H:\NERZ\NERZ-logo DokKopf.jpg

Softwareentwurf

Segment DUA, SWE Messwertersetzung LVE

|  |  |
| --- | --- |
| Version | 5.0 |
| Stand | 23.05.2016 |
| Produktzustand | Akzeptiert |
| Datei | SwEnt\_DUA-MWELVE\_FREI\_V5.0\_D2016-05-23.docx |
|  |  |
| Projektkoordinator | Herr Nelson |
| Projektleiter | Herr Bräuner |
| Projektträger | Regierungspräsidium Tübingen Landesstelle für Straßentechnik  Heilbronner Straße 300 - 302  70469 Stuttgart |
| Ansprechpartner | Herr Bräuner |

# Allgemeines

## Lizenz

by-sa

Dieses Dokument steht unter der Creative-Commons-Lizenz Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International. Um eine Kopie dieser Lizenz zu sehen, besuchen Sie http://creativecommons.org/licenses/by-sa/4.0/.

## Kurzbeschreibung

Das Dokument beschreibt den Feinentwurf der Softwareeinheit. Der Feinentwurf schließt die Lücke zwischen dem Grobentwurf der Softwarearchitektur und der Implementierung. Hierbei werden die notwendigen Klassen identifiziert und entworfen, zu verwendende Entwurfsmuster identifiziert, die Klassen und Schnittstellen beschrieben sowie die nichtfunktionalen Anforderungen berücksichtigt.

## Verteiler

| **Organisationseinheit** | **Name** | **Anzahl Kopien** | **Vermerk** |
| --- | --- | --- | --- |
| RPT LST | Herr Bräuner |  |  |
| Kappich Systemberatung | Herr Kappich |  |  |
|  |  |  |  |
|  |  |  |  |

## Änderungsübersicht

| **Version** | **Datum** | **Kapitel** | **Bemerkungen** | **Bearbeiter** |
| --- | --- | --- | --- | --- |
| 1.0 | 17.11.2006 |  | Erstellung erster Entwurf | Th. Thierfelder |
| 2.0 | 30.11.2006 |  | Änderungen des Prüfprotokolls V1.0 eingearbeitet | Th. Thierfelder |
| 3.0 | 06.12.2006 |  | Überführung in den Zustand "Akzeptiert" | T. Bräuner |
| 4.0 | 30.05.2012 |  | Anpassung Namenskonvention NERZ | T. Bräuner |
| 4.1 | 20.10.2015 |  | Anpassung Deckblatt Ergänzung Kapitel 0.1 Kurzbeschreibung | T. Bräuner |
| 4.2 | 05.01.2016 |  | Anpassung an neue AFo | J. Haas, G. Kappich |
| 4.3 | 12.01.2016 |  | Anpassung aufgrund des Prüfprotokolls vom 12.01.2016 | J. Haas, G. Kappich |
| 5.0 | 15.01.2016 |  | Schlussredaktion, Überführung in den Zustand „Akzeptiert“ , Lizenz | T. Bräuner |

## Inhaltsverzeichnis

[0 Allgemeines 2](#_Toc451779810)

[0.1 Lizenz 2](#_Toc451779811)

[0.2 Kurzbeschreibung 2](#_Toc451779812)

[0.3 Verteiler 2](#_Toc451779813)

[0.4 Änderungsübersicht 2](#_Toc451779814)

[0.5 Inhaltsverzeichnis 3](#_Toc451779815)

[0.6 Abkürzungsverzeichnis 5](#_Toc451779816)

[0.7 Definitionen 5](#_Toc451779817)

[0.8 Referenzierte Dokumente 5](#_Toc451779818)

[0.9 Abbildungsverzeichnis 5](#_Toc451779819)

[0.10 Tabellenverzeichnis 5](#_Toc451779820)

[1 SW-Einheit Messwertersetzung LVE 6](#_Toc451779821)

[1.1 Beschreibung der SW-Einheit 6](#_Toc451779822)

[1.2 Aufbau der Softwareeinheit 6](#_Toc451779823)

[1.3 Schnittstellen zur Umgebung 6](#_Toc451779824)

[1.4 Realisierung der SW-Einheit 6](#_Toc451779825)

[1.5 Lokale Daten 7](#_Toc451779826)

[1.6 Ausnahmeverhalten 7](#_Toc451779827)

[2 Modul Verwaltung 8](#_Toc451779828)

[2.1 Beschreibung des Moduls 8](#_Toc451779829)

[2.2 Aufbau des Moduls 8](#_Toc451779830)

[2.3 Schnittstellen zur Umgebung 8](#_Toc451779831)

[2.3.1 Allgemeine Verwaltungsschnittstelle innerhalb der DUA 8](#_Toc451779832)

[2.3.2 Schnittstelle zum Starter 8](#_Toc451779833)

[2.3.3 Schnittstelle zum Datenverteiler 8](#_Toc451779834)

[2.3.4 Schnittstelle VerwaltungPL-Prüfung formal 9](#_Toc451779835)

[2.4 Realisierung des Moduls 9](#_Toc451779836)

[2.4.1 Initialisierung 9](#_Toc451779837)

[2.4.2 Initialisierung des Moduls PL-Prüfung formal 9](#_Toc451779838)

[2.4.3 Initialisierung des Moduls PL-Prüfung logisch LVE 9](#_Toc451779839)

[2.4.4 Initialisierung des Moduls Messwertersetzung LVE 10](#_Toc451779840)

[2.4.5 Initialisierung des Moduls Publikation 10](#_Toc451779841)

[2.4.6 Versorgung der Module PL-Prüfung formal, PL-Prüfung logisch LVE, Messwertersetzung LVE und Publikation mit zu plausibilisierenden Daten 10](#_Toc451779842)

[2.5 Lokale Daten 10](#_Toc451779843)

[2.6 Ausnahmeverhalten 10](#_Toc451779844)

[3 Modul Messwertersetzung LVE 11](#_Toc451779845)

[3.1 Beschreibung des Moduls 11](#_Toc451779846)

[3.2 Aufbau des Moduls 11](#_Toc451779847)

[3.3 Schnittstellen zur Umgebung 11](#_Toc451779848)

[3.3.1 Allgemeine Schnittstellenbeschreibung BearbeitungsKnoten 11](#_Toc451779849)

[3.3.2 Schnittstelle zum Datenverteiler 11](#_Toc451779850)

[3.3.3 Schnittstelle VerwaltungMesswertersetzung LVE 11](#_Toc451779851)

[3.3.4 Schnittstelle zur Datenaktualisierung 12](#_Toc451779852)

[3.4 Realisierung des Moduls 12](#_Toc451779853)

[3.5 Lokale Daten 12](#_Toc451779854)

[3.6 Ausnahmeverhalten 13](#_Toc451779855)

## Abkürzungsverzeichnis

Die relevanten Abkürzungen sind in einem separaten Dokument zusammengefasst. Zusätzliche Abkürzungen werden in dem vorliegenden Dokument nicht verwendet.

## Definitionen

Es werden keine Definitionen benötigt.

## Referenzierte Dokumente

|  |  |
| --- | --- |
| AFo | Anwenderforderungen SE-02.00.00.00.00-AFo-6.0 |
| BibAPI | JavaDoc zu Bibliotheks-API aus kernsoftware-system-3.6.2 |
| SwEnt\_PlPrüfformal | SW-Entwurf, Segment DUA, SWE PL-Prüfung formal |
| SwEnt\_PlPrüflogLVE | SW-Entwurf, Segment DUA, SWE PL-Prüfung logisch LVE |
| SwEnt\_PlPrüflogUFD | SW-Entwurf, Segment DUA, SWE PL-Prüfung logisch UFD |
| DatK | Datenkatalog, SE-02.00.00.00.00-DatK |

## Abbildungsverzeichnis

Abbildung 1.1: Zerlegung der SWE *Messwertersetzung LVE* 6

## Tabellenverzeichnis

Tabelle 2‑1: Daten, für die das Modul *Verwaltung* angemeldet ist 8

Tabelle 3‑1: Vom Modul *Messwertersetzung LVE* interpretierte Parameter 11

Tabelle 3‑2: Vom Modul *Messwertersetzung LVE* durchgeführte Quellenanmeldung 11

# SW-Einheit Messwertersetzung LVE

## Beschreibung der SW-Einheit

Aufgabe der SWE *Messwertersetzung* *LVE* ist es*,* alle empfangenen Kurzzeitdatensätze einer formalen und logischen Plausibilisierung zuzuführen. Die aus diesen Plausibilisierungsstufen als *Implausibel* hervorgegangenen Attributwerte werden dann einer Messwertersetzung unterzogen. Eine genaue Beschreibung der Ersetzungsmethoden erfolgt in [AFo]. Nach dieser Prüfung bzw. Ersetzung werden die Daten unter dem Aspekt asp.messWertErsetzung publiziert.

## Aufbau der Softwareeinheit

Die SWE *Messwertersetzung* *LVE* ist ein Softwaresystem, das eigenständig gestartet und beendet werden kann. Sie besteht aus fünf Modulen (siehe Abbildung 1.1): Modul *Verwaltung*, Modul *PL-Prüfung formal,* Modul *PL-Prüfung logisch LVE,* Modul *Messwertersetzung LVE* und Modul *Publikation*[[1]](#footnote-1).

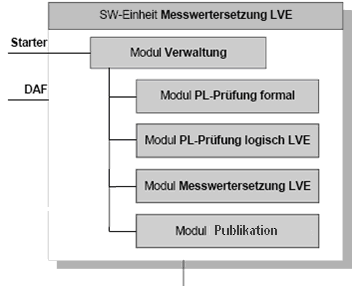


Abbildung 1.1: Zerlegung der SWE *Messwertersetzung LVE*

## Schnittstellen zur Umgebung

Die SWE *Messwertersetzung* *LVE* besitzt folgende Schnittstellen zu ihrer Umgebung:

* Schnittstelle zu einem Starterskript. Diese Schnittstelle wird verwendet, um die SWE zu starten. Sie ist in Abschnitt 2.3.2 beschrieben.
* Schnittstelle zum Datenverteiler. Über diese Schnittstelle wird die gesamte Kommunikation zu den übrigen Software-Einheiten abgewickelt. Die Verbindung wird über die Klasse ClientDavConnectionhergestellt. Die Klasse ist in [BibAPI] beschrieben.

## Realisierung der SW-Einheit

Die Beschreibung der Realisierung der SWE *Messwertersetzung* *LVE* ist vollständig gegeben durch die Beschreibung der Module, aus denen sie besteht.

## Lokale Daten

Die Beschreibung der lokalen Daten der SWE *Messwertersetzung* *LVE* ist vollständig gegeben durch die Beschreibung der lokalen Daten der Module, aus denen sie besteht.

## Ausnahmeverhalten

Das Ausnahmeverhalten der SWE *Messwertersetzung* *LVE* setzt sich zusammen aus dem Ausnahmeverhalten der Module, aus denen sie besteht.

# Modul Verwaltung

## Beschreibung des Moduls

Das Modul *Verwaltung* ist die zentrale Steuereinheit der SWE *Messwertersetzung* *LVE*. Seine Aufgabe besteht in der Auswertung der Aufrufparameter, der Anmeldung beim Datenverteiler und der entsprechenden Initialisierung der Module *PL-Prüfung formal*, *PL-Prüfung logisch LVE, Messwertersetzung* *LVE* sowie *Publikation*. Weiter ist das Modul *Verwaltung* für die Anmeldung der zu prüfenden Daten zuständig. Die Verwaltung gibt ein Objekt des Moduls *PL-Prüfung formal* als Beobachterobjekt an, an das die zu überprüfenden Daten durch den Aktualisierungsmechanismus weitergeleitet werden. Weiterhin stellt die Verwaltung die Verkettung der Module *PL-Prüfung formal*, *PL-Prüfung logisch* *LVE sowie* *Messwertersetzung* *LVE* sowie *Publikation* in dieser Reihenfolge her.

## Aufbau des Moduls

Der Aufbau des Moduls *Verwaltung* ergibt sich aus der Beschreibung der Realisierung in Kapitel 2.4.

## Schnittstellen zur Umgebung

Das Modul *Verwaltung* stellt die nachfolgend beschriebenen Schnittstellen zur Verfügung.

### Allgemeine Verwaltungsschnittstelle innerhalb der DUA

Diese Schnittstelle ist in [SwEnt\_ PlPrüfformal] beschrieben. Sie stellt allgemeine Funktionalitäten zur Verfügung (z. B. Ermittlung der betrachteten Systemobjekte).

### Schnittstelle zum Starter

Das Modul *Verwaltung* verarbeitet folgende Aufrufparameter, die vom Starter übergeben werden:

* -KonfigurationsBereichsPid:

PIDs der Konfigurationsbereiche, aus dem die Objekte für die *Messwertersetzung* *LVE* (Fahrstreifen) ermittelt werden sollen z. B. kb.objekteVerkehrXyz oder fs1:fs2:…

### Schnittstelle zum Datenverteiler

Das Modul *Verwaltung* hat eine Schnittstelle zum Datenverteiler. Über diese Schnittstelle wird die Anmeldung auf die zu überprüfenden Daten (siehe Tabelle 2‑1) durchgeführt. Die Verbindung wird über die Klasse ClientDavConnectionhergestellt und auch an untergeordnete Module weitergereicht. Die Klasse ist in [BibAPI] beschrieben.

|  |  |
| --- | --- |
| Objekt | typ.fahrStreifen **(KV)** |
| Attributgruppe | atg.verkehrsDatenKurzZeitIntervall, |
| Aspekt | asp.externeErfassung |
|  |  |
| Objekt | typ.datenflussSteuerung |
| Attributgruppe | atg.datenflussSteuerung |
| Aspekt | asp.parameterSoll |

Tabelle ‑: Daten, für die das Modul *Verwaltung* angemeldet ist

Die Attributgruppen sind im [DatK] beschrieben.

### Schnittstelle VerwaltungPL-Prüfung formal

Die Schnittstelle zwischen den Modulen *Verwaltung* und *PL-Prüfung formal* ist in Form der abstrakten Schnittstelle *BearbeitungsKnoten* gegeben (siehe [SwEnt\_ PlPrüfformal]). Diese wird vom Modul *PL-Prüfung formal* implementiert. Über sie findet die Rohdatenversorgung statt.

## Realisierung des Moduls

Das Modul *Verwaltung* besteht aus genau einer Instanz der Klasse:

**class** VerwaltungMessWertErsetzungLVE **implements** IVerwaltung

### Initialisierung

Die Initialisierung des Moduls *Verwaltung* läuft folgendermaßen ab:

* Die Klasse VerwaltungMessWertErsetzungLVE ist der Startpunkt der SWE *Messwertersetzung LVE*. Sie enthält die main()-Methode. Innerhalb dieser wird über die statische Methode StandardApplication.run(...) eine Instanz der Klasse VerwaltungMessWertErsetzungLVE als Empfänger aller Datenverteiler-Parameter initialisiert (ArgumentList, ClientDavInterface).
* Innerhalb der Methode parseArguments(ArgumentList arg0) findet das Auslesen und Speichern des Parameters KonfigurationsBereichsPid statt.
* Innerhalb der Methode initialisiere(ClientDavInterface dav) findet eine Ermittlung aller Objekte vom Typ typ.fahrStreifen statt, die innerhalb des übergebenen Konfigurationsbereichs definiert sind 🡪 Speicherung nach SystemObject[] objekte[[2]](#footnote-2). Schließlich wird die Instanziierung der Module *PL-Prüfung formal*, *PL-Prüfung logisch LVE,* *Messwertersetzung LVE* und *Publikation* durchgeführt:
* PlPruefungFormal plForm1 = new PlPruefungFormal();
* PlPruefungLogischLVE plLog1 = new PlPruefungLogischLVE();
* MessWertErsetzungLVE mwe = new MessWertErsetzungLVE();
* Publikation pub = new Publikation();

Diese Initialisierung der Module geschieht wie folgt.

### Initialisierung des Moduls PL-Prüfung formal

Bevor das Modul *PL-Prüfung formal* arbeiten kann, erfolgt die Initialisierung wie folgt:

* plForm1.setNaechstenBearbeitungsKnoten(plLog1, „SWE Messwertersetzung LVE“)[[3]](#footnote-3);
* plForm1.setPublikation(**true**);
* plForm1.initialisiere(**this**[[4]](#footnote-4));

Wie in [SwEnt\_ PlPrüfformal] beschrieben, ist das Modul *PL-Prüfung formal* damit arbeitsbereit und auf alle Parameter angemeldet.

### Initialisierung des Moduls PL-Prüfung logisch LVE

Nachdem das Modul *PL-Prüfung formal* initialisiert wurde kann das Modul *PL-Prüfung logisch LVE* initialisiert werden:

* plLog1.setNaechstenBearbeitungsKnoten(mwe, „SWE Messwertersetzung LVE“);
* plLog1.setPublikation(**true**);
* plLog1.initialisiere(**this**4);

Wie in [SwEnt\_ PlPrüflogLVE] beschrieben, ist das Modul *PL-Prüfung logisch LVE* damit arbeitsbereit und auf alle Parameter angemeldet.

### Initialisierung des Moduls Messwertersetzung LVE

Jetzt kann das Modul *Messwertersetzung LVE* initialisiert werden:

* mwe.setNaechstenBearbeitungsKnoten(plForm2, „SWE Messwertersetzung LVE“);
* mwe.setPublikation(**false**);
* mwe.initialisiere(**this**4);

Wie in Abschnitt 3.3.3 beschrieben, ist das Modul *Messwertersetzung LVE* damit arbeitsbereit und auf alle Parameter angemeldet. Innerhalb dieser SWE wird im Modul *Messwertersetzung LVE* keine Publikation der Daten vorgenommen.

### Initialisierung des Moduls Publikation

Zum Schluss kann das Publikationsmodul initialisiert werden:

* pub.setNaechstenBearbeitungsKnoten(**null**, „SWE Messwertersetzung LVE“);
* pub.setPublikation(**true**);
* pub.initialisiere(**this**4);

Wie in [SwEnt\_PlPrüflogUFD] beschrieben, ist das Modul *Publikation* damit arbeitsbereit und auf alle Parameter und insbesondere als Quelle zum Versenden von Daten angemeldet.

### Versorgung der Module PL-Prüfung formal, PL-Prüfung logisch LVE, Messwertersetzung LVE und Publikation mit zu plausibilisierenden Daten

Innerhalb des Moduls *Verwaltung* erfolgt eine Anmeldung als Empfänger für alle Objekte aus objekte unter den in Tabelle 2‑1 beschriebenen Attributgruppen/Aspekten. Diese Anmeldung erfolgt des Weiteren für alle Fahrstreifen-Objekte, die als Nachbarfahrstreifen bzw. Ersatzfahrstreifen aller Fahrstreifen aus objekte referenziert (und noch nicht angemeldet) sind. Die aufgrund der Anmeldung empfangenen Daten werden *permanent* über die Methode updateData(ResultData[] arg0) an das Modul *PL-Prüfung formal* weitergereicht.

Publikationsparameter werden über die Methode updatePublikation(…) an alle hier instanziierten Module weitergereicht.

## Lokale Daten

Es wird stets die Datenverteiler-Verbindung sowie die Menge aller zu plausibilisierenden Objekte vorgehalten.

## Ausnahmeverhalten

Es gibt folgende Arten von Ausnahmen:

* Für den Fall, dass keine Anmeldung beim Datenverteiler durchgeführt werden konnte, wird eine Meldung nach debug.error() ausgegeben und die Applikation beendet.
* Für den Fall, dass das Modul *PL-Prüfung formal*, *PL-Prüfung logisch LVE*, *Messwertersetzung LVE* oder *Publikation* nicht initialisiert werden konnte und eine DuAInitialisierungsException geworfen wurde, wird die Fehlermeldung nach debug.error() ausgegeben und die Applikation beendet.
* Für den Fall, dass einer der übergebenen Konfigurationsbereiche bzw. Fahrstreifen nicht existiert, wird die Fehlermeldung nach debug.warning() geleitet und der der fehlerhafte Konfigurationsbereich ignoriert.
* Für den Fall, dass eine Anmeldung auf Daten erfolglos war, wird eine Fehlermeldung nach debug.warning() ausgegeben.

# Modul Messwertersetzung LVE

## Beschreibung des Moduls

Das Modul *Messwertersetzung LVE* meldet sich auf alle benötigten Parameter (siehe Tabelle 3‑1) der vorgesehenen Objekte an. Es führt für die über die Methode updateData(ResultData[] arg0) übergebenen Kurzzeitdaten ggf. eine attributbezogene Messwertersetzung durch. Nach der Ersetzung werden die Daten je nach Parametrierung unter dem Aspekt asp.messWertErsetzung publiziert und an den nächsten Bearbeitungsknoten weitergereicht.

## Aufbau des Moduls

Das Modul *Messwertersetzung LVE* besitzt keine weitere Unterstrukturierung.

## Schnittstellen zur Umgebung

Das Modul *Messwertersetzung LVE* besitzt die nachfolgend beschriebenen Schnittstellen zur Umgebung.

### Allgemeine Schnittstellenbeschreibung BearbeitungsKnoten

Diese Schnittstelle ist in [SwEnt\_PlPrüfformal] beschrieben.

### Schnittstelle zum Datenverteiler

Das Modul *Messwertersetzung LVE* hat eine Schnittstelle zum Datenverteiler. Über diese Schnittstelle wird die Anmeldung auf die Parameter durchgeführt. Weiterhin wird darüber ggf. eine Quellenanmeldung für die Publikation der messwertersetzten Daten vorgenommen. Die Verbindung wird über ClientDavInterfacehergestellt (siehe [BibAPI]).

|  |  |
| --- | --- |
| Objekt | typ.fahrStreifen **(KV)** |
| Attributgruppe | atg.verkehrsDatenKurzZeitIntervallMessWertErsetzung |
| Aspekt | asp.parameterSoll |

Tabelle 3‑1: Vom Modul *Messwertersetzung LVE* interpretierte Parameter

|  |  |
| --- | --- |
| Objekt | typ.fahrStreifen **(KV)** |
| Attributgruppe | atg.verkehrsDatenKurzZeitIntervall |
| Aspekt | asp.messWertErsetzung |

Tabelle 3‑2: Vom Modul *Messwertersetzung LVE* durchgeführte Quellenanmeldung

### Schnittstelle VerwaltungMesswertersetzung LVE

Für die Verbindung zwischen einem Modul *Verwaltung* und dem Modul *Messwertersetzung LVE* werden folgende Methoden der in [SwEnt\_PlPrfüfformal] beschriebenen einheitlichen Schnittstelle *BearbeitungsKnoten* benötigt und implementiert.

* **void** initialisiere(IVerwaltung verwaltung) **throws** DuAInitialisierungsException:

Hier wird die Anmeldung auf alle Parameter aus Tabelle 3‑1 als Empfänger, sowie die Sendeanmeldung durchgeführt (so eine Publikation schon hier vorgesehen ist). Diese Methode wird erst verlassen, wenn die Anmeldungen erfolgreich waren. Sollte eine Anmeldung fehlgeschlagen sein, so wird eine DuAInitialisierungsException verursacht.

* **void** setNaechstenBearbeitungsKnoten(IBearbeitungsKnoten knoten):

Über diese Methode wird der nächste Verarbeitungsknoten übergeben und in diesem Modul gespeichert.

* **void** updatePublikation (ResultData[] arg0):

Über diese Methode wird das Publikationsverhalten des Moduls Messwertersetzung LVE analog zu den empfangenen Parametern angepasst. Es werden die vom Verwaltungsmodul empfangenen Daten interpretiert und ggf. Quellenanmeldungen zurückgenommen bzw. durchgeführt.

* **void** setPublikation(**boolean** publizieren):

Mit dieser Methode wird festgelegt, ob eine Publikation in den Datenverteiler stattfinden soll. Diese Information wird gespeichert.

Weiterhin werden für die Verbindung zwischen diesen beiden Modulen die folgenden Methoden der in [SwEnt\_PlPrüfformal] beschriebenen allgemeinen Verwaltungsschnittstelle benötigt:

* ClientDavInterface getVerbindung():

Über diese Methode kann von diesem Modul die Verbindung zum Datenverteiler erfragt werden.

* SystemObject[] getSystemObjects():

Über diese Methode kann von diesem Modul erfragt werden, welche Objekte zur Messwertersetzung vorgesehen sind.

### Schnittstelle zur Datenaktualisierung

Zur Datenaktualisierung wird die in [SwEnt\_ PlPrüfformal] beschriebene einheitliche Schnittstelle *BearbeitungsKnoten* benutzt. Folgende Methode wird dafür innerhalb des Moduls *Messwertersetzung LVE* implementiert.

* **void** updateData(ResultData[] arg0):

Über diese Methode werden dem Modul *Messwertersetzung LVE* die Daten zur Messwertersetzung übergeben. Dabei wird immer im Intervall n eine Messwertersetzung für das Intervall n-1 durchgeführt, da zu diesem Zeitpunkt sichergestellt ist, dass die Daten aller Fahrstreifen für das Intervall n-1 und n-2 schon innerhalb der Datenstruktur ErsetzungsTabelle zur Verfügung stehen[[5]](#footnote-5). Jedes einzelne Datum durchläuft hier folgende Arbeitsschritte:

* Zunächst wird die Methode aktualisiereErsetzungsTabelle(…) für den empfangenen Datensatz aufgerufen.
* Danach wird über die Methode isMWEMoeglich(…) überprüft, ob die grundsätzlichen Bedingungen erfüllt sind, die eine Messwertersetzung möglich machen.
* Fällt diese Überprüfung positiv aus, so wird die Methode ersetzeMessWerte(…) für den entsprechenden Datensatz aufgerufen.
* Zum Schluss werden die Werte des Intervalls (n-1) aus der Datenstruktur ErsetzungsTabelle für den entsprechenden Fahrstreifen ggf. publiziert und an den nächsten Bearbeitungsknoten weitergereicht.

## Realisierung des Moduls

Das Modul *Messwertersetzung LVE* wird durch eine Instanz der Klasse

**class** MessWertErsetzungLVE **implements** IBearbeitungsKnoten, ClientReceiverInterface, ClientSenderInterface

repräsentiert und über die Schnittstelle *BearbeitungsKnoten* initialisiert bzw. mit Daten versorgt. Die Initialisierung erfolgt innerhalb der DUA durch ein Modul *Verwaltung* wie in Abschnitt 2.4.4 beispielhaft beschrieben.

## Lokale Daten

Es wird für alle betrachteten Fahrstreifen jeweils der letzte Parameterdatensatz aus Tabelle 3‑1 vorgehalten (die Werte MaxErsetzungsDauer, MaxWiederholungZeit und MaxWiederholungAnzahl).

## Ausnahmeverhalten

Es ist kein spezielles Ausnahmeverhalten vorgesehen.

1. Die Module *PL-Prüfung formal* und *PL-Prüfung logisch LVE* sind in [SwEnt\_PlPrüfformal] bzw. [SwEnt\_PlPruflogLVE] beschrieben und werden hier nicht weiter erläutert. Das Modul *Publikation* implementiert die Schnittstelle *BearbeitungsKnoten* und dient nur der Publikation der übergebenen Daten in den Datenverteiler unter dem Aspekt asp.messWertErsetzung. Es besitzt ansonsten die gleiche Struktur wie das Modul *Publikation* aus [SwEnt\_PlPrüflogUFD] und wird hier ebenfalls nicht weiter erläutert. [↑](#footnote-ref-1)
2. Sollte kein Konfigurationsbereich übergeben worden sein, so werden alle Objekte vom Typ typ.fahrStreifen innerhalb des Standardkonfigurationsbereichs ermittelt. Diese Objekte werden über die Methode getSystemObjekte() anderen Modulen zur Verfügung gestellt. [↑](#footnote-ref-2)
3. An dieser Stelle findet die Verkettung der beiden Module *PL-Prüfung formal* und *PL-Prüfung logisch LVE* statt. [↑](#footnote-ref-3)
4. Der Verweis aus **this** bedeutet an dieser Stelle, dass dem Modul eine Referenz auf das aufrufende Verwaltungsmodul mitgegeben wird. [↑](#footnote-ref-4)
5. Sollten die Daten für ein Intervall X nicht zur Verfügung stehen oder die fraglichen Datensätze als *nicht ermittelbar* gekennzeichnet sein, so wird der betreffende Fahrstreifen für dieses Intervall als defekt interpretiert. Für die ersten beiden Intervalle beim Anlauf des Systems bedeutet das, dass keine Messwertersetzung wegen fehlender Werte durchgeführt werden kann. [↑](#footnote-ref-5)