



Whitepaper zur Literaturverwaltung

**Liabolo**

**„Signaturen und ihre Generierung in Liabolo“**  
von Thorsten Schlörmann

erstellt im Rahmen der Projektgruppe  
**„Virtuelle Organisation - Open Source“**

unter der Leitung von  
Prof. Uwe Schneidewind und Hendrik Eggers (Lehrbeauftragter)

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Signaturen im Bibliothekswesen</b>	<b>1</b>
2.1	Bremer Systematik . . . . .	1
2.2	Regensburger Verbundklassifikation . . . . .	2
<b>3</b>	<b>Signatur in Liabolo</b>	<b>3</b>
3.1	Motivation . . . . .	3
3.2	Signaturalgorithmus . . . . .	3
3.2.1	Natürlichsprachliche Beschreibung . . . . .	3
3.2.2	Graphische Darstellung . . . . .	4
3.2.3	Java-Code . . . . .	5
3.3	Netzwerkfähigkeit . . . . .	8
3.3.1	Problematik . . . . .	9
3.3.2	URI, URL, URN . . . . .	9
3.3.3	Umsetzung in Liabolo . . . . .	10
<b>4</b>	<b>Ausblick</b>	<b>10</b>

# 1 Einleitung

Liabolo ist in erster Linie ein Programm zur Literaturverwaltung; genauer gesagt verwaltet Liabolo Metadaten über Literatur. Innerhalb eines Literaturbestandes ist es wichtig, jedes Objekt aus dem Gesamtbestand identifizieren zu können. Dazu werden Signaturen verwendet. Jedem Medium wird eineindeutig<sup>1</sup> eine Signatur zugeordnet, d.h. wenn ein Buch gleich mehrfach im Gesamtbestand vertreten ist, so erhält jedes Exemplar eine eigene eineindeutige Signatur. Das unterscheidet die Signatur z.B. von der ISBN, wo jedes Exemplar eines Buches die gleiche Nummer hat<sup>2</sup>.

Anmerkung: Im folgenden Text wird häufig der Begriff „Buch“ stellvertretend für eine bibliothekarische Einheit benutzt werden.

## 2 Signaturen im Bibliothekswesen

Eine Signatur ist im Bibliothekswesen die Standortbezeichnung für ein Buch[1]. Die Signatur deutet also zusätzlich zu ihrer Funktion als eineindeutiger Schlüssel auch darauf hin, wo in der Bibliothek das Buch zu finden ist. Es gibt verschiedene Systeme, nach denen Bücher in der Bibliothek aufgestellt werden und anhand derer auch die Signatur gebildet wird. An dieser Stelle sollen zwei vorgestellt werden: die Bremer Systematik und die Regensburger Verbundklassifikation.

### 2.1 Bremer Systematik

Die Bremer Systematik wird an der hiesigen Bibliothek der Carl von Ossietzky Universität eingesetzt, aber natürlich auch, wie der Name vermuten lässt, an der Staats- und Universitätsbibliothek Bremen.

Jedes Buch wird dabei einem Fachbereich zugeordnet. Eine Auflistung der Fachbereiche der Staats- und Universitätsbibliothek Bremen ist online verfügbar<sup>3</sup>.

---

<sup>1</sup>mathematisch auch „bijektiv“ genannt

<sup>2</sup>das ist im mathematischem Sinne „surjektiv“

<sup>3</sup><http://suche.suub.uni-bremen.de/browse.html>

Die Signatur der UB Oldenburg setzt sich wie folgt zusammen:

- einem Kürzel für das Fachgebiet, bestehend aus drei Kleinbuchstaben,
- einer Systemstelle, die das Fachgebiet thematisch feiner untergliedert und aus drei Ziffern besteht,
- sowie einer Aufstellungsnummer. Sie wird aus zwei Großbuchstaben und einer vierstelligen Zahl gebildet. Diese Aufstellungsnummer gilt global für den Gesamtbestand und erhöht sich mit jedem Buch, das dem Bestand hinzugefügt wird.

An die Signatur kann noch weitere Information angehängt werden: eine Zahl für die Auflage eines Buches, ein Kleinbuchstabe für jedes weitere Exemplar und eine Zahl mit vorangestelltem Minuszeichen für die Bandnummer bei mehrbändigen Reihen.

## 2.2 Regensburger Verbundklassifikation

Die an der Universitätsbibliothek Regensburg entwickelte Regensburger Verbundklassifikation wird an einigen deutschen Bibliotheken eingesetzt, z.B. an der Bibliothek der Berliner Humboldt Universität oder der Bibliothek der Universität Rostock. Anhand dieser Verbundklassifikation wird die Signatur gebildet. Die besteht an der UB Regensburg aus folgenden Komponenten[2]:

- Lokalkennzeichen: Es gibt den Standort des Buches an und besteht aus einer Zahl, die mit einem Schrägstrich („Slash“) abgeschlossen wird.
- Notation: Sie ordnet das Buch in das Aufstellungssystem ein und gibt in verschlüsselter Form Auskunft über das Fachgebiet, dem das Buch zugehörig ist. Sie besteht aus zwei Großbuchstaben, wobei der erste Buchstabe die Hauptgruppe bezeichnet (z.B. Q für Wirtschaftswissenschaften), der zweite die Untergruppe angibt (z.B. QP für Allgemeine Betriebswirtschaftslehre) und eine mehrstellige Zahl die Feingruppe spezifiziert (z.B. QP500-QP550 für Produktion).

Danach können noch Zusätze an die Signatur angehängt werden:

- Verschlüsselung von Autornamen oder Titeln (Buchstaben und Ziffern),
- die Bandzahl (Gedankenstrich),
- die Auflage (in Klammern) und
- ein Mehrfachexemplar (Pluszeichen).

## 3 Signatur in Liabolo

### 3.1 Motivation

Liabolo dient der Verwaltung eines Literaturbestands. Mit anderen Worten kann man Liabolo also auch als Bibliotheksverwaltung bezeichnen. Nun ist Liabolo zwar nicht mit dem Ziel entwickelt worden, Literaturbestände in der Größenordnung einer Universitätsbibliothek zu verwalten, sondern eher für kleine und mittlere Bestände, doch auch hier ist der Einsatz einer Signatur sinnvoll.

Die Signatur dient in Liabolo als eindeutiger Schlüssel und als Mittel, den Gesamtbestand in Fachbereiche zu gruppieren und die Zugehörigkeit eines Buches darzustellen. Als Hinweis auf den Standort wird die Signatur in Liabolo nicht eingesetzt; hierfür werden Standorte verwendet, die der Benutzer selbst definieren kann.

### 3.2 Signaturalgorithmus

Die Signatur in Liabolo ist an die Signatur angelehnt, wie sie an der UB Oldenburg verwendet wird. Das lag für uns als Studenten der CvO-Universität Oldenburg nahe, da wir den Umgang mit dieser Signatur gewohnt sind. Aber auch die Tatsache, dass die Fachbereiche in der Signatur als Kürzel dargestellt und daher leicht zu identifizieren sind, erschien uns vorteilhaft, da dadurch die Transparenz der Signaturen erhöht wird.

Die Signatur besteht in Liabolo aus:

- dem Fachbereichskürzel,
- einem Unterstrich ('\_') und
- einer Aufstellungsnummer. Sie hat zwei Komponenten: eine zweistellige Großbuchstabenkombination und eine vierstellige Zahl.

Im Folgenden wird der Algorithmus zur Signaturbestimmung auf verschiedene Weise dargestellt. Einmal natürlichsprachlich, dann graphisch und zuletzt werden die wesentlichen Merkmale des Java Quellcodes vorgestellt.

#### 3.2.1 Natürlichsprachliche Beschreibung

An dieser Stelle wird versucht, den Signaturalgorithmus durch natürliche Sprache zu beschreiben, auch wenn solche Beschreibungen häufig recht umständlich klingen.

1. Bestimme den Fachbereich, dem das Buch zugeordnet werden soll und verwende das entsprechende Kürzel als erstes Element der Signatur.
2. Füge einen Unterstrich '\_' an die Signatur an.
3. Bestimme die Aufstellungsnummer. Das erste Buch erhält die Nummer AA0001, die Nummer erhöht sich für jedes folgende Buch wie folgt:
  - (a) Wenn die vierstellige Zahl kleiner als 9999 ist, so erhöhe sie um eins. Fahre mit Schritt 4 fort.
  - (b) Ist die Zahl 9999, so
    - i. setze sie auf 0001 und
    - ii. wenn der zweite Buchstabe der Buchstabenkombination nicht 'Z' ist, setze diesen auf auf den nächsten im Alphabet folgenden. Fahre mit Schritt 4 fort.
    - iii. Ist der zweite Buchstabe 'Z', so setze in auf 'A' und den ersten Buchstaben auf den nächsten im Alphabet folgenden. Fahre mit Schritt 4 fort.
    - iv. Ist der zweite und auch der erste Buchstabe 'Z', so sind keine Signaturen mehr generierbar und er Algorithmus schlägt fehl.
4. Füge die neu gewonnene Aufstellungsnummer an die Signatur an und beende den Algorithmus.

### 3.2.2 Graphische Darstellung

Der hier durch ein UML-Statechart graphisch dargestellte Signaturalgorithmus stellt nur eine grobe Skizze des oben beschriebenen Algorithmus dar, und soll lediglich der Veranschaulichung dienen. Der genaue Algorithmus folgt anschließend in Form des tatsächlichen Java-Codes.

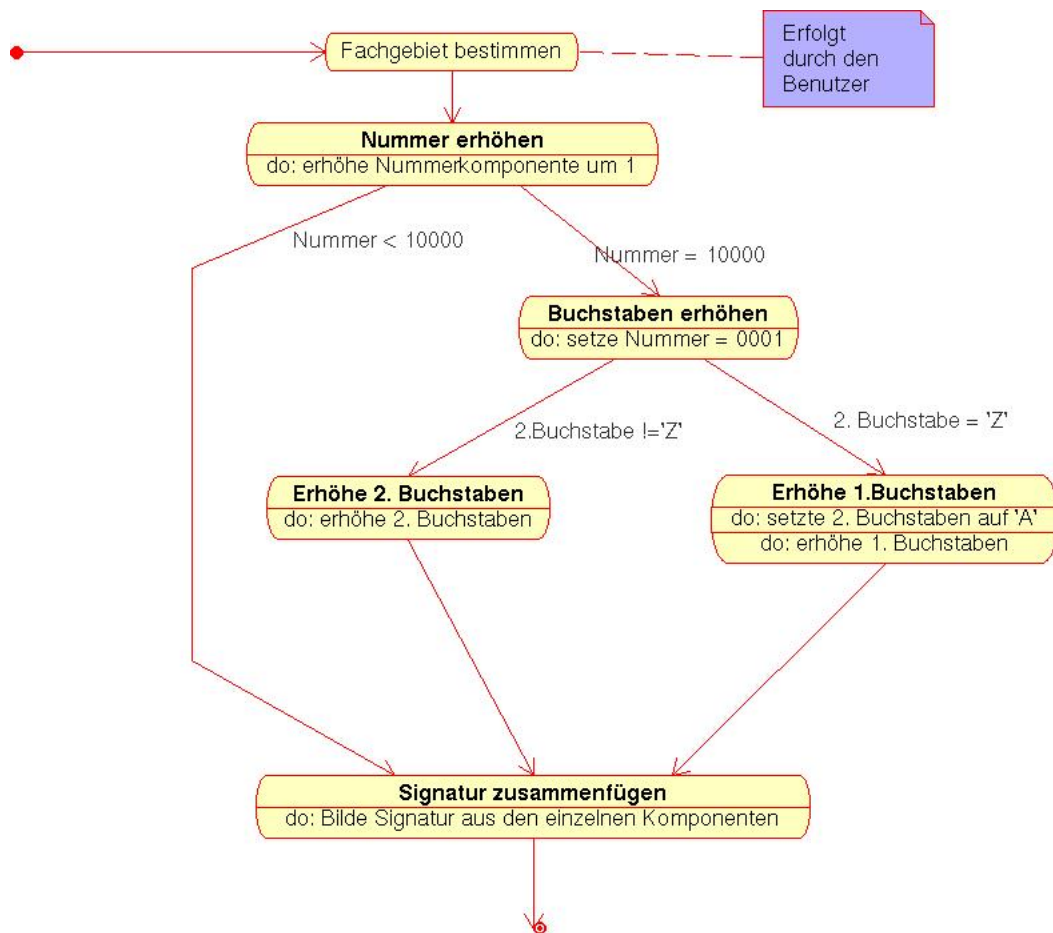


Abbildung 1: Skizze des Signaturalgorithmus als Statechart

### 3.2.3 Java-Code

Hier ist der Java Quellcode des Signatur-Generators. Einige für den Algorithmus unwesentliche Codeteile sind aus Gründen der Übersichtlichkeit entfernt worden. Außerdem mussten Formatierungen vorgenommen werden, um den Code auf die Breite der Seite anzupassen. Im Anschluss folgen einige Erläuterungen.

```

1 | package org.liabolo.common;
2 |
3 | import java.net.URL;
4 |
5 | public class SignatureGenerator {
  
```

```

6
7
8     static private int number;
9     static private String numberString;
10    static private char letter1, letter2;
11    static private boolean signatureRead = false;
12    static boolean changeLetter1 = false;
13    static boolean changeLetter2 = false;
14    static boolean emptyError = false;
15
16    static private void readSignatureNumber() {
17
18        number = Configurator.getIntProperty(
19            "signatureNumber", 0, "signature");
20
21        letter1 = Configurator.getProperty(
22            "signatureLetters", "ZZ", "signature").charAt(0);
23
24        letter2 = Configurator.getProperty(
25            "signatureLetters", "ZZ", "signature").charAt(1);
26
27        signatureRead = true;
28    }
29
30
31    public static Signature setSignature(String branch, String url){
32        String signatureString = "";
33
34
35        if (!signatureRead) readSignatureNumber();
36        number++;
37        if (number == 10000) {
38            number = 1;
39            changeLetter();
40            if (emptyError) return null;
41        }
42        numberString = Integer.toString(number);
43
44        switch (numberString.length()) {
45            case 1:
46                numberString = "000" + numberString;

```

```

47         break;
48     case 2:
49         numberString = "00" + numberString;
50         break;
51     case 3:
52         numberString = "0" + numberString;
53     }
54
55
56 // set properties and store them to file
57 Configurator.setProperty(
58     "signatureLetters", "" + letter1 + letter2, "signature");
59
60 Configurator.setProperty(
61     "signatureNumber", numberString, "signature");
62
63     signatureString = "" + letter1 + letter2 + numberString;
64
65     return new Signature(branch, signatureNumber, url);
66 }
67
68 private static void changeLetter() {
69     short charNr;
70     if (letter2 != 'Z') {
71         charNr = ((short) letter2);
72         charNr++;
73         letter2 = ((char) charNr);
74     } else {
75         if (letter1 == 'Z') emptyError = true;
76         letter2 = 'A';
77         charNr = ((short) letter1);
78         charNr++;
79         letter1 = ((char) charNr);
80     }
81 }
82
83 }
84

```

Erläuterungen zum Programm:

Die Methode *setSignature* in Zeile 31 wird beim Erstellen eines neuen Metadatensatzes aufgerufen. Als Argumente müssen die Branch (das Fachgebiet) und eine URL<sup>4</sup> angegeben werden. In Zeile 35 wird überprüft, ob die letzte Signatur schon aus der Konfigurationsdatei ausgelesen wurde. Das Auslesen ist nur für die erste Signaturerstellung nach dem Programmstart erforderlich; die Überprüfung erfolgt, um unnötige Dateizugriffe zu vermeiden.

In Zeile 36 wird die Nummer der Signatur um eins erhöht und anschließend überprüft, ob die Zahl schon 10000 erreicht hat (also nicht mehr vierstellig wäre). Ist das der Fall, wird die Nummer auf eins gesetzt (Zeile 38) und die Methode *changeLetter* aufgerufen (Zeile 39).

In der *changeLetter*-Methode (Zeile 68-81) wird die Buchstabenkombination erhöht. Dazu werden die Buchstaben in ihre entsprechenden ASCII-Nummern umgewandelt (Zeile 71 und 77) und dann inkrementiert. Abschliessend wird die ASCII-Nummer wieder in einen Buchstaben umgewandelt (Zeile 73 und 79).

Die Zeilen 42-53 dienen dazu, die Nummer in der Signatur immer in eine vierstellige Zeichenkette umzuwandeln, so dass z.B. aus der '5' eine '0005' wird.

Schließlich werden die Komponenten der Aufstellungsnummer der neuen Signatur in den Zeilen 57-65 in der Konfigurationsdatei gespeichert.

In Zeile 63 wird die Aufstellungsnummer zu einem String zusammengesetzt und danach ein Signaturobjekt erzeugt, in das auch der Fachbereich und die URL einfließen. Diese Signatur wird dann als Funktionswert der aufrufenden Methode zurückgegeben.

### 3.3 Netzwerkfähigkeit

Eine der Besonderheiten von Liabolo ist die Fähigkeit, einen Literaturbestand über ein Netzwerk zu verwalten. Hierzu kann ein Server eingerichtet werden, auf den von Liabolo aus Bücher hinzugefügt werden können und natürlich auch der Bestand durchsucht werden kann. Liabolo ist in der Lage,

---

<sup>4</sup>die Begründung für das Vorhandensein der URL folgt später

sowohl den lokalen Bestand zu verwalten, und auch als Client auf den Server zuzugreifen. Dadurch ist Liabolo gut geeignet für Betriebe, Organisationen, Vereine, etc. bei denen die Mitarbeiter bzw. Mitglieder einen eigenen Buchbestand pflegen, gleichzeitig aber auch bestimmte Bücher den Kollegen und Freunden auf dem Server sichtbar machen wollen.

### 3.3.1 Problematik

Durch die Netzwerkfähigkeit entstehen zwei Arten von Signaturen, die (lokalen) clientseitigen und die serverseitigen. Es muss in der Signatur deutlich gemacht werden, wo das betreffende Buch verwaltet wird, ansonsten kann es zu Mehrdeutigkeiten kommen.

Um die Problematik zu verdeutlichen, gebe ich hier ein Beispiel. Ein Benutzer verwaltet seinen eigenen Buchbestand, stellt aber auch Bücher in die Server-Datenbank ein. Fügt er seinem lokalen Bestand ein Buch hinzu, so generiert Liabolo eine Signatur für dieses Buch. Fügt er ein Buch auf dem Server hinzu, muss die betreffende Signatur vom Server erzeugt werden, andernfalls könnte es Kollisionen geben, wenn ein weiterer Benutzer Bücher (mit seiner lokalen Signatur) auf dem Server einstellt. Die Frage ist nun, wie der Benutzer seine lokalen Signaturen von denen des Servers unterscheiden kann, beispielsweise wenn er in den Beständen auf beiden Seiten suchen möchte? Die Signatur benötigt also ein Element, das angibt, woher die Signatur kommt. Dazu sollen nun zunächst die Begriffe URI, URL, URN erläutert werden.

### 3.3.2 URI, URL, URN

URI ist die Abkürzung für *Uniform Resource Identifier*. Ein URI ist eine kompakte Zeichenkette zur Identifizierung einer abstrakten oder physikalischen Ressource [3]. Eine Ressource kann alles sein, was einen Namen hat oder beschrieben werden kann, z.B. Dokumente, Bilder oder ein Service. Weiterhin versteht man unter URI den Oberbegriff für die Unterarten *Uniform Resource Locator*(URL) und *Uniform Resource Name*(URN).

URLs identifizieren eine Ressource über den Ort (im WWW), an dem sie zu finden ist, nicht über einen Namen oder Attribute [3]. Sie stellen also gewissermaßen einen Zeiger auf die Ressource dar.

URNs sollen als beständige, ortsunabhängige Bezeichner für Ressourcen dienen [4]. Ein URN bezeichnet eine Ressource eindeutig, d.h. ist sie mehrfach

im Netz vertreten, besitzt sie immer denselben URN, wo hingegen sie mehrere unterschiedliche URLs hat.

Die komplette Syntax von URIs ist im 'Request for Comment' (RFC) 2396 [3] angegeben.

### 3.3.3 Umsetzung in Liabolo

Soll ein Buch mittels Liabolo in die Datenbank des Servers eingestellt werden, erhält es eine Signatur, welche vom Server generiert werden muss. Diese Signatur wird dann Liabolo mitgeteilt. Dazu muss auf dem Server ein Web Service installiert sein. Dieser bekommt eine Anfrage von Liabolo, generiert eine Signatur und schickt diese als Antwort an Liabolo zurück.

Wir haben bereits festgestellt, dass die Signatur ein weiteres Element benötigt, das anzeigt, woher die Signatur kommt und damit auch, wo der Datensatz verwaltet wird. Wir benötigen also den Ort der Gültigkeit der Signatur und einen solchen Ort gibt man im Netz mit einer URL an, wie wir in Abschnitt 3.3.2 gesehen haben. Liabolo muss also an die vom Server empfangene Signatur noch die URL des Servers anheften, um deutlich zumachen, woher die Signatur stammt. Das geschieht, indem an die Signatur ein '@' angehängt wird, gefolgt von der URL des Servers. Zur Verdeutlichung bekommen die lokalen Signaturen statt der URL „localhost“ angeheftet.

Beispiel: inf\_AA0001@p30srv.wiwi.uni-oldenburg.de

## 4 Ausblick

Die Signatur stellt in Liabolo einen eindeutigen Schlüssel zur Identifikation eines Buches dar. Diese Eigenschaft ist auch im Netzbetrieb gewährleistet. Soll ein Server eingerichtet werden, muss jedoch der Web Service manuell vom Benutzer installiert werden und der Signaturgenerator angepasst werden, was einiges an Wissen über Liabolo erfordert und daher im Moment für einen Außenstehenden noch schwierig zu handhaben ist. Hier müssen sicherlich einige Erleichterungen geschaffen werden.

Außerdem wäre es wünschenswert, Zusatzinformationen über Auflage, Mehrfachexemplar und Bandnummer in die Signatur aufzunehmen. Dazu sind allerdings auch Änderungen bei der Eingabe der Metadaten nötig und ggf. eine Dublettenkontrolle, die überprüft, ob ein Buch möglicherweise bereits in der

Datenbank befindlich ist. So kann dann an die bestehende Signatur lediglich eine Auszeichnung als Mehrfachexemplar angefügt werden.

## Literatur

- [1] Wikipedia: Signatur. <http://de.wikipedia.org/wiki/Signatur>
- [2] Universität Regensburg: Regeln für die Signaturvergabe: <http://www.bibliothek.uni-regensburg.de/Systematik/sigreg.html>
- [3] Request for comment 2396. <http://www.ietf.org/rfc/rfc2396.txt>
- [4] Request for comment 2141. <http://www.ietf.org/rfc/rfc2141.txt>