Ansätze zur Föderation molekulargenetischer Datenbanken

Ulf Leser[1]

Technische Universität Berlin, Fachbereich 13, Lehrstuhl für ‚Computergestützte Informationssysteme', Einsteinufer 17, 10587 Berlin,

Zusammenfassung:

Die Daten, die im Rahmen des ‚Human Genome Project' weltweit in Tausenden von Labors produziert werden, sind größtenteils über WWW Datenbank Interfaces, statische HTML Seiten, FTP Server oder auf CD öffentlich verfügbar. In diesem Beitrag sollen drei Ansätze beschrieben werden, die das Ziel verfolgen, die verschiedenen, hochgradig heterogenen Datenquellen in einem integrierten Informationssystem zusammenzufassen. Ihre Vor- und Nachteile werden anhand des klassischen Fünf - Schichten Modells für föderierte Datenbanken diskutiert.

Einleitung:

Die Erforschung der menschlichen Erbinformation gehört heute zu den weltweit wichtigsten Forschungsgebieten. Potentielle Auswirkungen liegen zum Beispiel im Bereich der verbesserten Diagnose und Therapie von Erbkrankheiten sowie der zielgerichteten Erstellung von pharmazeutischen Wirkstoffen (‚*Drug Design*'). Gegenstand der Forschung ist sowohl das primäre Bestimmen der vererbten Information, codiert in einer Sequenz aus 4 verschiedenen Molekülen in jedem Zellkern jedes heute bekannten Organismus, sowie auch die Erforschung der Mechanismen, mit denen diese Information gelesen und umgesetzt werden, um einen so komplexen Organismus wie zum Beispiel den menschlichen Körper zu formen und am Leben zu erhalten.

Diese Untersuchungen werden weltweit in Tausenden von Labors und Instituten durchgeführt, die zu mindestens teilweise durch das sogenannte ‚*Human Genome Project*' koordiniert werden. Dem Austausch von Daten kommt dabei eine Schlüsselrolle zu: er erspart doppelte Arbeit, ermöglicht Synergieeffekte und bietet die Möglichkeit, eigene Ergebnisse zu verifizieren und gegebenenfalls zu korrigieren.

Zur Speicherung und Verwaltung der erzeugten Daten kommen eine Vielzahl von Datenbanken und Fileformaten zum Einsatz. Zur Zeit sind alleine über das WWW über 200 Informationsquellen erreichbar (NAR 1997). Relationale Datenbanktechnik ist nicht so weit verbreitet wie in anderen Anwendungsgebieten, sondern oft werden proprietäre, speziell entwickelte Systeme verwendet. Der Grund dafür liegt in den Anforderungen der naturwissenschaftlichen Anwendung: hochkomplexe Schema mit häufigen Veränderungen und Erweiterungen, komplexe, non-standard Datentypen, aufgeweichte Integritätsbedingungen, große Datenmengen etc. (Aberer 1994, Letovsky und Berlyn 1994). Eine Reihe von Neuentwicklungen basiert deshalb auf objektorientierten Konzepten.

Diese Besonderheiten müssen bei Ansätzen zur Föderation molekulargenetischer Datenbanken berücksichtigt werden. Als weiterer Unterschied zu beispielsweise betrieblichen Szenerien muß die vollständige

[1] Email: leser@cs.tu-berlin.de

Autonomie aller Datenquellen gewahrt bleiben, was sich für ein globales Informationssystem unangenehm in der spontanen Veränderung von Datenschema bemerkbar macht, auf die es flexibel zu reagieren gilt.

In diesem Beitrag werden drei Ansätze vorgestellt, die der Integration heterogener Datenquellen aus dem Bereich der Molekularbiologie dienen und dazu unterschiedliche Techniken und Architekturen verwenden. Alle Systeme beschäftigen sich ausschließlich mit dem *Information Retrieval* und bieten keine Möglichkeit eines schreibenden Zugriffs auf die integrierten Komponenten. Als Referenzarchitektur wird zunächst das Fünf - Schichten Modell nach (Sheth und Larson 1990) vorgestellt. Abschließend werden die drei Ansätze in ihren Möglichkeiten und Einschränkungen verglichen.

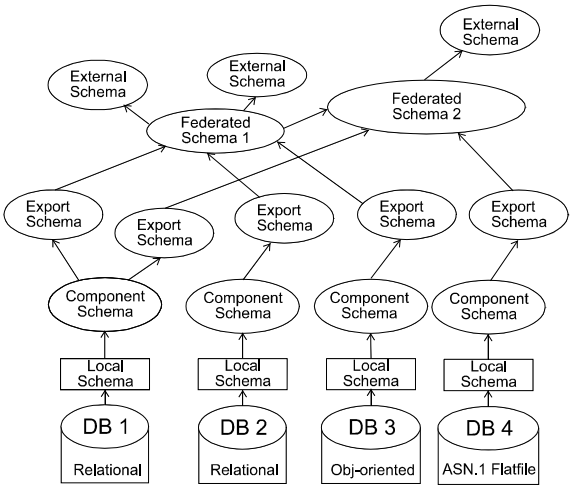

Abbildung 1: 5 Schichten Modell[2]

Die Fünf - Schichten Architektur für föderierte Datenbanksysteme

Abb. 1 stellt ein Fünf - Schichten Modell dar, das sich als allgemeine Referenzarchitektur für föderierte Datenbanksysteme durchgesetzt hat. Die Basiskomponenten, im folgenden Quelldatenbanken genannt, besitzen eine gewisse Überlappung in dem Ausschnitt der Welt, den sie repräsentieren, verwenden für die Repräsentation aber unterschiedliche Datenmodelle und Datenschema. Enthalten im Föderationsgedanken ist die Vorgabe, daß alle beteiligten Quellen ihre Autonomie zum größtmöglichen Grad erhalten, insbesondere also lokale Applikationen nicht beeinflußt werden.

Die Integration der Quellen findet nur virtuell statt und wird i.A. nicht materialisiert. Das ‚Common Data Model‘ (CDM) der Föderation ist das Datenmodell, in dem integrierte Schema ausgedrückt werden. Damit bestimmt es wesentlich die Auswahl einer Anfragesprache auf der Föderationsebene. Es ist naheliegend, ein semantisch mächtiges Datenmodell als CDM zu wählen, da alle Komponentenschema in diesem ausdrückbar sein müssen, um einen Informationsverlust zu vermeiden (Saltor, Castellanos et al. 1991).

Die einzelnen Schichten des Modells bedeuten im Einzelnen:

- ‚*Local Schema*‘: Lokales Schema der Quelldatenbank, ausgedrückt in ihrem jeweiligen Datenmodell.
- ‚*Component Schema*‘: Das lokale Schema, ausgedrückt im CDM der Föderation. Die notwendige Transformation beinhaltet eventuell auch einen Prozeß des ‚Semantic enrichment‘ (Hohenstein und Plesser 1996).
- ‚*Export Schema*‘: Der Teil bzw. die Teile des Komponentenschemas, der der Föderation zur Verfügung gestellt wird. Dient insbesondere zur Realisierung von Zugriffsbeschränkungen.
- ‚*Federated Schema*‘: Integrierte Schema, die eine Menge von Exportschema subsumieren. Die Existenz eines einzigen integrierten Schemas ist keine Bedingung, sondern verschiedene Bereiche können durch verschiedene Schema erfaßt werden (siehe Abb. 1).
- ‚*External Schema*‘: Teile der föderierten Schema, die bestimmten Anwendern bzw. Applikationen zugänglich sind und wiederum insbesondere der Zugangskontrolle dienen.

[2] Der Begriff ‚Objekt-orientiertes Datenmodell‘ wird hier und im folgenden als Oberbegriff über die Vielzahl tatsächlich vorhandener objekt-orientierter Datenmodelle verwendet.

Diese Architektur wird im Folgenden zur Erläuterung der dargestellten Ansätze verwendet.

OPM: ‚Object – Protocol Model‘

Der OPM Ansatz zur Datenbankföderation (Chen und Markowitz 1995) basiert auf der Definition eines objekt-orientierten Datenmodells sowie auf der Bereitstellung einer Menge von Werkzeugen. Mit diesen ist die komfortable Erstellung und das Editieren von Schema sowie die automatische Übersetzung dieser in relationales Schema möglich. Auf dem OPM Datenmodell ist die Anfragesprache OPM-QL definiert, die zur Abarbeitung in SQL Aufrufe übersetzt wird.

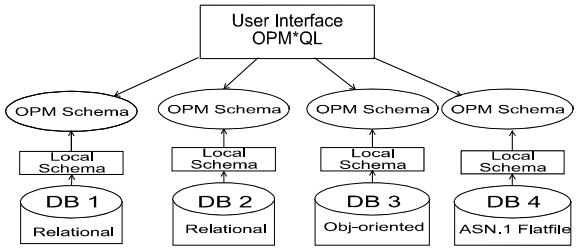

Abbildung 2 : Der OPM Ansatz zur Datenbankföderation

Abb. 2 skizziert die Verwendung von OPM zur Datenbankföderation (Chen, Kosky et al. 1995). Die Quelldatenbanken müssen ein Schema ihrer Daten im OPM Datenmodell zur Verfügung stellen[3]. OPM*QL ist eine Erweiterung von OPM-QL, die eine Qualifikation von Klassennamen mit Datenbanknamen erlaubt. Zur Abarbeitung einer Anfrage wird diese zerlegt und die jeweiligen Teilanfragen an die Quelldatenbanken verteilt. Damit entspricht OPM*QL im wesentlichen der Idee der ‚Multidatabase Languages‘ (Litwin, Mark et al. 1990).

Im Vergleich zum 5-Schichten Modell fällt auf, daß föderierte, also integrierte Schema, gänzlich wegfallen. Komponentenschema müssen das OPM Datenmodell verwenden, womit faktisch eine Beschränkung auf relationale Datenbanken einhergeht, da bisher nur für diese eine automatische Transformation in ein OPM Modell verfügbar ist. Um eine Anfrage stellen zu können, muß ein Benutzer die Schema der einzelnen Quellen genau kennen, insbesondere auch die Zusammenhänge zwischen Schema verschiedener Quellen (zum Beispiel identische Klassen oder Attribute, die als Inter-Datenbank Referenzen benutzt werden können). Dies ist insbesondere schon notwendig, um die Quellen identifizieren zu können, die die gesuchten Informationen enthalten könnten. Um diese Aufgabe zu erleichtern, werden zusätzliche Komponenten in das System eingeführt, die Hilfestellungen geben können: Ein ‚Multidatabase Directory‘ enthält Beschreibungen aller Quellen, eine ‚Schema Library‘ enthält die dokumentierten Schema der Quellen, und eine ‚Link Library‘ verweist auf Entsprechungen in den unterschiedlichen Schema.

BioKleisli

Das System BioKleisli (Davidson, Overton et al. 1997) beruht auf der Sprache CPL (‚Collection Programming Language‘) zur Formulierung von Anfragen und Datenbanktransformationen.

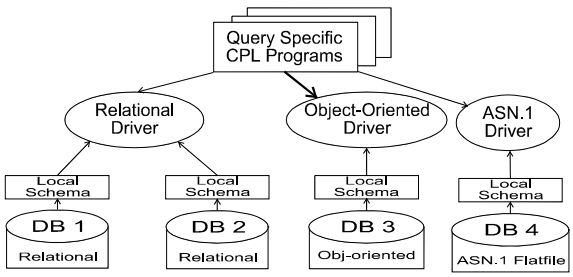

Abbildung 3 : BioKleisli

Zur Abarbeitung von Anfragen greift CPL auf Datenmodell – spezifische Treiber zurück, die CPL Konstrukte in die entsprechenden nativen Anfragesprachen übersetzen. Diese Treiber sind eng in das System integriert und werden eigens entwickelt – die Quelldatenbanken müssen damit keine Eigenleistung erbringen. CPL subsumiert als Anfragesprache beispielsweise SQL und OQL.

Wie im OPM Ansatz gibt es wiederum keine integrierten Schema. Im Unterschied zu OPM verwendet bioKleisli generische Treiber, die pro Datenmodell nur einmal entwickelt werden und dann für andere

[3] Werkzeuge zur Erstellung eines solchen aus relationalen Schema sind vorhanden.

Datenbanken mit demselben Modell wiederverwendet werden können[4].

Da CPL keine Anfragesprache im herkömmlichen Sinne, sondern eher eine komplexe Programmiersprache ist, gibt es für einen Benutzers keinen einfachen Zugang zum System. Anfragen müssen statt dessen in CPL programmiert werden. Dem Programmierer müssen dazu die Quellschema genau bekannt sein (siehe oben). Bestimmte Integrationsroutinen, die einmal auf diese Weise erstellt wurden, können aber in anderen Programmen als Makros wiederverwendet werden.

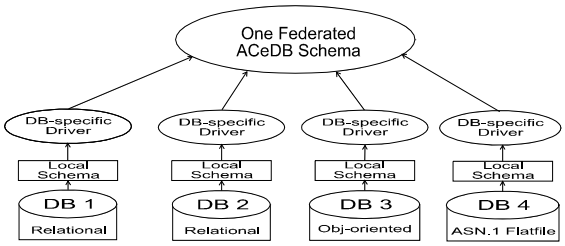

Abbildung 4 : Die 'Integrated Genomic Database'

IGD: Die ‚Integrated Genomic Database'

Das IGD Projekt beschreitet einen grundsätzlich anderen Weg (Ritter, Kocab et al. 1994) als die vorherigen zwei Ansätze. Im Gegensatz zu diesen, die Anfragen grundsätzlich zum Zeitpunkt ihres Auftretens dynamisch beantworten, indem die entsprechenden Daten ‚just-in-time' aus den Quelldatenbanken extrahiert werden, basiert IGD auf dem statischen Kopieren und Integrieren kompletter Datenbankinhalte – erstellt also eine Art *Datawarehouse* für die Molekulargenetik. IGD basiert auf einem globalen, integrierten Schema, das alle Quellschema subsumiert. Das verwendete Datenmodell kann grob als objekt-orientiert minus Vererbung charakterisiert werden. Es basiert auf ACeDB, ein in der Molekulargenetik sehr beliebtes, proprietäre Datenverwaltungs- und Visualisierungssystem (Durbin and Mieg 1991-).

In Abb. 4 wird die Architektur von IGD im Bezug auf den Föderationscharakter dargestellt. Die Quelldatenbanken werden über spezifische Treiber angesprochen, die die Originaldaten sowohl in das ACeDB Modell transformieren als auch im gleichen Schritt die regelbasierte Schemaintegration vornehmen.

Für einen Benutzer bietet dieser Ansatz verschiedene Vorteile. Insbesondere muß er sich nur mit einem Schema vertraut machen, und braucht keine Kenntnis der unterschiedlichen Datenbankdomänen. Anfragen werden nicht auf verschiedene Quellen verteilt, sondern lokal beantwortet, was einen erheblichen Geschwindigkeitsvorteil mit sich bringt. Die Anfragesprache von ACeDB ist allerdings stark auf Navigation ausgerichtet und hat große Schwächen bei mengenorientierten Anfragen (Markowitz und Ritter 1995).

Die Datenbanktreiber basieren auf Flatfile-Dumps der integrierten Quelldatenbanken – wo diese bereits vorhanden sind, muß eine Quelle also keinen technischen Beitrag zu ihrer Integration leisten.

Vergleich der drei Ansätze

Im Vergleich der drei Ansätze zum Fünf-Schichten Modell zeigen sich bestimmte Unterschiede. Exportschema werden prinzipiell nicht betrachtet, da sie klar in den Aufgabenbereich der Quellen fallen und für die Föderation unerheblich sind. Die Transformation der Daten in das CDM muß bei OPM von den Quellen geleistet werden, während IGD und BioKleisli dies innerhalb der Systeme ausführen und damit die maximale Autonomie der Quellen ermöglichen. Bezüglich des Datenmodells ist es auffallend, daß alle drei Ansätze nicht-standard Modelle benutzen, was vermutlich auf die in der Einleitung erwähnten besonderen Anforderungen im naturwissenschaftlichen Bereich zurückzuführen ist.

Eine Schemaintegration, die zu einem großen Teil die Anwenderfreundlichkeit des Systems bestimmt, wird von OPM und BioKleisli nicht durchgeführt, sondern bleibt jedem Benutzer selber überlassen. BioKleisli geht

[4] Abgesehen von spezifischen Dialekten - was erhebliche Einschränkungen bedeuten kann.

diesbezüglich noch weiter, indem es auch keinen einfach zu benutzenden Mechanismus zur Formulierung von Multidatenbankanfragen zur Verfügung stellt, und ist damit eher ein Hilfsmittel zur Integration von Daten als ein tatsächliches integriertes Informationssystem. Eine ausreichende Schema- und Datenbanktransparenz bietet nur IGD. Der Preis dafür ist eine hohe Anfälligkeit gegen Schemaänderungen der Quelldatenbanken, wogegen insbesondere BioKleisli dank der generischen Treiber nahezu immun ist. Der in IGD verfolgte Warehouse Ansatz hat außerdem zwei gravierende Nachteile: die Sicherstellung der Synchronisation der Datenbestände ist nicht trivial, und die schiere Menge der Daten stellen hohe Anforderungen an den lokal verfügbaren Speicherplatz.

Die Möglichkeit zur Definition von externen Schema ist in keinem der drei Systeme gegeben.

Literatur:

Aberer, K. (1994). "The Use of Object-oriented Data Models in Biomolecular Databases". Conf. on Object-Oriented Computing in the Natural Sciences, Heidelberg, Germany.

Chen, A., A. Kosky, et al. (1995). "OPM*QS: The Object-Protocol Model Multidatabase Query System". Berkeley, Lawrence Berkeley National Laboratory. Technical Report LBL-38181.

Chen, I. A. und V. M. Markowitz (1995). ”An Overview of the Object-Protocol Model (OPM) and OPM Data Management Tools.” Information Systems 20(5): 393 - 418.

Davidson, S. B., C. Overton, et al. (1997). ”BioKleisli: a digital library for biomedical researchers.” Int. Journal on Digital Libraries 1: 36 - 53.

Durbin, R. und J. T. Mieg (1991-). "A C. Elegans Database.". Documentation, code and data available from anonymous FTP servers at lirmm.lirmm.fr, cele.mrc-lmb.cam.ac.uk and ncbi.nlm.nih.gov.

Etzold, T., A. Ulyanov, et al. (1996). ”SRS: Information Retrieval System for Molecular Biology Data Banks.” Methods in Enzymology 266: 114 - 128.

Hohenstein, U. und V. Plesser (1996). "Semantic Enrichment: A First Step to provide Database Interoperability". Workshop "Föderierte Datenbanken", Magdeburg, 3 - 17.

Letovsky, S. und M. B. Berlyn (1994). "Issues in the Development of Complex Scientific Databases". The 27th Annual Hawaii International Conference on System Sciences, Hawaii, Vol. 5, 5 - 14.

Litwin, W., L. Mark, et al. (1990). ”Interoperability of Multiple Autonomous Databases.” ACM Computing Survey 22(3): 267 - 293.

Markowitz, V. M. und O. Ritter (1995). ”Characterizing Heterogeneous Molecular Biology Database Systems.” Journal of Computational Biology 2(4): 547 - 556.

NAR (1997). ”Nucleic Acids Research.” January 1997 Issue.

Ritter, O., P. Kocab, et al. (1994). ”Prototype Implementation of the Integrated Genomic Database.” Computers and Biomedical Research 27: 97 - 115.

Saltor, F., M. Castellanos, et al. (1991). ”Suitability of data models as canonical models for federated databases.” ACM SIGMOD Record 20(4): 44 - 48.

Sheth, A. P. und J. Larson (1990). ”Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases.” ACM Computing Survey 22(3): 183-236.