allegro-C : Innovationswerkzeuge für Bibliotheken

Bernhard Eversberg (UB Braunschweig) zum Bibliothekartag Berlin 2011

Ein Integriertes Bibliothekssystem (engl. ILS) ist längst keine Innovation mehr. Man mag sich daher wundern, was denn wohl "allegro" zu suchen habe in einer Veranstaltung, in der es um "Innovative Lösungen" geht. Einige Hinweise sollen deshalb zunächst einmal klären, was im Titel mit „Innovationswerkzeuge“ gemeint ist:

1. *allegro* ist **kein Integriertes Bibliothekssystem**. Man kann aber eines draus machen, denn es ist ein **konfigurierbares, skriptfähiges Datenbanksystem**.

2. Im Kern ist es **anwendungsneutral**, man könnte sagen „unbibliothekarisch“. Das ausgelieferte „Gesamtpaket“ enthält neben den Programmen alle Zutaten (und zwar Skripte, Parameter, Hilfetexte usw.), die zu einem ILS nötig sind. Diese Zutaten kann man beiseitelassen und das System auf eigene Weise für andersartige Projekte mit anders strukturierten Daten nutzen, d.h. mit eigener Konfiguration, eigenen Skripten, Parametern usw.

3. Vielleicht wäre *allegro* trotzdem schon von der Bildfläche verschwunden, hätte es nicht **eine lange Kette von Innovationen** erlebt, die noch nicht zu Ende ist. Ein ILS mag heute zur Grundausstattung von Bibliotheken gehören, aber es muß Schritt halten mit dem innovativen Wandel, soll die Bibliothek darin bestehen können.

4. *allegro* ist das **Gegenteil einer black box**. Das heißt konkret: es überläßt dem ambitionierten Endnutzer alle Macht für eigene Innovationen in neuen Projekten. Diese Weltoffenheit wird vollendet mit der Freigabe der Quellcodes als OpenSource-Paket bis Ende 2012. Das Bibliothekswesen hat dann eine vollständig frei verfügbare und frei veränderbare Software mit einem breiten Anwendungspotential.

5. Damit *allegro* offen bleibt für weitere Innovationen, ist eines besonders wichtig: die **Unabhängigkeit der Datenbank von externen Normen**. Die Datenbanksoftware selbst muß sich deshalb nicht wandeln, wenn Neuerungen an der Oberfläche (einschl. Schnittstellen) gewünscht werden. Dies gilt auch für die Zeichencodierung, aber vor allem für das Datenformat: Hier gibt es keine Festlegung auf externe Standards wie MAB2, MARC21, Unicode, XML (nicht zu reden von RAK) und keine Notwendigkeit, die Wandlungen solcher Normen intern nachzuvollziehen.

6. Auch wer als "Nur-Anwender" nichts als eine fertige Anwendung möchte, profitiert von den Innovationen und der Offenheit, denn es wird ein bibliothekarisches „Gesamtpaket“ bereitgestellt und dieses immer wieder mit Neuerungen angereichert, die der Anwender stets ohne aufwendige Reorganisationen nutzen kann. Technisch gesprochen haben wir eine **Entkoppelung von Syntax und Semantik**: die *allegro*-Programme, d.h. das Datenbanksystem, kennt nur die *Struktur* der Daten, und die ist seit langem stabil; mit der *Bedeutung* haben die Programme nichts zu tun, die ist Sache externer, frei zugänglicher Konfigurations-, Parameter- und Skriptdateien. Das betrifft auch z.B. Art und Umfang der Indexierung und die Konversion von Fremddaten. Mit andern Worten: höhere Funktionen sind nicht in C sondern in höheren Sprachen programmiert, die seit langem offen und vollständig dokumentiert sind, auch viele Nichtprogrammierer kennen sich inzwischen damit aus, d.h. eine Abhängigkeit von „Braunschweig“ besteht in vielen Dingen längst nicht mehr.

Die Präsentation auf dem Bibliothekartag 2011 soll den Stand der Entwicklung umreißen, und sie ist ihrerseits eine Innovation: integriert in die neueste Web-Oberfläche namens **a30** zeigt sie einen Überblick über die wichtigsten Aspekte des Systems, wobei dieses selbst zugleich ausprobiert werden kann an einer realen Datenbank: (es handelt sich um die RDA-Testdaten der LC)

http://www.allegro-c.de/db/a30/allegro.htm

Dieses Papier: http://www.allegro-c.de/doku/a30/berlin2011.pdf

a30 ist eine **Rich Internet Application** auf der Basis von Adobe Flash. Damit können nun Web-Anwendungen mit einem Minimum an Skripting erstellt werden: Praktisch entfallen die gängigen Skriptsprachen und Normen wie PHP, Perl, Java, JavaScript, HTML, CSS und XML; man braucht fast nur noch die *allegro*-eigene Skriptsprache **FLEX**, die zur Kommunikation mit der Datenbank ohnehin nötig ist. Im Vergleich zu **SQL**, das ja immer für den Verkehr mit einer relationalen Datenbank wie MySQL genutzt wird, kann FLEX auch noch fast alles andere übernehmen, wozu man sonst eben Java, PHP etc. verwendet, und a30 funktionert dann browserunabhängig ohne JavaScript-Kunstgriffe. Besonders das Erstellen und Verwenden von Formularen im sog. **FreiRaum** ist denkbar einfach. Die Ajax-Technik (dynamische Client-Server-Kommunikation unter der Oberfläche) ist so integriert, daß man sie als solche gar nicht kennen muß. (Beispiele im Menü der Präsentation.)
Mit diesem Ansatz wird sich *allegro* auch in das dynamische Umfeld mobiler Dienste (Android, Tablets, Google Chromebook) mit recht geringem Aufwand einbringen können.

FLEX ist auch das Mittel, wenn man **Web-Services** innerhalb *allegro* nutzen will bzw. wenn eigene Web-Services auf der Grundlage einer allegro-Datenbank zu programmieren sind. Auf diese Weise – allein mit FLEX - wurde z.B. ein OAI-Service geschaffen.

Die Kürze der Zeit erlaubt keine umfassende Darstellung, es wird nur auf einige wichtige Aspekte eingegangen. Die a30-Präsentation ist aber mit Links zu ausführlicheren Darstellungen angereichert.
Nach dieser Einleitung nun ein Blick auf die Grundprinzipien der Entwicklung.

Grundprinzipien

- **Angemessenheit**
  Jede Programmieraufgabe hat viele denkbare Lösungen, jede davon hat ihre eigenen Vor- und Nachteile. Stets muß man den Kontext des Gesamtsystems im Auge haben und die Flexibilität einer Lösung. Unangemessene Ansätze sind dabei viel weniger leicht erkennbar als z.B. beim Schießen mit Kanonen auf Spatzen oder beim Braten eines Schweins mit dem Brennglas.

- **Unabhängigkeit**
  allegro soll quasi alles selber können, d.h. nicht angewiesen sein auf Fremdkomponenten, von deren Weiterentwicklung seitens dritter das System dann abhängig wäre, sowohl in rechtlicher wie in technischer Hinsicht (Probleme bei Versionswechseln!). Das Gesamtpaket für den „Normalanwender“ enthält alles, was gebraucht wird, die Quellcodes enthalten nichts, was Rechten dritter unterliegt. (Siehe auch oben 4.) Unabhängigkeit ist im Grunde aber auch sonst das wichtigste Prinzip: die Punkte 1. bis 5. drehen sich alle um Aspekte der Unabhängigkeit.

- **Tempo**
  Nicht nur schnelle Reaktion ist ein Thema, sondern auch, daß in der Alltagsarbeit die ständig benötigten Vorgänge schnell zu lernen und bequem auszuführen sind. Schnell gehen auch lokale Änderungen und Erweiterungen. Schnelle Hilfe bei Fehlern und Anwendungsproblemen erhält man meistens über das E-Mail-Forum.

- **Offenheit**
  Bis Ende 2012 werden alle noch nicht frei verfügbaren Programmtexte, und das sind jetzt nur noch die in C++ geschriebenen Kernprogramme, unter der Lizenz „Apache2“ veröffentlicht werden. Im Endeffekt bedeutet das die Unabhängigkeit der Anwender von den Entwicklern.

Softwaretyp

allegro-C ist ein **objektorientiertes Datenbanksystem**, kein relationales. Es gibt keine Tabellen und deshalb auch keine SQL-Implementierung. In etwa eine Entsprechung zu SQL ist die Skriptsprache FLEX, deren Funktionsumfang jedoch über SQL weit hinausgeht.

Zum Thema Offenheit

Das Konzept „Open Source“ hat für *allegro* drei Aspekte:

- ***Strukturale Offenheit***
  Darunter fallen die Datenstruktur sowie die Konfigurations- und Parameterdateien. Hier war von Anbeginn alles offen, also lokaler Modifikation zugänglich.

- ***Prozedurale Offenheit***
  Diese bietet die Skriptsprache FLEX, mit der heute alle höheren Funktionen und ganze Anwendungen sowohl im Windows- wie im Web-System geschrieben werden. An den Datenstrukturen und Parametern (darin stecken oft anwenderseitige Investitionen) hat sich durch diese Entwicklung nichts geändert, sie bleiben gültig.

- ***Fundamentale Offenheit***
  Nur die Quelltexte in C++ waren bislang nicht veröffentlicht. Sie bedürfen einiger Vorarbeiten, einer Bereinigung älterer Teile sowie gründlicher und vereinheitlichter Kommentierung, bevor sie freigegeben werden. In der ersten Phase (Mai 2011) zugänglich gemacht wurden die wichtigsten Teile, die sog. „Klassenbibliothek“, worauf die heutigen Programme beruhen. Damit können schon eigene Programme geschrieben werden, die auf *allegro*-Datenbanken lesend und schreibend zugreifen.

Für den Zugang zu allen offenen Materialien wurde eine Versionsverwaltung mit „Subversion“ eingerichtet.

Datenkommunikation

Als angemessene Lösung für den Gesamtbereich der Datenmanipulation, für Export und Import wie auch für andere Aufgaben des Umgangs mit Daten, wurde schon früh ein zweigeteilter Ansatz gewählt. Es gibt zwei Datenmanipulationssprachen:

1. **Konvertieren** von Fremddaten in das eigene Format (Import)

2. **Output produzieren** aus dem eigenen Format (Export)

So ist erreicht, daß Konvertierungen von einem Fremdformat in ein anderes Fremdformat, z.B. MAB2 in MARC21, in einem Durchlauf möglich sind: Man koppelt eine Importparameterdatei MAB2.AIM und eine Exportparameterdatei MARC21.APR hintereinander; die erste wandelt die Fremddaten in das interne Format, das sog. A-Schema, die zweite produziert daraus einen Output im MARC21.
Dieses Verfahren, entstanden 1987, gehört zu den frühen Innovationen. Über einen solchen universellen Lösungsansatz wurde damals auch anderswo diskutiert, konsequent umgesetzt wurde die Idee aber, soweit wir sehen, nur in *allegro*.
Für Anwender ergibt sich nebenbei der Vorteil, ihr internes Format nicht wechseln zu müssen. Der Umstieg von MAB2 zu MARC21 bedeutet nur, daß man zum Importieren von Fremddaten eine andere Parameterdatei einsetzt. (Keine Software kann den Arbeitsaufwand eliminieren, der sich aus einem Wechsel des Regelwerks ergibt, aber das ist ein anderes Thema.)

Suchmaschinensoftware

Seit 2009 wurden drei Wege eröffnet, die *allegro*-typischen Suchmöglichkeiten zu erweitern und damit die traditionell geprägten, heute als umständlich empfundenen Eigenheiten des Bibliothekskatalogs zu überwinden, aber ohne seine Vorzüge aufzugeben. Der zweite und dritte Weg greifen auf externe Software zurück, die aber ebenfalls quelloffen verfügbar ist.

1. **Das ALL-Register (2011)**
   Es wird eine zusätzliche Indexdatei angelegt, in der „alle Wörter“ aus den Datensätzen indexiert werden. Dazu gibt es dann FLEX-Skripte für die Windows- und Web-Oberflächen, die eine Nutzereingabe – typischerweise 2 oder mehr Stichwörter und/oder Namen – mittels UND-Kombination in diesem Register suchen. Bleibt dies ergebnislos, werden die Wörter nochmals mit Trunkierung gesucht, hilft auch dies nicht, werden sie verkürzt.
   Diese Methode wurde schnell von vielen Anwendern aufgegriffen und ist sehr beliebt.

2. **Der externe Index mit Solr von Apache (2010)**
   Die kompletten Titeldaten werden in einer Form exportiert, aus der Solr dann einen Index erzeugen kann. Dieser wird aus der Windows- bzw. Web-Oberfläche heraus genutzt, ohne daß der Endnutzer davon etwas merkt, d.h. der externe Index fügt sich nahtlos in die *allegro*-Oberfläche ein. Möglich wird dann jedoch ein Ausbau mit weiteren Funktionen, wie Phrasensuche, Ranking etc., die zum Repertoire von Solr gehören.

3. **Trennung von OPAC und Datenverwaltung mit VuFind (2009)**
   In den Wogen der Web2.0-Bewegung entstand das OpenSource-OPAC-System VuFind. Jeder kann damit einen Web-OPAC aufsetzen; Voraussetzung ist nur, die eigenen Daten im MARC21-Format exportieren zu können. Das kann *allegro*, und so wurde der Einsatz von VuFind als neue Option möglich.

Scripting

Unter der früheren DOS-Oberfläche gab es separate Module für Ausleihe und Erwerbung. Das waren getrennte Programme mit spezialisierten Aufgaben. Dieser Ansatz erschien nicht mehr angemessen, als mit Windows die engen Grenzen der MS/DOS-Umgebung wegfielen. Eingebaut in das Windows-Hauptprogramm wurde ein Skript-Interpreter, der die für *allegro* entwickelte Skriptsprache namens FLEX interpretieren und ausführen kann. In der Sprache FLEX, die seit 1999 immer mächtiger wurde (unter DOS gab es nichts vergleichbares), konnten in der Folge **alle höheren Funktionen** neu programmiert werden. Für die Web-Oberflächen kommt FLEX gleichfalls zum Einsatz; dies entspricht in etwa dem Verfahren, die Sprache SQL in gängige Skriptsprachen wie Perl und PHP einzubeziehen, um auf relationale Datenbanken zuzugreifen. Die Beispiel- und Sonderfunktionen im Menü der Präsentation beruhen darauf.

Ein schwieriges System?

Das System *allegro-C* gilt als umfangreich und komplex, als schwer zu erlernen und zu beherrschen. Wer die Klippen und Lernschwellen einmal überwunden hat, entwickelt Gespür für seine Logik, schätzt Leistung und Vielseitigkeit und versteht, daß dies alles nicht zu haben wäre ohne Komplexität. Hat man aber eine Anwendung auf der Basis von *allegro* erst einmal fertig konfiguriert, ist der Alltag damit nicht schwieriger als mit anderen Systemen. Komplex ist, wie wir doch alle wissen, zuerst einmal die Bibliotheksarbeit selbst, und *allegro* ist im Dialog mit Praktikern geworden und gewachsen, um dieser Arbeit zu entsprechen. Es sind die realen Anforderungen aus ungezählten, verschiedenartigen, mitunter „exotischen“ Projekten der Praxis, die immer wieder Anstöße gegeben haben zu Innovationen. Diesen fortwährenden Prozeß kann man im Mailforum beobachten. Dort beteiligen sich etliche erfahrene „Allegrologen“ mit großem Engagement nicht nur an Problemlösungen, sondern sie haben auch ungezählte Vorschläge für Neuerungen und Verbesserungen eingebracht. Das alles hat auch sehr geholfen, die Software hinsichtlich Zuverlässigkeit und Sicherheit auf ein hohes Niveau zu bringen, was wiederum allen anderen zugute kommt. Wer ohne lange Lernphase mit allegro arbeiten will, kann komplette Anwendungen von anderen Anwendern übernehmen oder sich von erfahrenen Supportern eine Lösung nach Maß erstellen lassen, samt Installation, Schulung und regelmäßige Betreuung. Dies entspricht dem, was bei OpenSource-Produkten allgemein üblich ist. Für Öffentliche Bibliotheken bietet die Büchereizentrale Niedersachsen (Lüneburg) umfassenden Service.

Gefördert durch das Land Niedersachsen

Das Niedersächsische Ministerium für Wissenschaft und Kultur (MWK) hat die allegro-Entwicklung längerfristig gefördert: Seit 1991 ist diese Arbeit als „zusätzliche Sonder-aufgabe“ der Bibliothek übertragen gewesen. Mit dieser Unterstützung war es möglich, das System kontinuierlich bis zum heutigen Umfang und Potential auszubauen. Die Förderung läuft Ende 2012 aus, dann jedoch soll und wird die Software als Open-Source-Paket der bibliothekarischen Öffentlichkeit auch im Quelltext vollständig zur Verfügung stehen. Die Bibliothek versteht es als Verpflichtung gegenüber der zahlreichen und langjährig zufriedenen Anwenderschaft, auf diese Weise Zukunftssicherheit zu gewährleisten.

Links zu *allegro-C* (alle auch auf der Homepage zu finden)

Homepage:
http://www.allegro-c.de

Anwender
http://www.stepmap.de/karte/allegro-c-nutzer-bunt-148115 (Karte)
http://www.allegro-c.de/ac-dbs.htm (*allegro*-Kataloge im Web)

Download des Demopakets: (einschl. Ausleihe, Erwerbung, Zeitschriften u.a.)
http://www.allegro-c.de/download.htm

Chronik der Entwicklung ab 1980:
http://www.allegro-c.de/chronik

FLEX, die *allegro*-Skriptsprache
http://www.allegro-c.de/doku/flex

OAI-Service mit *allegro*
http://www.allegro-c.de/oai.htm

a30, neue Oberfläche und „Rich Internet Application“:
http://www.allegro-c.de/doku/a30/ ab 2013 a35: http://www.allegro-c.de/doku/a35/

Mailforum mit Archiv:
http://sun250.biblio.etc.tu-bs.de/mailman/listinfo/allegro

SVN-Repositorium für das OpenSource-Angebot:
http://svn.allegro-c.de/svn

Weiterführende Quellen

Rich Internet Applications (RIA) – ein Überblick zum Thema
http://www.allegro-c.de/doku/a30/rias.htm

Solr von Apache (früher *Lucene*)
http://lucene.apache.org/solr/

VuFind („The library OPAC meets Web 2.0”)
http://vufind.org/

Subversion (Versionsverwaltungssystem)
http://subversion.tigris.org

OpenSource Lizenz „Apache 2“
http://www.apache.org/licenses/LICENSE-2.0.html