

von *Michael Birkner*

### **Inhalt**

1. Was ist VuFind?
2. Installation des Testsystems
3. Technischer Überblick
4. Indexieren
5. Problem: Hierarchien
6. SolrMab – ein Indexer für MAB-Daten
7. SolrMab und mehrbändige Werke
8. Anzeige von Bänden
9. VuFind und Goobi
10. VuFind und Aleph
11. Layout
12. Was wir wollen
13. Zum Schluss

**Zusammenfassung:** In diesem Artikel geht es um die bisherigen Tests der AK Bibliothek Wien mit dem Discovery System VuFind. Es handelt sich um eine mögliche Open Source Alternative zu Primo, einer weit verbreiteten proprietären Suchlösung an österreichischen Bibliotheken. Nach einem kurzen technischen Überblick wird beschrieben, welche Entwicklungen und Anpassungen vorgenommen werden mussten, um eine Kompatibilität zwischen VuFind und den in der AK Bibliothek Wien vorherrschenden Bedingungen zu schaffen. Dabei wird näher auf die Verarbeitung des Datenformates MAB2 sowie die Schnittstellen zum Bibliothekssystem Aleph und dem Bibliotheks-Repository eingegangen. Die ersten Ergebnisse werden präsentiert und noch zu lösende Probleme angesprochen. Zudem wird ein Ausblick auf zukünftige Vorhaben gegeben.

**Schlüsselwörter:** VuFind; Discovery System; Suche; AK Bibliothek Wien

## VUFIND AT THE LIBRARY OF THE CHAMBER OF LABOUR IN VIENNA

**Abstract:** This article is about the tests that the Library of the Chamber of Labour in Vienna made with VuFind so far. VuFind is an open source software that could be used as an alternative to Primo, a widespread proprietary search engine at Austrian libraries. After a short technical overview, the article describes the developments and

*modifications that were necessary to guarantee the compatibility between VuFind and the conditions found at the Library of the Chamber of Labour in Vienna. Especially the processing of the MAB2 data type and the interfaces to the integrated library system Aleph and the digital repository of the library are treated. The first results as well as problems that still need to be solved are mentioned. The article also gives an outlook to future plans.*

**Keywords:** *VuFind; Discovery System; Search; Library of the Chamber of Labour in Vienna*

„Ich hab da was von etwas gehört, das heißt VuFind, kannst du dir das bitte mal ansehen?“ Dieser Satz zwischen Tür und Angel war der Start für die Arbeit mit VuFind an der AK Bibliothek Wien. Ausgesprochen hat ihn deren Leiterin Ute Weiner, gerichtet war er an Michael Birkner, zuständig für EDV-Angelegenheiten der Bibliothek und Autor des vorliegenden Textes.

## 1. Was ist VuFind?

Ohne große Umwege brachte mich eine Google-Suche zur Website von VuFind. Was dort zu erfahren war, klang recht beeindruckend. Wurde doch nichts weniger versprochen als ein Discovery-System, das alle möglichen Features mitbringt, die von Next-Generation-Systemen wie beispielsweise Primo aus dem Hause ExLibris bekannt sind. So werden unter anderem folgende Möglichkeiten geboten:

- Facetten zur Einschränkung der Suchergebnisse
- Indexierung mehrerer Quellen (z. B. Katalog, Repository, Datenbanken, Websites, Blogs, etc.)
- Browsen nach verschiedenen Kriterien
- Anlegen von Favoriten
- Crowdsourcing (Tagging, Kommentierungen)
- Einbindung von Normdaten
- Anzeige von „Ähnlichen Einträgen“
- Unterstützung mehrerer Sprachen
- ...

Umso mehr staunte ich, als ich erfuhr, dass dies alles als Open Source Software angeboten wird. Der Code von VuFind kann also kostenlos aus dem Internet heruntergeladen und eingesehen bzw. verändert werden. Wie

bei vielen solchen Projekten gibt es auch hier eine aktive Community rund um die Hauptentwickler, die an der Villanova University in den USA (Pennsylvania) angesiedelt sind. Von der Bezeichnung der Universität leitet sich übrigens auch der Name des Discovery Systems ab. Die Aktivitäten der Entwickler/innen- und Anwender/innen-Gemeinde reichen von Tagungen über Mailinglisten und Blogs bis hin zur Veröffentlichung von Zusatzmodulen für VuFind. Auch im deutschsprachigen Raum gibt es bereits eine Community, die sich regelmäßig trifft. So wurde das dritte VuFind-Anwendertreffen Ende September 2014 in Frankfurt a. M. veranstaltet. Das kommende wird voraussichtlich 2015 im süddeutschen Raum stattfinden.

## **2. Installation des Testsystems**

Zunächst etwas skeptisch, ob die Software ihre Versprechungen tatsächlich hält, las ich die recht gut beschriebenen Installations- und Konfigurationsanleitung und erfuhr, dass VuFind auf Linux oder Windows installiert werden kann. Ich entschied mich für die Linux-Distribution Ubuntu in der Version 12.04 LTS und installierte diese in eine virtuelle Umgebung mithilfe von Oracles VirtualBox. Der Installationsvorgang für VuFind selbst kann sehr einfach gehalten werden. In Ubuntu reicht ein einfacher Befehl, um das Discovery System mit allen in Abhängigkeit stehenden Softwarepaketen zu installieren. Will man mehr Kontrolle über die Installation des Ökosystems, können die einzelnen Softwarepakete manuell aufgesetzt werden. Dies kann allerdings etwas aufwändig und komplex sein. Denn neben VuFind selbst müssen ein Webserver, eine Datenbank, eine Java-Umgebung und PHP zum Laufen gebracht und miteinander in Verbindung gesetzt werden. Ist dies jedoch geschafft, kann VuFind gestartet werden.

## **3. Technischer Überblick**

Das Starten von VuFind ist im Grunde das „Hochfahren“ des Indexes. Bei diesem handelt es sich um den ebenfalls als Open Source Software zur Verfügung gestellten und weithin gebräuchlichen Solr Index, der von der Apache Software Foundation entwickelt wird. Er ist das Rückgrat von VuFind. Alle Daten, die durchsucht und schlussendlich den Nutzer/innen angezeigt werden, kommen aus diesem Index. Neben Solr ist für eine VuFind Installation – wie schon weiter oben erwähnt – eine MySQL-Datenbank, Java und PHP erforderlich. Dabei handelt es sich um typische, weit verbreitete

und auf dem aktuellen Stand der Technik gehaltene Webtechnologien, die in stabilen Versionen vorliegen. Außerdem sind sie alle frei zum Download verfügbar, was dem Open Source Gedanken von VuFind Rechnung trägt. Eine weitere Open Source Software, die zum Einsatz kommt, ist das in Java programmierte SolrMarc. Dabei handelt es sich um einen Indexer, der Daten, die im MARC21- oder MARCXML-Format vorliegen, in den Solr Index von VuFind bringt. SolrMarc ist also dafür verantwortlich, dass überhaupt Daten nach VuFind kommen. Und hier zeigt sich auch schon die erste Hürde, die zu nehmen war: Dafür zu sorgen, dass nicht nur MARC-Daten, sondern auch MAB-Daten indiziert werden können.

#### **4. Indexieren**

An dieser Stelle muss zunächst erwähnt werden, dass die AK Bibliothek Wien eine Teilnehmerin am Österreichischen Bibliothekenverbund (OBV) ist. Wie allgemein bekannt sein dürfte, wird im Verbund (noch) mit dem Bibliothekssystem Aleph und dem Datenformat MAB gearbeitet. Dieses wird von VuFind allerdings nicht von Haus aus unterstützt. Eine der ersten Fragen war also: Wie können wir unsere Daten in den Index von VuFind bringen? Prinzipiell sollte es so ablaufen, dass zunächst alle Daten aus Aleph exportiert werden und dann mithilfe des Indexers nach VuFind importiert werden. Aleph bietet auch die Möglichkeit, die bibliographischen Daten (die zwar im MAB-Format) vorliegen, als MARCXML zu exportieren. Zunächst hat es also den Anschein, dass die Hürde mit den Formatunterschieden aus dem Weg geräumt wäre: Es gibt mit SolrMarc einen Indexer für MARCXML-Daten und mit dem Export von Aleph eine Datei, die Daten in MARCXML beinhaltet. Es würde nur noch ein entsprechendes Mapping zwischen MAB- und MARC-Feldern vorgenommen werden müssen, was aufgrund der Konkordanzliste der DNB (<http://nbn-resolving.de/urn:nbn:de:101-2008032106>) auch keine große Herausforderung gewesen wäre. Doch wie so oft liegt der Teufel im Detail.

#### **5. Problem: Hierarchien**

Denn mit dem SolrMarc-Indexer sind keine Hierarchien abbildbar und VuFind ist dafür von Haus aus auch nicht ausgestattet. Es geht in erster Linie um die mehrbändigen Werke, mit denen bekanntermaßen auch andere moderne Discovery Systeme ihre Probleme haben. Natürlich wäre es

ein Einfaches gewesen, auf eine ordentliche Darstellung zu verzichten und keine Hierarchien abzubilden, die einzelnen Bände also nicht unter ihrem übergeordneten Datensatz anzuzeigen. Lösungen dafür gibt es bereits. Beispielsweise der Rückgriff auf die Bandanzeige im alten OPAC. Dazu muss dieser im Hintergrund allerdings noch aktiv sein, was wahrscheinlich irgendwann nicht mehr der Fall sein wird. Dieser Ansatz ist also keinesfalls zukunftssicher, abgesehen davon ist er recht unschön. Momentan scheint sich die Variante etabliert zu haben, im übergeordneten Datensatz die für Benutzer/innen eher kryptisch anmutende und mit einem Link hinterlegte AC-Nummer des Verbundes anzuzeigen. Nach einem Klick darauf öffnet sich ein Popup, in dem – meist völlig unsortiert – die einzelnen Bände angezeigt werden. Nach der Meinung meiner Kolleg/innen in der AK Bibliothek Wien und mir selbst ist auch diese Lösung alles andere als optimal. Beide konnte ich übrigens zum Zeitpunkt des Verfassens dieses Textes (Anfang November 2014) im Katalog der UB Wien vorfinden (siehe z. B. <http://ub-data.univie.ac.at/AC00168374>). Da für uns also keiner dieser Ansätze in Frage kam, andere aber mit den bordeigenen Mitteln von VuFind nicht zu realisieren waren, musste eine eigene Lösung entwickelt werden. Vor allem die Art und Weise, wie die Daten indexiert werden, ist hierfür ausschlaggebend. Zum Glück ist VuFind diesbezüglich sehr flexibel. In der Dokumentation ist folgender Satz zu finden: „[...] VuFind also has the capability to support any [...] form of metadata if you are willing to do at least one of two things“. Das Erste dieser beiden Dinge ist obligatorisch. Es muss ein eigener Indexer für das zu indexierende Metadatenformat entwickelt werden. Das Zweite ist optional, aber durchaus sinnvoll, nämlich einen so genannten „Record Driver“ zu erstellen.

## **6. SolrMab – ein Indexer für MAB-Daten**

An die Erstellung eines eigenen Indexers habe ich mich aufgrund meiner glücklicherweise vorhandenen Programmierkenntnisse sogleich gewagt und ihn – in Anlehnung an „SolrMarc“ – „SolrMab“ genannt. SolrMab kann Exporte von Aleph verarbeiten, die über das Service „print-03“ im Katalogisierungsmodul oder den Aleph Publisher erstellt wurden. Wichtig ist allerdings, dass die SYS-Nummer mitexportiert wird, denn diese benötigt VuFind für die Abfrage von Exemplardaten über die Schnittstellen von Aleph. Aus diesem Grund ist ein Export des Aleph Publishing Systems zu bevorzugen, da dieser die SYS-Nummer standardmäßig mit ausliefert. Obwohl es sich bei den Daten aus Aleph um MAB-Daten handelt, müs-

sen sie als MARCXML exportiert werden, da sie so leicht weiterzuverarbeiten sind. Die dabei entstehenden XML-Dateien werden von SolrMab Zeile für Zeile gelesen und die Werte der einzelnen Felder in den Solr Index geschrieben, der ebenfalls aus verschiedensten Feldern besteht. So wird beispielsweise der Wert des für den Hauptsachtitel vorgesehenen MAB-Feldes 331 in das Index-Feld „title“ geschrieben. Das Mapping, also welches MAB-Feld in welches Index-Feld geschrieben wird, kann in einer recht einfach handzuhabenden Datei nach eigenen Wünschen konfiguriert werden.

## **7. SolrMab und mehrbändige Werke**

Das Ziel der Entwicklung von SolrMab war natürlich, bibliographische Daten auf die Art und Weise in den Index von VuFind zu schreiben, dass Hierarchien ordentlich abbildbar sind. Mehrbändige Werke sollen also berücksichtigt werden. Um dies zu erreichen, findet ein zweistufiger Indexvorgang statt. In einem ersten Schritt werden lediglich die übergeordneten H-Sätze indexiert. In einem zweiten Schritt werden alle untergeordneten U-Sätze indexiert. Dabei wird für jeden dieser U-Sätze geprüft, ob die AC-Nummer seines H-Satzes, die in seinem MAB-Feld 010 stehen sollte, bereits im Index vorhanden ist. Sie sollte ja schon im ersten Schritt beim Indexieren der H-Sätze in den Index geschrieben worden sein. Falls der H-Satz also vorhanden ist, wird ihm die AC-Nummer des U-Satzes in ein eigenes Index-Feld geschrieben. Somit „weiß“ der H-Satz also, welche AC-Nummern seine U-Sätze haben.

## **8. Anzeige von Bänden**

Dies wiederum nützt der bereits vorhin erwähnte Record Driver für die Anzeige der einzelnen Bände in der Benutzeroberfläche. Er ist also dafür verantwortlich, Daten zu aus einer Quelle wie z. B. dem Solr Index zu holen und dem Benutzer anzuzeigen. Einen eigenen Record Driver zu erstellen ist zwar kein Muss, jedoch durchaus empfehlenswert. Er kann nämlich für jeden Datentyp eine eigene Anzeige generieren. Ein Beispiel: In VuFind könnte auch die eigene Bibliotheks-Homepage indexiert und durchsuchbar gemacht werden. Somit könnten Benutzer/innen nicht nur den Bibliotheksbestand, sondern auch Infos zur Bibliothek selbst suchen. Es würde allerdings kaum Sinn machen, die Daten von der Homepage in einer Suchtreffer-Anzeige auf dieselbe Art wie die Metadaten zu einem Buch darzustellen, für das es Exemplarinformationen, Verfügbarkeitsdaten, etc. gibt. Deshalb

käme jeweils ein eigener Record Driver für die Homepagedaten und die Metadaten des Buches zum Einsatz, der die jeweiligen Such-Ergebnisse optimal aufbereiten und anzeigen würde. Aus diesem Grund entwickelte ich also auch einen Record Driver für unsere MAB-Daten. Das Ziel dabei war vor allem, einen Tab namens „Bände“ und darin ein Tabelle für die Einzelbände eines mehrbändigen Werkes anzuzeigen. Das (vorläufige) Ergebnis ist auf der folgenden Abbildung zu sehen. Die Links, mit dem die Einträge der Bände hinterlegt sind, führen natürlich zum jeweiligen Datensatz des einzelnen Bandes.

**Philosophie**

Publikationsform: mehrbändiges begrenztes Werk - nicht Teil eines Gesamtwerkes

Sprache: ger

Veröffentlicht: Springer

Schlagworte: [Philosophie](#)

Tags: [Metaphysik](#) (1), [Philosophie](#) (1) [Tag hinzufügen](#)

Exemplare **Bände** Beschreibung Kommentare Internformat

Titel	Band Nr.	Erscheinungsjahr	Ausgabe
<a href="#">Philosophische Weltorientierung</a>	1	1956	3. Aufl.
<a href="#">Existenzerhellung</a>	2	1956	3. Aufl.
<a href="#">Metaphysik</a>	3	1956	3. Aufl.

## 9. VuFind und Goobi

Neben unseren Katalogdaten wollen wir natürlich auch die Daten unseres Repositories in VuFind indexieren und somit für unsere Benutzer/innen auffindbar machen. Als Repository-System nutzen wir Goobi, ebenfalls eine Open Source Software, mithilfe derer wir Meta- und Strukturdaten erfassen. Beispielsweise wird für die elektronische Version eines Zeitschriftenheftes jeder darin enthaltene Artikel gekennzeichnet und mit Metadaten (Artikeltitel, Autor/innen, etc.) versehen, um ihn später in einem eigenen Goobi-Frontend entsprechend anzeigen zu können. Die dabei entstandenen Daten können über eine OAI-Schnittstelle abgerufen werden. Sie

steht allen öffentlich unter <http://emedien.arbeiterkammer.at/viewer/oai/> zur Verfügung. Auch die AK Bibliothek Wien selbst will diese Schnittstelle nutzen, um VuFind zu „füttern“. Allerdings gibt es ein Problem: Der Großteil der Daten, die in Goobi vorhanden sind, werden mit großer Wahrscheinlichkeit bereits in VuFind vorhanden sein. Der Grund: Beide Systeme werden mit Daten aus Aleph versehen. Wie bereits weiter oben ausführlich beschrieben, wird der VuFind-Index mit einem Export aus Aleph versorgt. Auch in Goobi werden für die elektronischen Publikationen Daten aus Aleph importiert. So wird beispielsweise auch die AC-Nummer übernommen. Wenn nun also der Export aus Aleph und anschließend die OAI-Schnittstelle von Goobi nach VuFind indexiert werden, kann es zu Doppelungen von Datensätzen kommen. Dies soll natürlich vermieden werden. Für die Indexierung der Goobi-Daten muss daher ein Skript entwickelt werden, das im VuFind-Index zunächst prüft, ob bereits ein Datensatz mit der entsprechenden AC-Nummer existiert. Falls ja, soll nur ein Link zur elektronischen Version der Publikation im Goobi-Frontend eingefügt werden. Falls nein, sollen alle Daten aus Goobi indexiert werden. Dieses zum Teil bereits realisierte aber nicht fertiggestellte Skript hat zur Zeit noch keine hohe Priorität, da zunächst sichergestellt werden muss, dass die Daten aus Aleph ordentlich nach VuFind indexiert werden, was trotz aller bisherigen Fortschritte wohl noch einiges an Arbeitsaufwand erfordern wird.

## 10. VuFind und Aleph

Ein sehr wichtiger Aspekt bei der Verwendung von VuFind ist natürlich die Integration mit Aleph. Ohne Kommunikation zwischen den beiden Systemen können keine Exemplardaten abgefragt werden, das einloggen in die Benutzer/innen-Konten wäre nicht möglich, Vormerkungen können nicht getätigt werden, etc. Glücklicherweise ist VuFind von Haus aus für die Zusammenarbeit mit Aleph ausgelegt. Dazu werden dessen X-Server und REST-Schnittstelle genutzt. Da wir an der AK Bibliothek Wien zurzeit leider keine dieser beiden Interfaces zur Verfügung haben, stellte uns die Oberösterreichische Landesbibliothek (OÖLB) ihr Aleph-Testsystem zur Verfügung. Nachdem wir einen Abzug deren Daten erhielten und unsere IP-Adresse für ihre Schnittstellen freigeschaltet wurde, konnten die Tests starten. Dazu mussten wir zunächst VuFind mit dem Aleph-Testsystem der OÖLB bekannt machen. Um dies zu tun, muss der so genannte Aleph Driver angepasst werden. Dabei handelt es sich um eine Datei, die mit einem normalen Texteditor verändert werden kann. Es müssen beispielsweise



die Adresse zum Aleph-Server und Angaben zu den verschiedenen Datenbanken (LBO01, LBO50, etc.) eingetragen werden. Waren die richtigen Werte einmal gesetzt, wurden in VuFind auch tatsächlich Exemplarinformationen zum jeweiligen Werk direkt aus Aleph abgefragt und angezeigt, wie auf folgendem Screenshot zu sehen ist:



Neben den Verfügbarkeitsinformationen werden auch Links zur Bestellung oder Vormerkung angezeigt. Eingeloggte BenutzerInnen können diese Vorgänge direkt auslösen. Für die Authentifizierung stehen übrigens verschiedene Möglichkeiten zur Verfügung, die auch kombiniert werden können. Einerseits können sich NutzerInnen über das Aleph-Konto (Strichcode und Strichcode-Verifikation) oder ein eigens in VuFind angelegtes Konto einloggen. Andererseits werden auch LDAP und Shibboleth sowie SIP2 unterstützt. Die Funktionen des Benutzer/innen-Kontos reichen von der Anzeige ausgeliehener bzw. vorgemerakter oder bestellter Medien über die Abfrage von Gebühren bis hin zum Anlegen von Favoriten, Speichern von Suchen und Ändern der persönlichen Daten.

Wie an den Screenshots oben zu erkennen ist, funktioniert die Kommunikation zwischen Aleph und VuFind problemlos. Leider ergaben sich trotzdem einige Schwierigkeiten, die jedoch hauptsächlich auf Seiten der

Aleph-Schnittstellen zu finden sind. Zum einen sollten lt. Schnittstellen-Dokumentation von ExLibris mehr Daten über REST ausgegeben werden, als es im Testsystem der OÖLB tatsächlich der Fall ist. VuFind bekommt somit nicht alle Informationen, die für gewisse Vorgänge vonnöten wären. Zum anderen werden einige Kommandos der Schnittstellen überhaupt nicht ausgeführt. Es werden lediglich Fehlermeldungen zurückgegeben. Daten, mit denen VuFind arbeiten muss, fehlen in diesen Fällen zur Gänze. Diese Problematik kann möglicherweise durch eine abgeänderte Konfiguration der Schnittstellen behoben werden. An dieser Stelle kann ich jedoch auch noch eine positive Entwicklung verkünden. Nach meinem Vortrag auf der ODOK 2014 fanden mehrere Gespräche mit Wolfram Seidler von der UB Wien, seinem System-Team und mir statt. Die Kolleg/innen zeigten sich an der Arbeit mit VuFind an der AK Bibliothek Wien sehr interessiert und boten an, auch ihr Aleph-Testsystem zur Verfügung zu stellen. Nach der Übersendung eines größeren Datenabzuges, der auch für eine Belastungsprüfung des SolrMab-Indexers herangezogen werden kann, laufen die Tests zum Zeitpunkt des Verfassens dieses Textes, also Anfang November 2014, gerade an. Erste Verbindungsversuche zu den Schnittstellen der UB Wien ergaben, dass im Gegensatz zu den Schnittstellen der OÖLB alle Daten, die VuFind benötigt, auch ausgegeben werden. Es kann also davon ausgegangen werden, dass alle Funktionen von VuFind korrekt mit dem System der UB Wien zusammenarbeiten. Ob dem wirklich so ist, werden hoffentlich die nächsten Wochen zeigen.

## 11. Layout

Bezüglich der Gestaltung des Layouts von VuFind kann gesagt werden, dass es relativ einfach ist, Anpassungen vorzunehmen. Es wird mit Themes gearbeitet, also einem Satz an hauptsächlich HTML- und CSS-Dateien, die für die verschiedenen Ansichten (z. B. Suchtrefferliste, Detailsicht eines Datensatzes, Ansicht des Benutzerkontos, etc.) verantwortlich sind und individuell abgeändert werden können. Die einfachste Variante besteht darin, das Standard-Theme zu kopieren und auf dieser Basis das eigene Corporate Design umzusetzen. Anschließend muss VuFind nur noch gesagt werden, welches Theme es zu verwenden hat. Natürlich können auch die angezeigten Texte geändert werden. Auf eine Abbildung von Screenshots zur Illustrierung der Möglichkeiten, die VuFind in grafischer Hinsicht bietet, wird an dieser Stelle verzichtet, um nicht unnötig Seiten zu füllen. Wer sich jedoch die Layouts verschiedenster Installationen ansehen will, sei auf die Seite

[https://vufind.org/wiki/installation\\_status](https://vufind.org/wiki/installation_status) hingewiesen, wo zahlreiche Links zu VuFind-Instanzen zu finden sind, durch die man sich klicken kann.

## 12. Was wir wollen

Zum Schluss möchte ich kurz erläutern, was wir uns in der AK Bibliothek Wien von der Beschäftigung mit VuFind erhoffen. Zunächst haben wir das Ziel, eine Open Source Alternative zum im Österreichischen Bibliothekenverbund bereits weit verbreiteten Discovery System Primo zu schaffen. Es ist bekannt, dass diese proprietäre Software der Firma ExLibris gerade für viele Einrichtungen mit kleinerem Budget eine nicht unerhebliche finanzielle Last darstellen kann. Natürlich gibt es mit dem Modell „Primo Basic“ der OBVSG auch eine kostenschonende Variante, jedoch bietet diese für viele sicherlich nicht den Funktionsumfang, den moderne Bibliotheken erwarten, handelt es sich dabei doch lediglich um einen reinen Ersatz des OPAC ohne Möglichkeit zur Indexierung zusätzlicher Quellen. Natürlich ist VuFind zunächst auch nicht mehr als das, aber immerhin soll diese Basisversion nach ihrer Fertigstellung kostenlos allen interessierten Bibliotheken zur Verfügung gestellt werden. Der bloße Ersatz des OPAC ist außerdem nur als erster Schritt geplant. Anschließend sollen nämlich auch standardisierte Verfahren zum Indexieren weiterer Quellen wie etwa einem Repository (z. B. Goobi, Visual Library, etc.) oder Datenbanken (z. B. DOAJ, PubMed, etc.) angeboten werden – natürlich ebenfalls Open Source und kostenfrei. An dieser Stelle kann auch erwähnt werden, dass seit VuFind 2.3 auch Primo Central sowie die Schnittstellen zu EBSCO EBS und EIT standardmäßig unterstützt werden (siehe dazu Changelog zum Release 2.3 unter [https://vufind.org/wiki/changelog#release\\_23\\_-\\_8\\_11\\_2014](https://vufind.org/wiki/changelog#release_23_-_8_11_2014)).

## 13. Zum Schluss

Natürlich muss auch folgendes ganz klar gesagt werden: Selbst wenn VuFind kostenlos ist, müssen gewisse Ressourcen investiert werden. Tatsache ist, dass nicht jede Einrichtung Personen zur Verfügung hat, die beispielsweise das Layout in HTML- und CSS-Dateien anpassen können. Auch gibt es wohl in fast jeder Bibliothek gewisse Eigenheiten, für die spezielle Entwicklungen nötig wären. All dies ist sehr arbeitsintensiv und erfordert entsprechendes Knowhow. Sollte VuFind in Österreich tatsächlich zum Einsatz kommen, muss irgendetwas diese Entwicklungen durchführen bzw. finan-

zieren, denn die Bibliothek der Arbeiterkammer Wien kann dies natürlich nicht alleine leisten. Daher ist es unerlässlich, eine Community aufzubauen, in der alle Anwender-Bibliotheken und die OBVSG vertreten sind, um sich gegenseitig zu unterstützen und einen gewissen Support zu bieten. Ein Vorbild dafür könnte die deutsche Community sein, die sich regelmäßig trifft und austauscht. Es müssen auch Szenarien wie z. B. die gemeinsame Finanzierung einer Person für die Weiterentwicklung von VuFind angedacht werden. Durch die Einsparung der Kosten eines proprietären Systems könnte dies in einer gemeinsamen Anstrengung durchaus machbar sein.

Wie an meinen obenstehenden Ausführungen abzulesen ist, sind noch einige Fragen und Problemstellungen offen, und es werden sicherlich noch viele hinzukommen. VuFind ist dennoch eine echte und ernstzunehmende Alternative zu anderen Discovery Systemen. Die Tatsache, dass es sich um eine Open Source Software handelt, öffnet viele Möglichkeiten. Neben der Kostenersparnis können Erweiterungen genau nach Wunsch entwickelt werden. Außerdem setzt VuFind auf bewährte Technologien, die stabil, aber auch zukunftssicher sind. Die Abhängigkeit von einem einzigen Unternehmen, das schlimmstenfalls in Konkurs gehen kann und seine Software für Weiterentwicklungen nicht freigibt, fällt weg. Diese und andere Tatsachen überzeugen mich davon, weiterhin Zeit und Arbeit in VuFind zu investieren. Zu guter Letzt möchte ich mich bei der Oberösterreichischen Landesbibliothek und der Universitätsbibliothek Wien für ihre Unterstützung danken. Ein weiterer Dank geht an alle, die bei der ODOK 2014 im Rahmen des ÖGDI-Preises für meinen Vortrag gestimmt haben. Die große Unterstützung, durch die ich den Preis völlig überraschend gewinnen konnte, zeigt mir, dass das Thema Discovery Systeme viele Kolleg/innen beschäftigt. Es bestärkt mich zudem in der Überzeugung, auf dem richtigen Weg zu sein und spornt mich an, ihn weiterhin zu beschreiten. Vor diesem Hintergrund hoffe ich, Ihnen in Zukunft weitere Fortschritte berichten zu können. Natürlich sind Sie auch gerne eingeladen, sich mit mir in Verbindung zu setzen, wenn Sie Interesse an VuFind haben.

Mag. Michael Birkner  
AK Bibliothek Wien für Sozialwissenschaften  
Prinz Eugen-Straße 20-22, A-1040 Wien  
E-Mail: [Michael.Birkner@akwien.at](mailto:Michael.Birkner@akwien.at)



Dieses Werk ist lizenziert unter einer [Creative-Commons-Lizenz Namensnennung 3.0 Österreich](https://creativecommons.org/licenses/by/3.0/au/).