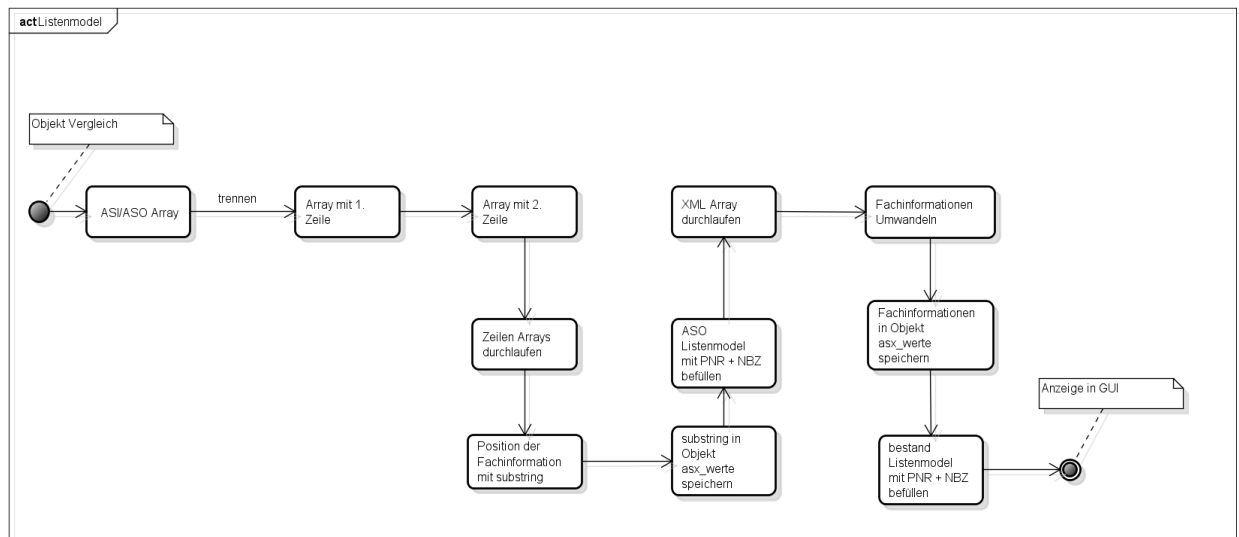


Abbildung 20: utm_aso- und utm_bestand Modell füllen

Der Aufbau der Listmodelle ist modular. Die Module separieren die eigentliche Listenkomponente vom Hinzufügen und Anzeigen der Einträge sowie von Events wie der Selektion. In den beiden Listenmodellen *utm_aso* und *utm_bestand* sind die Listenkomponenten (JList) *list_utm_aso* und *list_utm_bestand* enthalten. Der Inhalt der JListen wird farblich markiert um Neupunkte und veränderte Vermessungspunkte zu erkennen. Neupunkte sind in der *list_utm_aso* Liste gelb und veränderte Vermessungspunkte in der *list_utm_bestand* Liste blau markiert. Um die Einträge zu färben wird die GUI XML-ASO um das Modul *GUI_list_renderer* erweitert. Das *GUI_list_renderer* Modul beinhaltet je eine Methode für die JListen, um die Anzeige zu verändern. Die Methoden haben folgende Parameter: JListe, Einträge in der JListe als Objekt (*value*), Index des Listeneintrages (*index*), Ausgewählter Eintrag (*select*) und ob der Eintrag Fokussiert wurde (*fokus*). Um die Neupunkte zu filtern ist es nötig die PKZ Einträge der *list_utm_aso* Liste mit der *list_utm_bestand* Liste zu vergleichen. Hierzu wird die Methode *vergleich_liste_pkz* in der GUI XML-ASO aufgerufen. Diese Methode durchläuft die Einträge der beiden JListen und vergleicht diese auf Übereinstimmung. Ist eine Übereinstimmung vorhanden, werden diese Einträge entfernt und die Übrigen in einem String Buffer gespeichert. Dieser String Buffer wird von der ersten Methode im *GUI_list_renderer* genutzt. Sollte der String Buffer die Werte des Objektes *value* enthalten, werden diese Werte in der *list_utm_aso* gelb gefärbt und ansonsten weiß. Die Methode behandelt auch die Farbe bei einer Selektion. Beim Anklicken eines Eintrages in der Liste erhält dieser die Farbe grau.

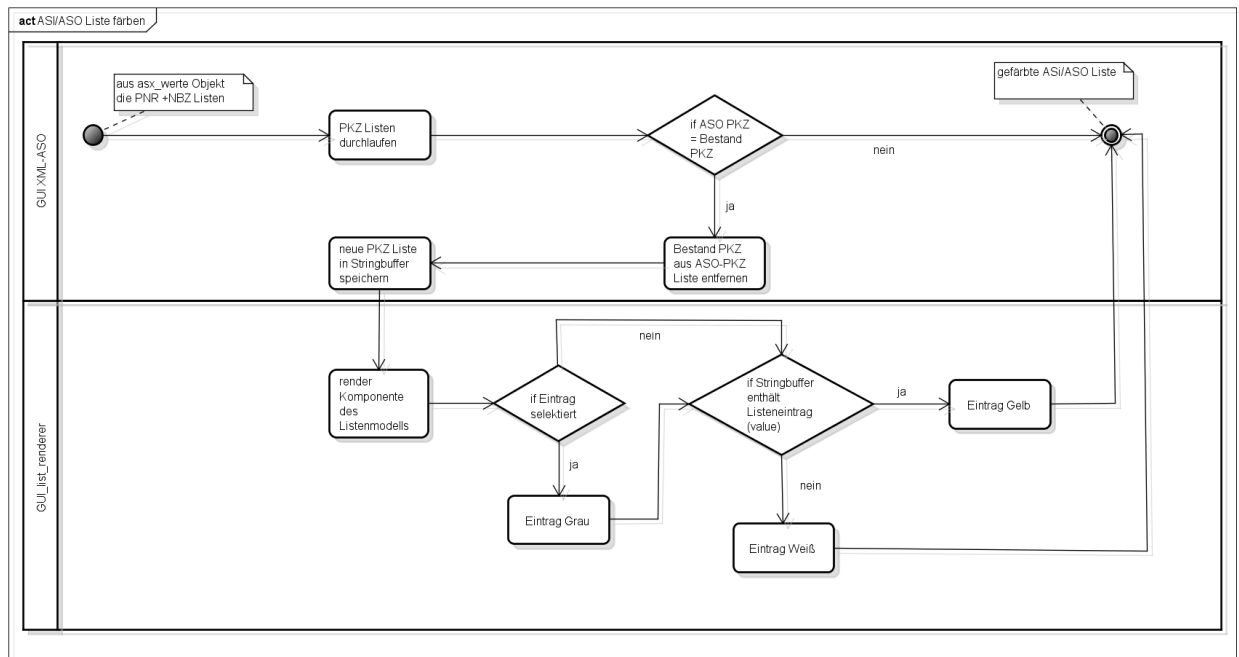


Abbildung 21: ASI/ASO Liste färben (Quelle: eigene Darstellung)

Beim Einfärben der veränderten Vermessungspunkte muss der Inhalt der beiden *Objekt_asx_werte* Objekte verglichen werden. Die Methode *asx_list_werte_vergleich* in der GUI XML-ASO durchläuft beide JList Modelle und vergleicht über die Indices der JListen und Indices der Listen im Objekt *Objekt_asx_werte* die Fachinformationen der Vermessungspunkte. Sollte der Inhalt nicht gleich sein, wird die PKZ des Index in einen String Buffer gespeichert und dieser von der zweiten Methode im Modul *GUI_list_renderer* verarbeitet. Ist im String Buffer der Wert des Objektes *value* enthalten, soll der Eintrag in der *list_utm_bestand* Liste blau gefärbt sein, andernfalls weiß.

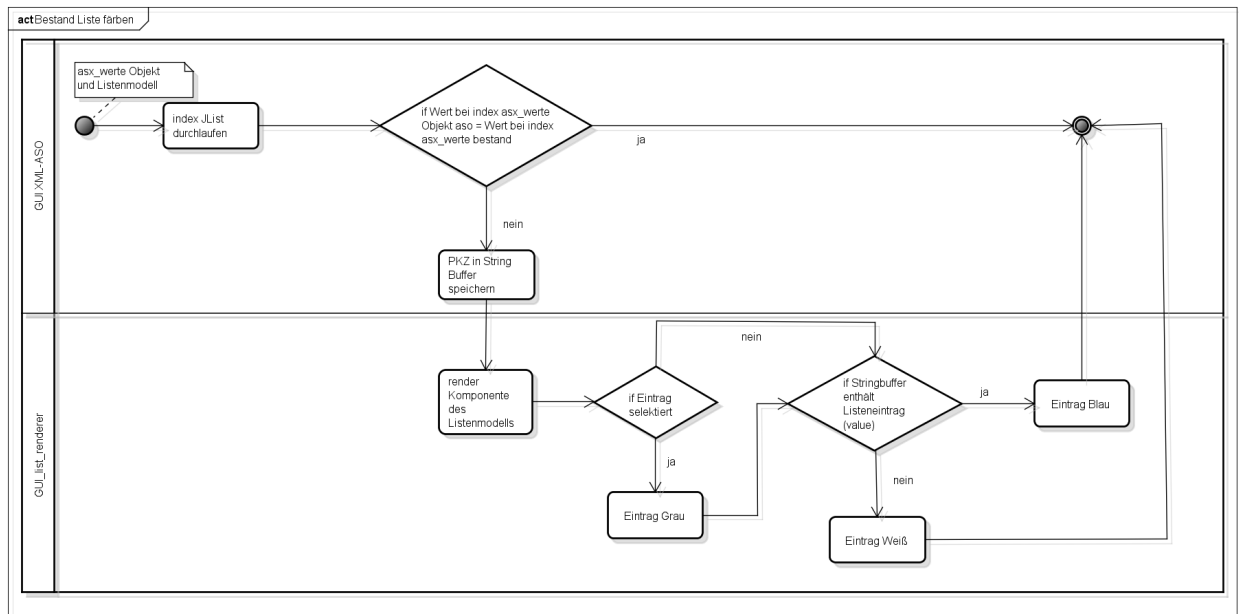


Abbildung 22: Bestandsliste färben (Quelle: eigen Darstellung)

Die Einträge in den beiden Listen sind in der Reihenfolge des Einlesens dargestellt. Für eine bessere Übersicht ist ein numerisches Sortieren nötig. Da aber der Vergleich zwischen JList Modellen und den *Objekt_asx_werte* Objekten über deren Indices realisiert wird, kann nicht einfach nur der Inhalt der JListen sortiert, sondern es müssen zwei zusätzliche Listenmodelle definiert werden. Dafür wird die Komponente GUI XML-ASO um das Modul *SortedListModel* erweitert. Dieses Modul ist aus dem JAVA development Forum [http://java.sun.com/developer/technicalArticles/J2SE/Desktop/sorted_jlist/, Autor: John O`Conner] übernommen worden. Es beinhaltet unter anderen eine Methode, um den Inhalt eines Default Listenmodells zu sortieren, ohne dessen Indices zu verändern.

Bei der Selektion eines Eintrages in den JListen kommt es zur Anzeige der Fachinformationen in einer Tabelle sowie dem Selektieren der jeweils anderen JList. Diese JTable ist, wie die JListen ein Modul, von einem Modell. Der Aufbau des Tabellenmodells ist an das Listenmodell angelehnt, aber mit der Erweiterung einzelne Zellen in der JTable anzusprechen. Das Tabellenmodell verwendet die Parameter Titelarray und Datenarray für die Darstellung. Mit Hilfe der JAVA Methode *addListSelectionListener* kann der ausgewählte Index der JList auf den Index der Listen im *Objekt_asx_werte* Objekte überführt werden. Somit wird das Datenarray des Tabellenmodells mit den passenden Fachinformationen gefüllt und angezeigt. Zu beachten bei den Indices der JListen ist, dass das unsortierten Listenmodell verwendet wird.

Die JTable stellt den Vergleich der einzelnen Fachinformationen dar. Bei Veränderungen werden diese farblich in rot angezeigt. Durch den modularen Aufbau des Tabellenmodells erfolgt die farbliche Markierung durch die Erweiterung der GUI ASO-XML mit einem Modul

Namens *GUI_table_renderer*. In diesem Modul sind zwei Methoden enthalten. Die Erste vergleicht die Einträge in den Spalten und Zeilen der JTable und setzt die Farbe rot. Die zweite Methode greift auf die Tabellenkomponente (JTable), den Inhalt im Datenarray (value), auf die Selektion, den Fokus und auf die Zeilen (row) als auch auf die Spalten (column) zu. Zu Beginn setzt die Methode die Farbe grün und durch den Aufruf der ersten Methode kommt es zum Vergleich der Zellen in der Tabelle und somit zur roten Markierung. Ein Neupunkt besitzt beim Vergleich kein Gegenpunkt in der Bestandsdatenliste. Dadurch würde die gesamte Tabelle in roter Farbe erscheinen, was ein Nutzer als unangenehm empfindet. Um dies zu verhindern, werden in der ersten Methode des *GUI_table_renderer* die Zellen auf einen leeren String geprüft. Sollte das eintreten, setzt die Methode die Farbe auf weiß.

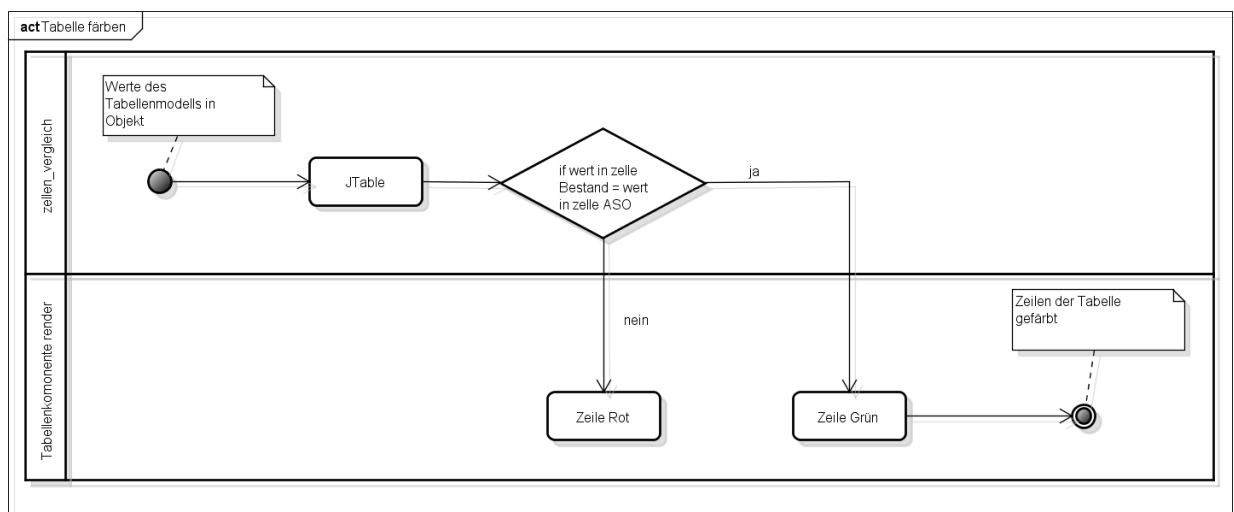


Abbildung 23: Tabelle färben (Quelle: eigene Darstellung)

Das Bedienelement *prot* gibt eine .CSV Datei aus, mit den veränderten Vermessungspunkten als Inhalt. Über den JAVA JFileChooser wird ein Dialog geöffnet, an welchen Ort die CSV Datei gespeichert werden soll. Anschließend kommt es zum Aufruf der Methode *saveProt*. Die Methode schreibt die Fachinformationen der veränderten Vermessungspunkte, welche mit Hilfe der *asx_liste_werte_vergleich* Methode ermittelt werden, in einer bestimmten, mit Silikonon getrennten Reihenfolge.

Die GUI enthält das Bedienelement *kopieren* um alle Fachinformationen der Vermessungspunkte in den Bestandsdaten auf die Fortführungsdaten zu übertragen. Dieses ActionEvent ruft die Methode *daten_kopieren* auf. Bei der Methode kommt es, wie bei der Methode *asx_list_werte_vergleich*, zu einem durchlaufen der JListen über den Index. Wenn der Index in beiden Listen gleich ist, wird der Wert aus *list_werte_aso* (Objekt vom Typ Objekt_asx_werte) entfernt und der Wert aus *list_werte_xml* dafür eingefügt. Das Bedienelement *einzel_kopieren* überträgt durch die Methode *einzelne_punkte_kopieren* den Wert eines in den JListen

selektierten Vermessungspunkt auf die Fortführungsdaten. In der Methode wird der Inhalt von *list_werte_aso* über den selektierten Index entfernt und der Inhalt von *list_werte_bestand* dafür eingefügt. Um einzelne Werte, wie z.B. die Koordinaten zu übertragen, ist es nötig die JTablele um eine Spalte, mit einer Checkbox, zu erweitern. Je nach Auswahl wird von der Methode *einzelne_punkte_kopieren* nur der Inhalt von *list_werte_aso* übertragen, der den Booleanwert „true“ besitzt in der Checkbox.

Mit dem Bedienelement *speichern* wird wieder über einen *JFileChooser* der Speicherort gewählt und die Methode *save_file* schreibt den Inhalt des *list_wert_aso* Objektes in eine ASO-Datei.

Mit dem letzten Bedienelement, *loschen* ist das Zurücksetzen aller Objekte, Listen und Arrays möglich. Dadurch muss das Programm beim Einlesen von neuen Dateien nicht neu gestartet werden, was zu einer freundlicheren Bedienung führt.

Das Textfeld in der GUI XML-ASO dient dazu, dem Benutzer Informationen über die geladenen Dateien zu liefern. Es beinhaltet zudem ein Laufzeitprotokoll über Veränderungen der Vermessungspunkte.

Nach den Tests mit ASI/ASO- und XML-Datei ergab sich die Wichtigkeit einer Konsistenzprüfung der verglichenen Fachinformationen. In der Klasse *Konsistenz_check* werden die Fachinformationen, welche mit Steuerdateien verarbeitet werden, durchlaufen. Die eingetragenen Werte im Objekt *Objekt_asx_werte* werden mit dem Inhalt in den Steuerdateien gegengeprüft und es ist kein speichern als ASO-Datei möglich bevor alle Werte korrekt sind.

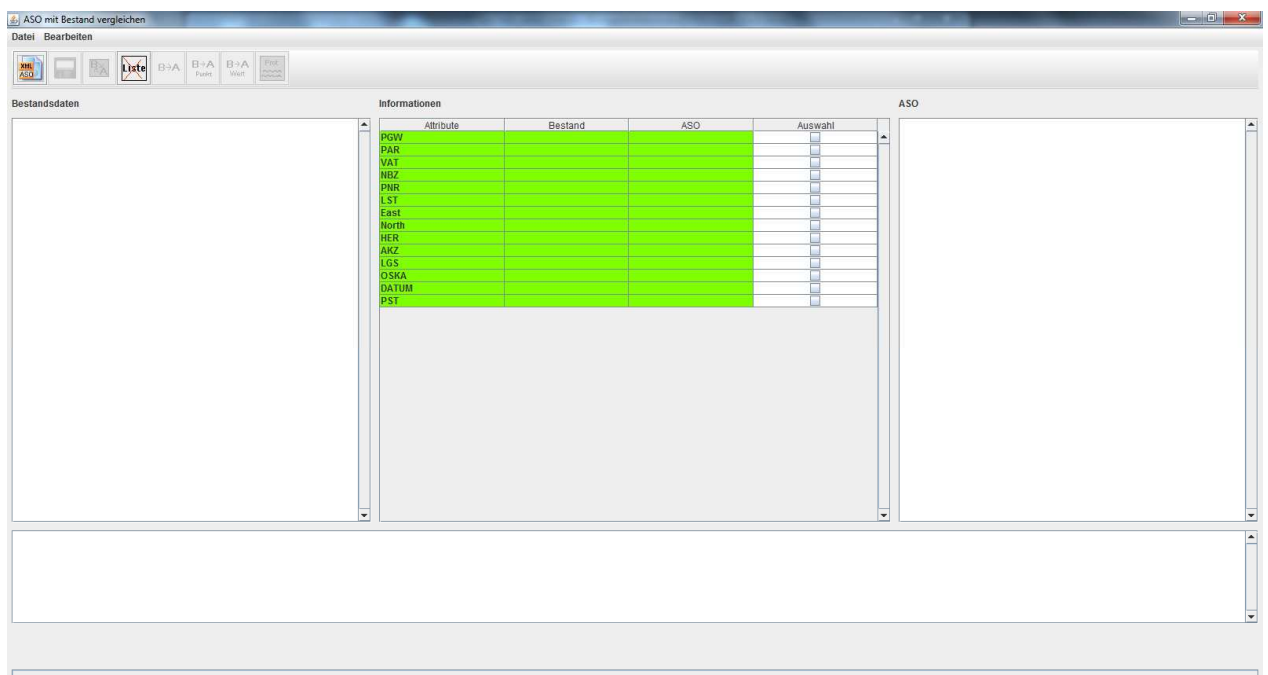


Abbildung 24: GUI XML-ASO (Quelle: eigene Darstellung)

7.4 Phase der Integration

Die Integration ist das Zusammenfügen der einzelnen Komponenten zu einem System. Anschließend folgen erste Tests des gesamten Systems und erste Leistungsmessungen. Das Ergebnis ist ein ausführbares Programmsystem.

7.4.1 Verbinden der Komponenten

Die Verbindung der einzelnen Komponenten erfolgt über die Klasse *GUI_menu*. In der Hauptmethode des Programmes wird das Menü aufgerufen. Über zwei Buttons im Menü ist es möglich die Konstruktoren der zwei Unterprogramme auszuführen. Anschließend folgt das Verarbeiten der Dateien über die Bedienelemente.

7.4.2 Test des Gesamtsystem

Da die einzelnen Komponenten im Vorfeld auf ihre Lauffähigkeit getestet wurden, kam es zu keinen Entwurfsfehlern in den Komponenten. Bedienungsfehler sind aufgetreten, als sämtliche Bedienelemente beim Start des Programms zur Verfügung standen. Durch das Ausblenden und Einblenden der Elemente, sobald sie verwendet werden können, wurde dieser Fehler behoben. Ein anderer Mangel war bei der Bedienerfreundlichkeit der GUI aufgetreten. Beim Lade- und Speicherdialog des *JFileChooser* wurde das Verzeichnis „eigene Dateien“ angezeigt. Dies ist beim Laden und anschließenden Speichern von mehreren Dateien nutzerunfreundlich. Durch das Einfügen einer Variable, welche sich den letzten Ort des öffnen bzw. speichern merkt, wurde diese Aufgabe gelöst. Ein anderer Mangel zur Bedienerfreundlichkeit waren die Anzeigen in der Toolbar. Es wurden Bilder eingefügt, welche die Textbeschreibungen ersetzt.

Die Leistungsmessung erfolgte im Rahmen des Einlesens von XML-Dateien. Ein Test hat ergeben, dass für eine 20 MB große XML-Datei ca. 3 Sekunden benötigt werden. Damit entspricht dies den Anforderungen.

7.5 Phase der Installation

Bei der Installation wird beschrieben, wie das Programm in den Vermessungsstellen aufgespielt wird und die Bedienung erklärt. Das Ergebnis ist ein lauffähiges Programmsystem.

Mit Hilfe der Entwicklerumgebung „Eclipse Helios“ können die JAVA Pakete des Programms als JAR-Datei exportiert werden. Um das Programm ausführen zu können, ist neben der JAR-Datei der Ordner *KonfigDateien* nötig. Dieser enthält alle Steuerdateien als DAT-Format. Außerdem ist der Ordner *icons* nötig, in dem sämtliche Bilder für die Toolbars enthalten sind.

Die Bedienung des Programms ist intuitiv konzipiert. Grundsätzlich ist immer folgende Reihenfolge einzuhalten: Datei Laden, Datei Vergleichen, Datei Speichern. Das Programm startet mit einem Menü zum Auswählen der Unterprogramme. Die genaue Bedienung findet sich in Anlage eins.

7.6 Phase der Wartung

Im Abschnitt Wartung wird das Programm angepasst und Veränderungen durchgeführt. Die Wartung wird bis zu einem gewissen Zeitpunkt vom Entwickler übernommen und später an das Dezernat 35 (IT-Entwicklung) des TLVermGeo übertragen.

Die Entwicklung der ALKIS-Komponenten ist planmäßig erst 2016 abgeschlossen. Bis zu diesem Zeitpunkt kommt es sowohl zu Änderungen in der ASI/ASO Formatbeschreibung als auch in den XML-Dateien. Nach Auslieferung der Betaversion wurde für beide Dateiformate ein neues Attribut mit dem Namen Zuverlässigkeit eingeführt. Dieses Attribut konnte einfach in den Quellcode integriert werden und wird mit dem Programmstand vom 17.08.2012 verarbeitet.

8. Ergebnis

Das Ergebnis dieser Arbeit ist eine Umsetzung des Entwurfes als lauffähiges Programm. Das Programm befindet sich in der Betaversion und wird im September 2012 beim Parallelbetrieb von ALK und ALKIS von den Katasterbereichen des Freistaates Thüringen weiter getestet. Die Betaversion ist in der Lage Fachinformationen der Bestandsdaten im XML Format auszulesen und diese in das ASI/ASO Format zu konvertieren. Bei dem XML Format handelt es sich um ein von der Adv festgelegte Struktur, die vom Inhalt her vom TLVermGeo erweitert werden kann. Die ASI/ASO-Datei ist ein thüringenspezifisches ASCII Format zur Darstellung von Vermessungspunkten. Die Konvertierung der Dateien ist über eine GUI übersichtlich und intuitiv zu bedienen. Außerdem kann mit Hilfe des Programms ein Vergleich zwischen den Bestandsdaten im XML Format und den Fortführungsdaten im ASO Format durchgeführt werden. Der Vergleich beider Daten ist in einer GUI realisiert und beinhaltet zusätzlich die Manipulation der Fortführungsdaten.

Das Programm ist so entworfen worden, dass ein Anpassen der Umwandlungsregeln über Steuerdateien gegeben ist. Veränderungen in den beiden Formatbeschreibungen XML und ASI/ASO, wie z.B. das Hinzufügen zusätzlicher Attribute kann durch die Architektur des Programms einfach integriert werden.

9. Fazit

Der Freistaat Thüringen führt eine Umstellung von der ALK und dem ALB nach ALKIS durch. In beiden Systemen sind unterschiedliche Datenformate für die Datenerhebung zur Beschreibung eines Vermessungspunktes vorhanden. In ALKIS erfolgt die Datenerhebung mit dem von der AdV festgelegten XML Format. In diesem kann der Inhalt vom TLVermGeo erweitert werden. Das alte System führt die Datenerhebung im thüringenspezifische ASI/ASO Format. Dieses Format beschreibt einen Vermessungspunkt in ASCII. ALKIS unterstützt das ASI/ASO Format nicht. Aber es ist für die Übergangszeit der Umstellung nötig, den Vermessungsstellen dies zur Verfügung zu stellen. Der Grund hierfür ist, dass die Vermessungsstellen noch nicht in der Lage sind Bestandsdaten und Fortführungsdaten im XML Format zu verarbeiten. Zum Lösen dieser Problematik ist ein Konverter zwischen beiden Formaten erforderlich. Der Konverter wandelt die Fachinformationen des XML Formates in das ASI/ASO Format um. Zu beachten ist dabei die unterschiedliche Struktur der beiden Formate und somit eine unterschiedliche Kodierung der Fachinformationen. In den XML-Dateien gibt es z.B. keinen OSK mehr, aber dieser ist für die Messung im Außendienst Entscheidend. Über die VAT und der PAT aus der XML-Datei wird der OSK in die ASI/ASO-Datei überführt. Da sich die Anforderungen des Freistaates Thüringen an ALKIS in der Entwicklung befindet, ist es nötig die Umsetzung der Fachinformationen festzulegen, ohne in den Quellcode einzugreifen. Hierfür dienen Steuerdateien, die den Schlüssel der XML-Dateien enthalten und ihnen ist ein Wert in ASI/ASO Kodierung zugeordnet. Durch die Anpassung von ALKIS und somit der Möglichkeit die XML-Dateien um Inhalte zu erweitern, ist der Konverter so entworfen worden, dass ein Einfügen der zusätzlichen Attribute in den Quellcode ohne großen Aufwand möglich ist. Ein Beispiel ist das Attribut Zuverlässigkeit, welches am 17.08.2012 in das Programm eingefügt wurde.

Dadurch, dass die Bestandsdaten im XML Format sind und die Fortführungsdaten im ASI/ASO Format nicht von ALKIS unterstützt werden, ist ein Programm zum Vergleich beider nötig. Dieses Programm ist der zweite Teil der Aufgabe. Es konvertiert die Fachinformationen der Bestandsdaten und vergleicht diese mit den Fortführungsdaten. Hierbei lag die Schwierigkeit darin, eine gut strukturierte GUI mit allen nötigen Funktionen zu entwerfen. Der Entwurf soll sowohl die Problematik des einfachen Ergänzens von zusätzlichen Attributen erfüllen, als auch übersichtlich gestaltet sein.

Abschließend ist zu erwähnen, dass sich die Betaversion des „MW-Konverter“ im September 2012 in einer weiteren Testphase befindet. Nach Abschluss des Testes müssen ca. 400 Personen in den Vermessungsstellen im Freistaat Thüringen während der Übergangszeit mit diesem Programm arbeiten. Der Grundentwurf ist gelegt, umgesetzt und kann den Anforderungen des Freistaates Thüringen, im Zuge der ALKIS Veränderungen, weiter angepasst werden.

Literaturverzeichnis

Kummer und Frankenberger: Das deutsche Vermessungs- und Geoinformationswesen – ALKIS: Das Amtliche Liegenschaftskataster-Informationssystem; Verlag: Wichmann, Auflage von 2012

Christian Ullerbloom: Java ist auch eine Insel; Verlag: Galileo Computing, 9. Auflage

Arbeitsgemeinschaft der der Vermessungsverwaltung der Länder der Bundesrepublik Deutschland: Dokumentation zur Modellierung der Geoinformationen des amtlichen Vermessungswesen (GeoInfoDok) – Hauptdokument; Version 6.1 vom 01.07.2009

Arbeitsgemeinschaft der der Vermessungsverwaltung der Länder der Bundesrepublik Deutschland: Dokumentation zur Modellierung der Geoinformationen des amtlichen Vermessungswesen (GeoInfoDok) – ALKIS-Objektartenkatalog, Version 6.0 vom 11.04.2008

Markus Seifert: Wissenschaftlicher Beitrag für den Aufbau einer Geodateninfrastruktur zur Lösung von Aufgaben des E-Government, ETH Zürich, Dissertation von 2008

Kreitlow Stefanie, Brettschneider Andreas, Cord-Hinrich Jahn, Feldman-Westendorf Uwe: ETRS89/UTM – Der Bezugssystemwechsel und die Auswirkungen auf Geodatennutzung; in Kartographische Nachrichten, 60. Jahrgang, Heft 4/ 2010; Verlag: Kirschbaum Verlag GmbH

Thüringer Landesamt für Vermessung und Geoinformation: Thüringer Verwaltungsvorschrift für das Liegenschaftskataster (ThürVV-Lika); Ausgabe vom 12/2011

Thüringer Landesamt für Vermessung und Geoinformation: Amtliches Liegenschaftskatasterinformationssystem (ALKIS) – Migrationstabellen; Version 1.5 vom 07.02.2012

Wehrenpfennig Andreas: Skript - Software-Engineering, Grundlagen; Hochschule Neubrandenburg

Internet Quellen:

www.thueringen.de/de/tlvermgeo; Stand: 07.09.2006; URL:

<http://www.thueringen.de/de/tlvermgeo/wir/aktuelles/themen/23701/uiindex.html>; letzter Abruf am 28.08.2012

Wandinger Marcus: <http://www.adv-online.de>; Stand: 04.07.2011; URL:

<http://www.adv-online.de/icc/extdeu/broker>; letzter Abruf am 28.08.2012

O`Conner John: <http://java.sun.com/developer>; URL:

http://java.sun.com/developer/technicalArticles/J2SE/Desktop/sorted_jlist; letzter Abruf am 28.08.2012

Mündliche Quellen

Dipl.-Ing. (FH) Matthias Franke (02.07.2012): Aktueller Stand des ASI Export; TLVermGeo Erfurt

Dipl.-Ing. (FH) Volker Rossow, Dipl.-Ing. Frank Fuchs (03.07.2012): Zukünftiger ASI Export; TLVermGeo Erfurt

Anlagen

Anlage 1 Bedienung

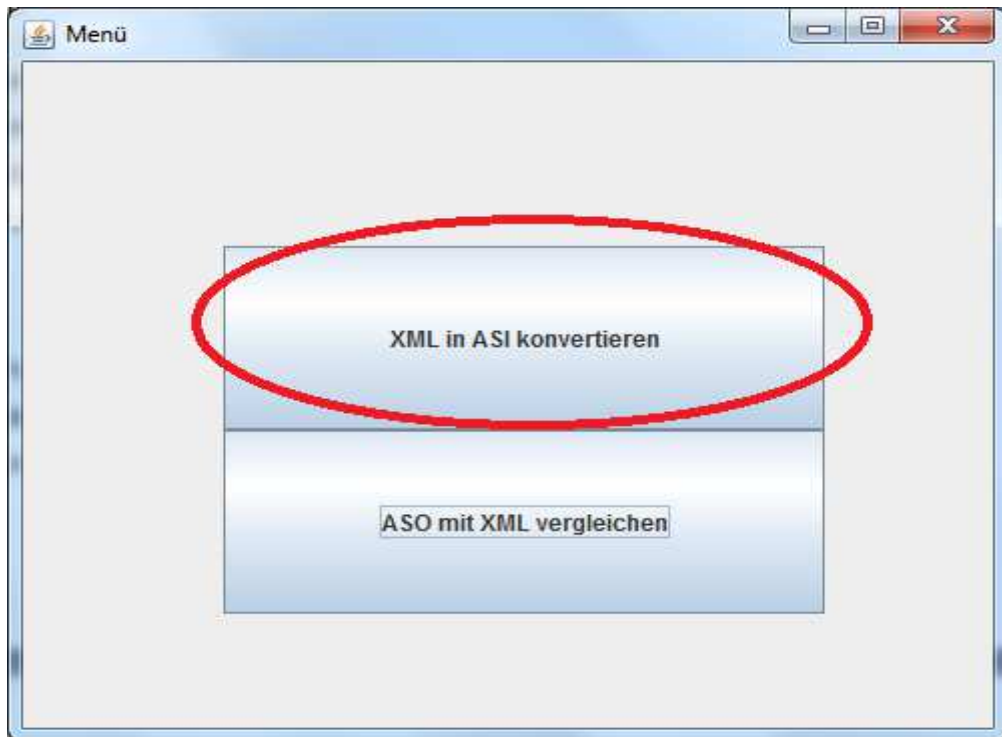


Abbildung 1: XML2ASI Unterprogramm starten

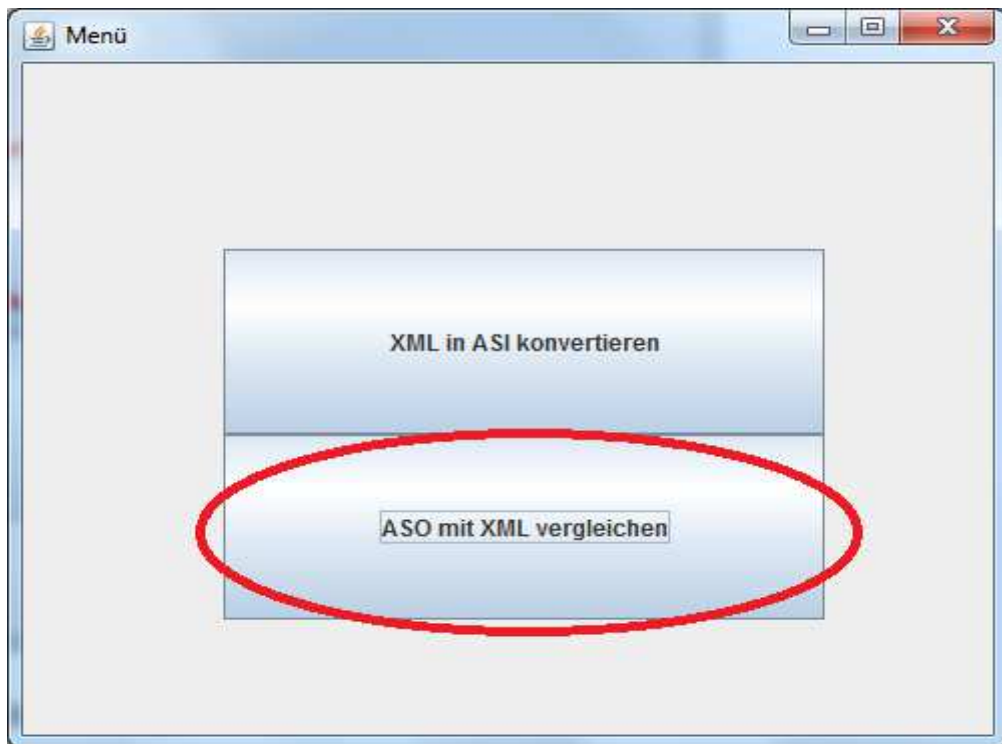


Abbildung 2: XML-ASO Unterprogramm starten

Das Unterprogramm XML2ASI hat folgenden Ablauf:

1. Eine XML-Datei Laden

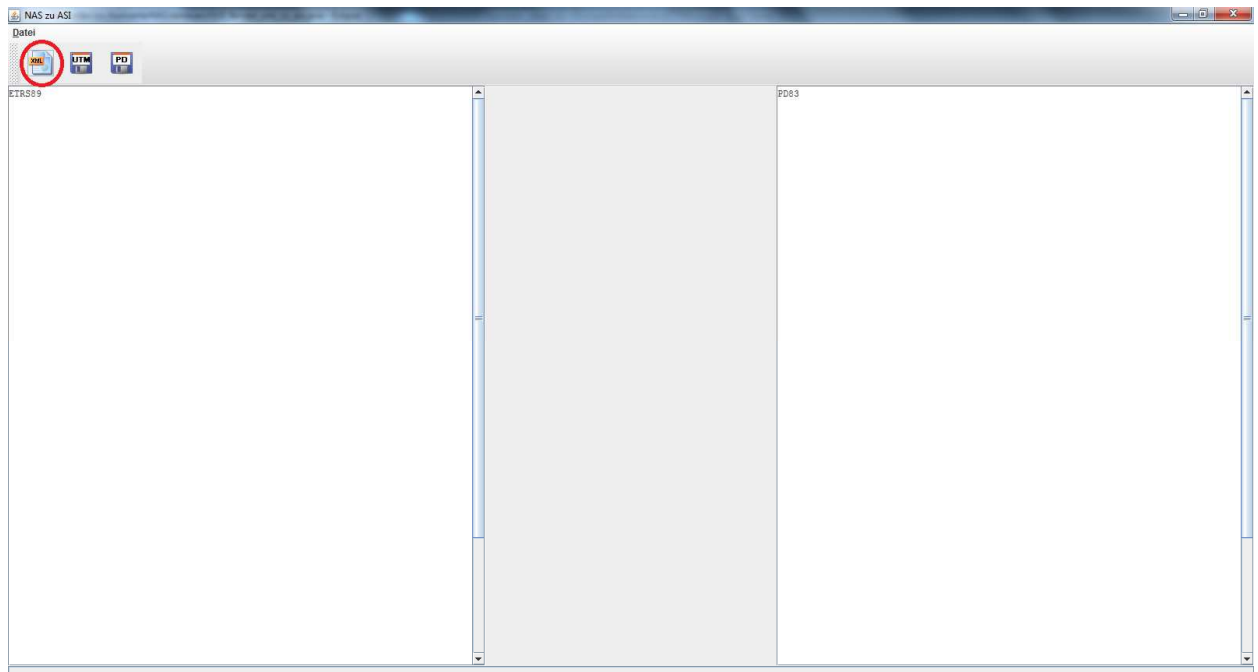


Abbildung 3: XML-Datei Laden (XML2ASI)

2. ETRS89/UTM oder PD83/GK Feld speichern als ASI-Datei

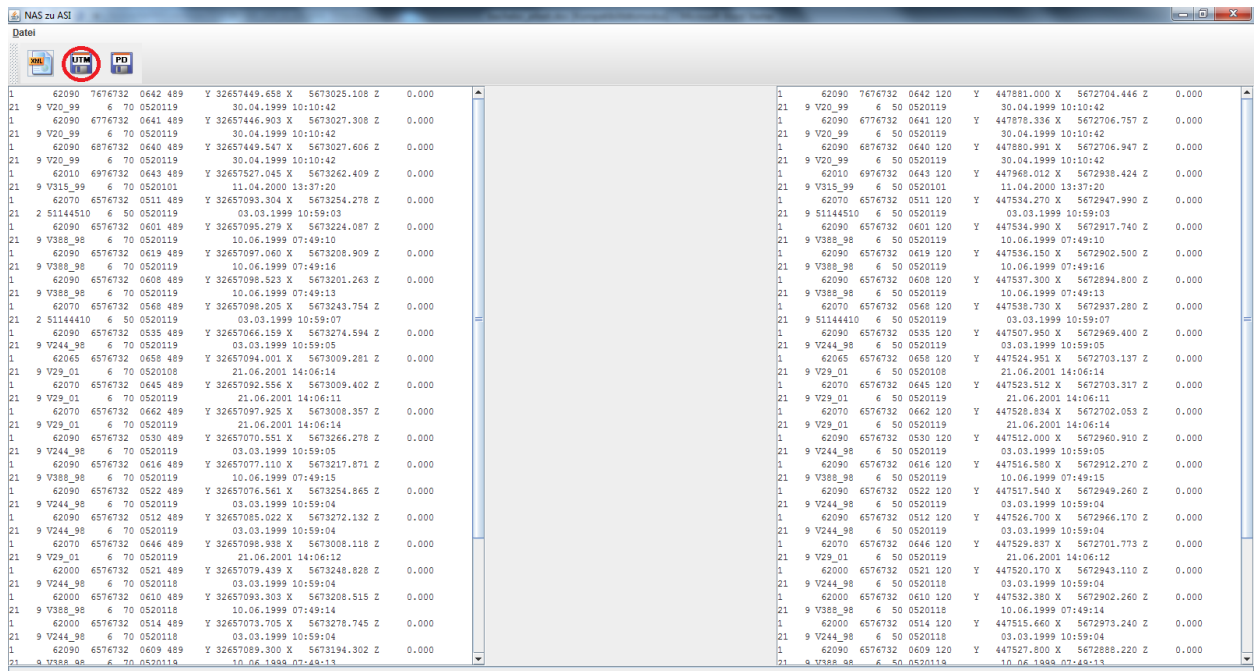


Abbildung 4: ETRS89/UTM speichern

Line	Code	Year	Month	Day	Time	Source	Target	Time	Source	Target	Time
1	62090	676732	0642	489	Y	32657449.658	X	5673025.108	Z	0.000	
21	9	V20_99	6	70	0520119			30.04.1999	10:10:42		
1	62090	6776732	0641	489	Y	32657446.903	X	5673027.308	Z	0.000	
21	9	V20_99	6	70	0520119			30.04.1999	10:10:42		
1	62090	6876732	0640	489	Y	32657449.547	X	5673027.606	Z	0.000	
21	9	V20_99	6	70	0520119			30.04.1999	10:10:42		
1	62010	6976732	0643	489	Y	32657527.045	X	5673262.409	Z	0.000	
21	9	V315_99	6	70	0520101			11.04.2000	13:37:20		
1	62070	6576732	0511	489	Y	32657095.304	X	5673254.278	Z	0.000	
2	2	S1144510	6	50	0520119			03.03.1999	10:59:03		
1	62090	6576732	0601	489	Y	32657095.279	X	5673224.087	Z	0.000	
21	9	V388_98	6	70	0520119			10.06.1999	07:49:10		
1	62090	6576732	0619	489	Y	32657097.060	X	5673208.909	Z	0.000	
21	9	V388_98	6	70	0520119			10.06.1999	07:49:16		
1	62090	6576732	0608	489	Y	32657095.523	X	5673201.263	Z	0.000	
21	9	V388_98	6	70	0520119			10.06.1999	07:49:13		
1	62070	6576732	0568	489	Y	32657096.205	X	5673243.754	Z	0.000	
2	2	S1144410	6	50	0520119			03.03.1999	10:59:07		
1	62090	6576732	0535	489	Y	32657064.159	X	5673274.594	Z	0.000	
21	9	V244_98	6	70	0520119			03.03.1999	10:59:05		
1	62065	6576732	0658	489	Y	32657094.001	X	5673009.281	Z	0.000	
21	9	V29_01	6	70	0520108			21.06.2001	14:06:14		
1	62070	6576732	0645	489	Y	32657092.556	X	5673009.402	Z	0.000	
21	9	V29_01	6	70	0520119			21.06.2001	14:06:11		
1	62070	6576732	0662	489	Y	32657097.925	X	5673008.357	Z	0.000	
21	9	V29_01	6	70	0520119			21.06.2001	14:06:14		
1	62090	6576732	0530	489	Y	32657070.551	X	5673266.278	Z	0.000	
21	9	V244_98	6	70	0520119			03.03.1999	10:59:05		
1	62090	6576732	0616	489	Y	32657077.110	X	5673217.871	Z	0.000	
21	9	V388_98	6	70	0520119			10.06.1999	07:49:15		
1	62090	6576732	0522	489	Y	32657076.561	X	5673254.865	Z	0.000	
21	9	V244_98	6	70	0520119			03.03.1999	10:59:04		
1	62090	6576732	0512	489	Y	32657085.022	X	5673272.132	Z	0.000	
21	9	V244_98	6	70	0520119			03.03.1999	10:59:04		
1	62070	6576732	0646	489	Y	32657096.938	X	5673008.118	Z	0.000	
21	9	V29_01	6	70	0520119			21.06.2001	14:06:12		
1	62000	6576732	0521	489	Y	32657079.459	X	5673248.828	Z	0.000	
21	9	V244_98	6	70	0520118			03.03.1999	10:59:04		
1	62000	6576732	0610	489	Y	32657093.303	X	5673208.515	Z	0.000	
21	9	V388_98	6	70	0520118			10.06.1999	07:49:14		
1	62000	6576732	0514	489	Y	32657073.705	X	5673278.745	Z	0.000	
21	9	V244_98	6	70	0520118			03.03.1999	10:59:04		
1	62090	6576732	0609	489	Y	32657089.300	X	5673194.302	Z	0.000	
21	9	V388_98	6	70	0520119			10.06.1999	07:49:13		

Abbildung 5: PD83/GK speichern

Beim Unterprogramm XML-ASO ist die Bedienung folgende:

- Über das Bedienelement kann sowohl die XML-Datei als auch die ASO-Datei geladen werden. Hierbei ist die Reihenfolge der der zu Ladeneden Datei nicht von Bedeutung und es können mehrere Dateien eingelesen werden.

Attribute	Bestand	ASO	Auswahl
PCW			<input type="checkbox"/>
PAR			<input type="checkbox"/>
VAT			<input type="checkbox"/>
NBZ			<input type="checkbox"/>
PNR			<input type="checkbox"/>
LST			<input type="checkbox"/>
East			<input type="checkbox"/>
North			<input type="checkbox"/>
HER			<input type="checkbox"/>
AKZ			<input type="checkbox"/>
LGS			<input type="checkbox"/>
OSKA			<input type="checkbox"/>
DATUM			<input type="checkbox"/>
PST			<input type="checkbox"/>

Abbildung 6: XML- und ASO-Datei Laden

2. Das Bedienelement zum Vergleichen auswählen

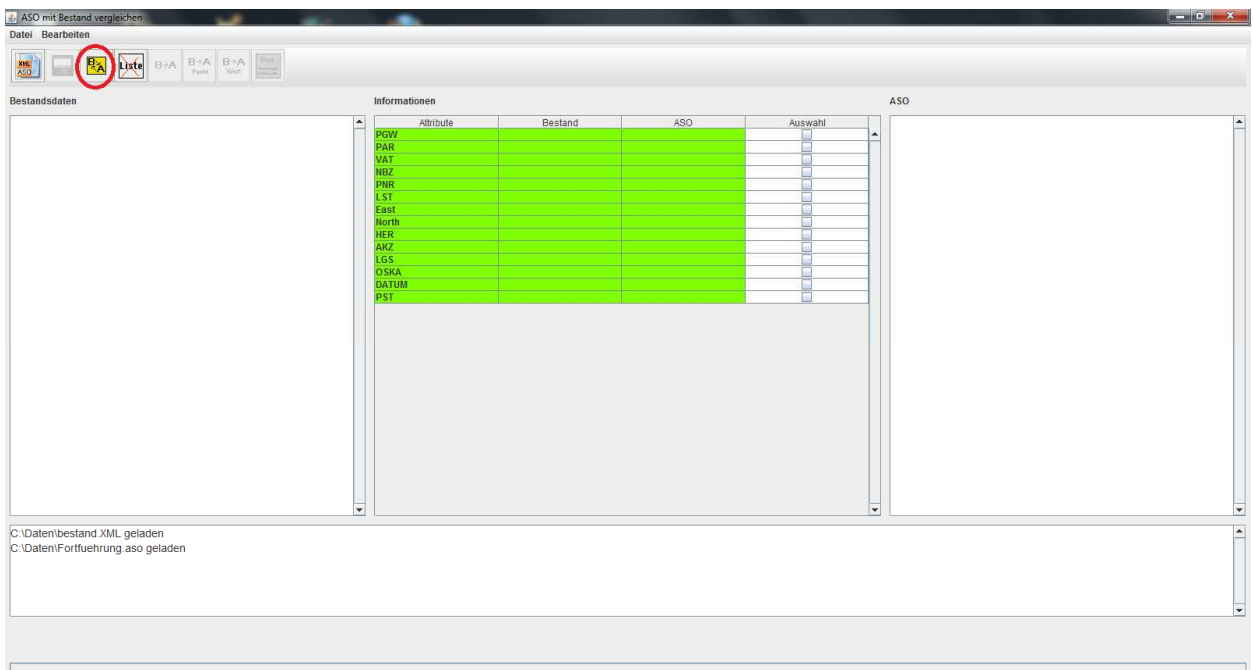


Abbildung 7: Vergleich von Bestandsdaten und Fortführungsdaten

3. Die Kontrolle und ggf. Veränderung der Daten kann behandelt werden

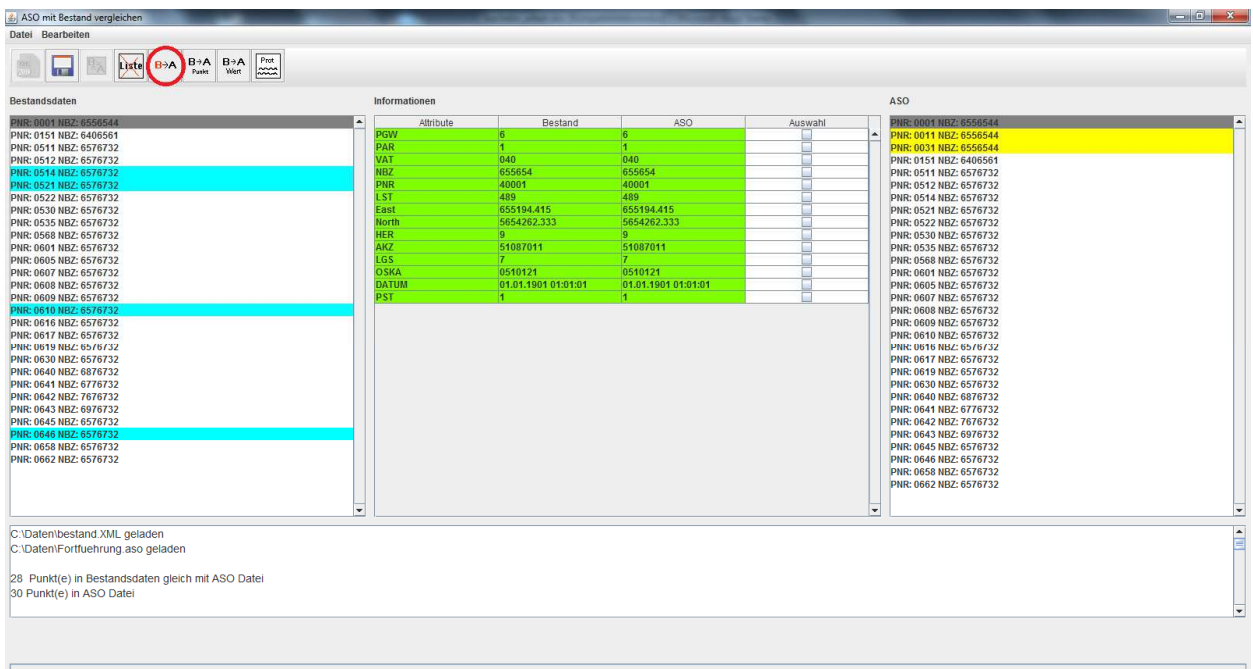


Abbildung 8: alle Informationen von Bestandsdaten zu Fortführungsdaten kopieren

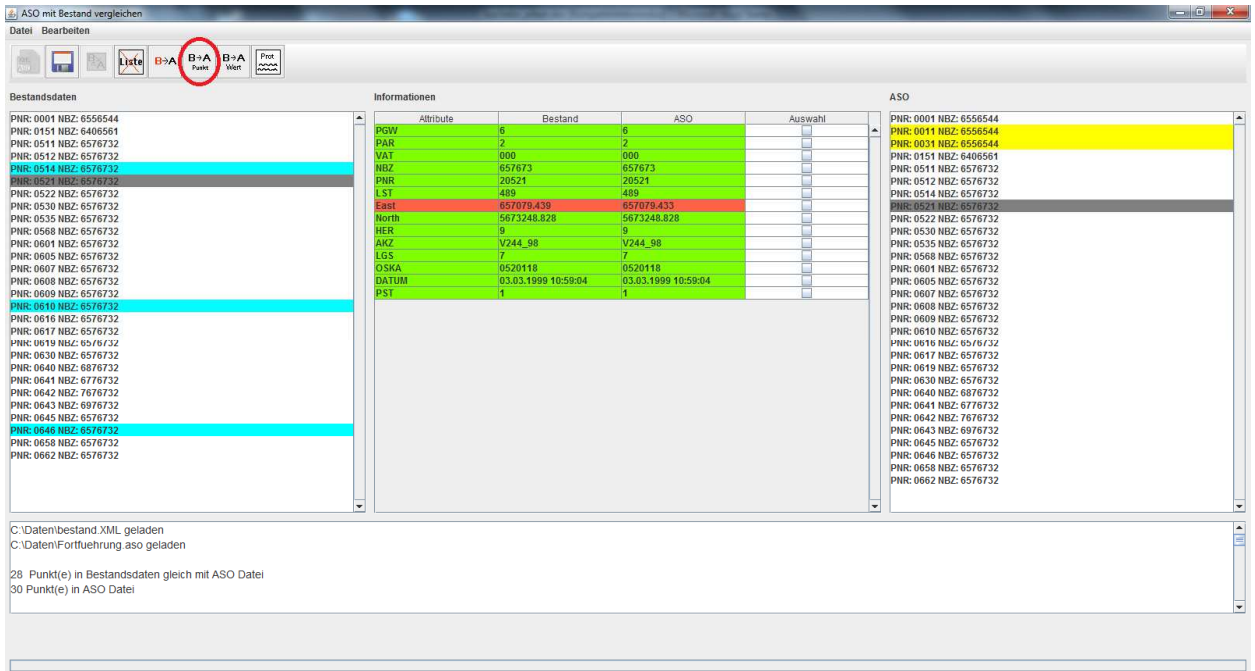


Abbildung 9: alle Informationen des Selektierten Punktes von Bestandsdaten zu Fortführung kopieren

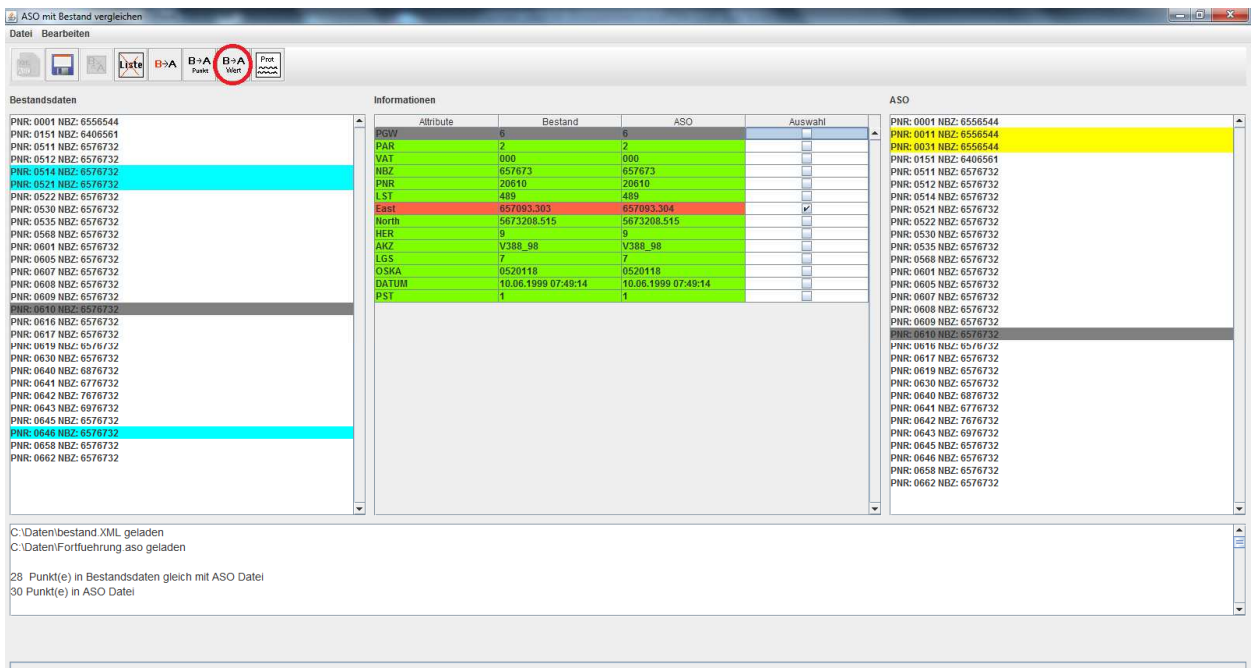


Abbildung 10: in Tabelle durch Hacken Ausgewählte Information kopieren

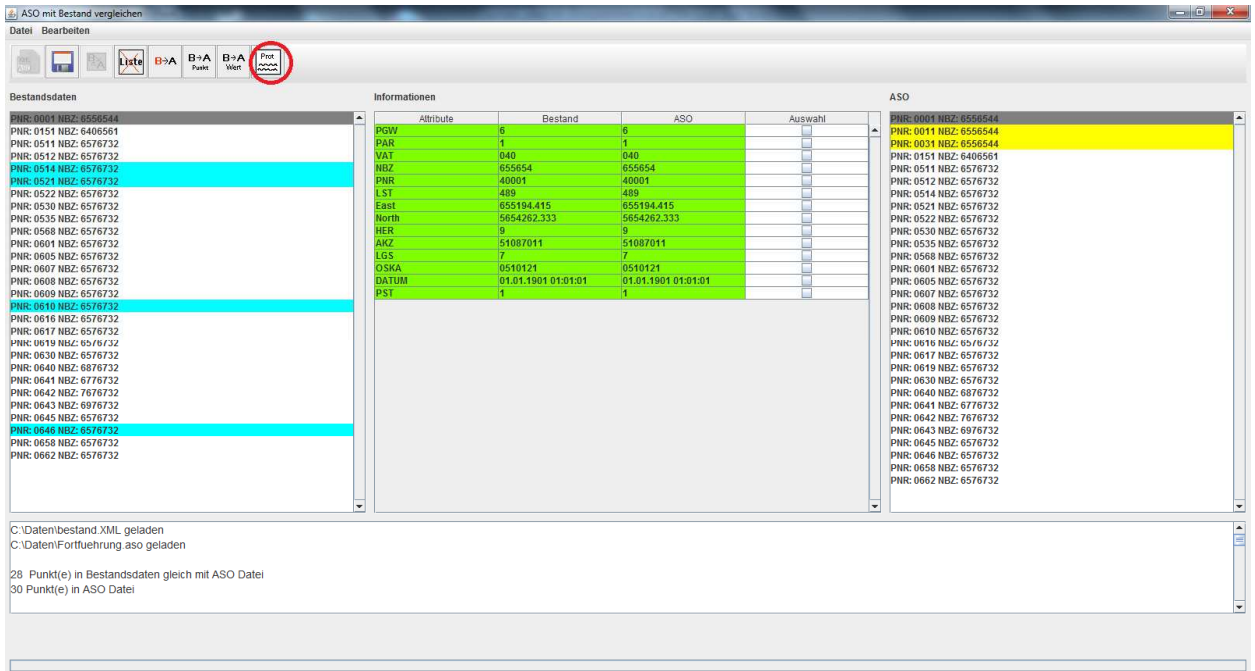


Abbildung 11: Protokoll Ausgaben als CSV-Datei

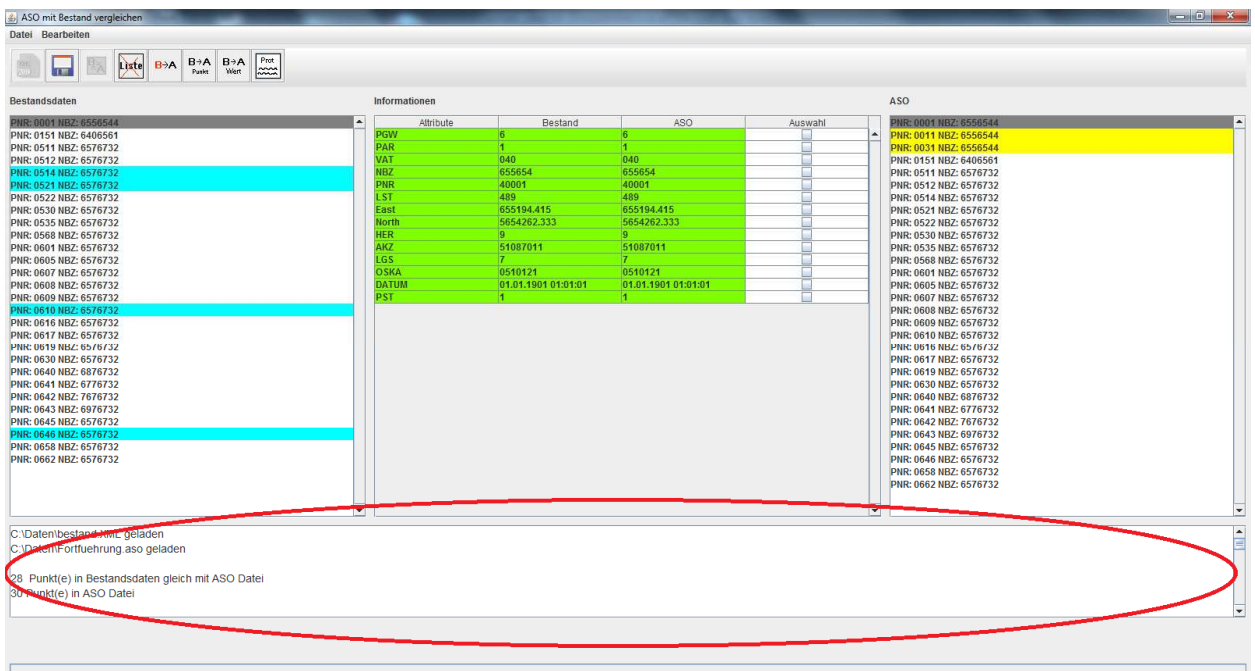


Abbildung 12: Textfeld mit Informationen zu geladenen Dateien und Laufzeitprotokoll

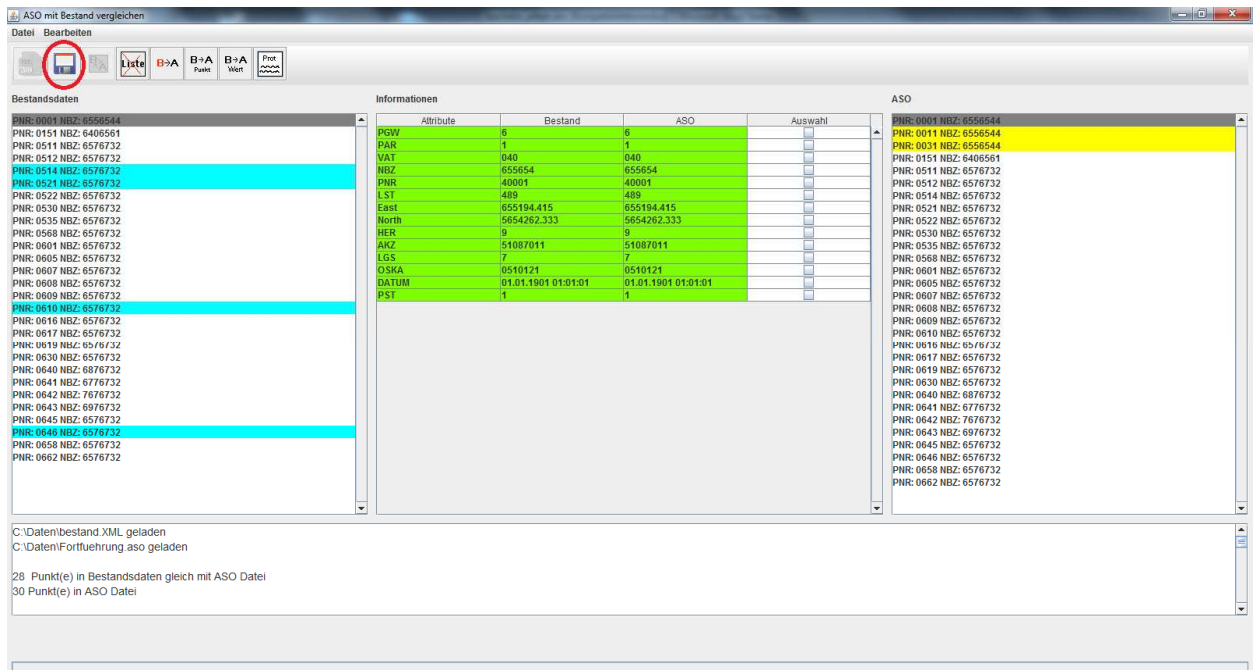


Abbildung 13: Fortführungsdaten speichern als ASO-Datei

Anlage 2 ASI/ASO in Erweiterten Backus-Naur-Form

Leerzeichen= " " ;
 Ziffern= "0" | "1" | "2" | "3" | "4" | "5" |
 "6" | "7" | "8" | "9" ;
 PGW= Ziffern{1} ;
 PAT= Ziffern{1} ;
 VAT= Ziffern{3} ;
 NBZ= Ziffern{7} ;
 PNR= Ziffern{4} ;
 LST= Ziffern{3} ;
 Rechtswert= Ziffern{12} ;
 Hochwert= Ziffern{10} ;
 Hoehe= Ziffern{8} ;
 PST= Ziffern{1} ;
 HER= Ziffern{1} ;
 AKZ= Ziffern{8} ;
 HST= Ziffern{1} ;
 LGS= Ziffern{1} ;
 OSK= Ziffern{7} ;
 Datum= Ziffern{2}, ".", Ziffern {2}, ".", Ziffern{4},
 Leerzeichen, Ziffern{2}, ":", Ziffern{2}, ":", Ziffern{2} ;
 Zeilenkennung= Ziffer{1} ;

ersteZeile= Zeilenkennung, Leerzeichen{8}, PGW, PAT, VAT,
Leerzeichen{2},
NBZ, Leerzeichen{2}, PNR, Leerzeichen, LST, Leerzeichen{6},
Rechtwert, Leerzeichen{6}, Hochwert, Leerzeichen{4}, Hoehe ;

zweiteZeile= Zeilenkennung, PST, Leerzeichen{3}, HER, Leerzeichen, AKZ,
HST, LGS, Leerzeichen{4}, OSK, Leerzeichen{10}, Datum ;

Eidesstattliche Erklärung

„Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig und nur unter Benutzung der angegebenen Literatur und Hilfsmittel angefertigt habe. Wörtlich übernommene Sätze oder Satzteile sind als Zitat belegt, andere Anlehnungen hinsichtlich Aussage und Umfang unter Quellenangabe kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen und ist nicht veröffentlicht“.

Ort, Datum

Unterschrift