

Datenbankanwendung für veränderbare Fachdatenmodelle im Umweltbereich

Dr. Ulrich Hussels, Stephanos Camarinopoulos, Torsten Lüdtkke, Dr. Georgios
Pampoukis

RISA Sicherheitsanalysen GmbH, risa@risa.de

Abstract

Immer dann, wenn eine fachliche Aufgabenstellung mit Hilfe einer Datenbankanwendung gelöst werden soll und Unsicherheiten hinsichtlich der Struktur, des Umfangs oder auch des Detaillierungsgrads des Datenmodells aktuell oder für die Zukunft besteht, kann eine generische Datenbankanwendung den Aufwand für die Umsetzung und Pflege erheblich reduzieren.

Gleichzeitig erlaubt der beschriebene Ansatz mit unterschiedlichen Sichten auf die Daten zu arbeiten und neue Sichten zu finden.

Die Voraussetzung dafür ist, dass die in der generischen Anwendung umgesetzte Vorinformation zu der fachlichen Aufgabenstellung passt. In diesem Fall können die Teile der Entwicklung der Datenbankanwendung, die mit der generischen Anwendung bereits erfolgt sind, eingespart werden. Dies reduziert die Entwicklungszeit deutlich. Gleichzeitig wird eine Flexibilität hinsichtlich des fachlichen Datenmodells erreicht, die bei konventionellen Entwicklungen nicht gegeben ist.

1 Einführung

Wir stehen als Firma regelmäßig vor der Herausforderung, eine Datenbankanwendung zu einer fachlichen Aufgabenstellung (Fachinformationssystem) zu entwickeln, bei der das Datenmodell im Detail (noch) nicht spezifiziert ist. Gründe für das Fehlen eines detaillierten Datenmodells gibt es viele. Tatsache ist, dass sich auch während und nach der Entwicklung einer Datenbankentwicklung laufend neue Erkenntnisse ergeben, die Einfluss auf das

Datenmodell haben und damit auch eine vorhandene detaillierte Spezifikation verändern würden. Insbesondere bei neuen Aufgabenstellungen resultieren diese Erkenntnisse häufig aus dem Einsatz der Datenbankanwendung selbst, denn ohne jemals mit den Daten gearbeitet zu haben, ist die Spezifikation eines Datenmodells in vielen Fällen kaum möglich. Dies gilt insbesondere dann, wenn vermieden werden soll, dass „Datengräber“ anstelle von schlanken Datenbankanwendungen entstehen. Der Ansatz, möglichst viele Daten zu sammeln, weil sie evtl. irgendwann einmal benötigt werden könnten, ist nicht effizient. Er führt auch nicht zu transparenten und effizienten Datenmodellen.

Eine weitere Anforderung, die sich häufig stellt, ist die, unterschiedliche Betrachtungsweisen auf dieselben Daten zu ermöglichen. Ein starrer Ansatz bietet diese Möglichkeit nicht.

Die Lösung kann in diesen Fällen eine parametrisierte, d. h. generische Datenbankanwendung auf der Basis eines gängigen relationalen Datenbankmanagementsystems (RDBMS) sein. Eine solchermaßen generische Datenbankanwendung kann vom ersten Tag an mit realen Daten arbeiten, bleibt dabei aber veränderbar.

Entscheidend für die Effizienz ist es, die richtigen Eigenschaften der Datenbankanwendung und hier insbesondere die richtigen Eigenschaften des Datenmodells zu parametrisieren. Gleichzeitig muss die Anzahl der Parameter so gering wie möglich gehalten werden, denn weniger Parameter bedeuten mehr Transparenz, mehr Performanz und mehr Wartungsfreundlichkeit. Um mit wenigen Parametern auskommen zu können, ist es unverzichtbar gleichzeitig Erfahrungen im Bereich der generischen Datenbankanwendungen und vertiefte Fachkenntnisse im Bereich der zu lösenden Aufgabenstellung zu besitzen (z. B. [Lüdtke, 2012]).

2 Datenmodell

Da wir seit zwei Jahrzehnten generische Datenbankanwendungen und entsprechende Schnittstellen entwickeln ([Hussels, 2006], [Becker, 2002], [Hussels, 2001]), wissen wir sehr gut, welche Parameter nützlich und welche überwiegend Ballast sind. Es hilft zwar, sich über das Metamodell, welches die Parameter der Anwendung aufnimmt, theoretische Gedanken zu machen, ohne Erfahrungen sind

diese theoretischen Gedanken aber nicht effizient anwendbar. Z. B. betrifft dies die Berücksichtigung der Zeit. Um die Zeit effizient abbilden zu können, müssen erhebliche Vereinfachungen vorgenommen werden. Die für Umweltfragestellungen sinnvollen Vereinfachungen liefern aber nur die Erfahrung und die Kenntnis der fachlichen Fragestellungen. Diese Vorinformationen gilt es in eine parametrisierbare Struktur zu überführen. Der hier beschriebene Ansatz stellt unserer Ansicht nach das Optimum für Umweltdatenbanken dar.

Im Folgenden wird zwischen dem fachlichen Datenmodell, dem Meta-Datenmodell und dem Gesamt-Datenmodell der generischen Datenbankanwendung unterschieden. Das fachliche Datenmodell bildet den fachlich-inhaltlichen Teil der Daten ab und ist konfigurierbar. Im Meta-Datenmodell werden alle Parameter, die zur Interpretation des fachlichen Datenmodells erforderlich sind abgelegt. Im Rest des Datenmodells befinden sich alle festen Strukturen, die darüber hinaus erforderlich sind, wie z. B. die Benutzer- und Rollenverwaltung.

2.1 Fachdatenmodell

Die Basis für die Modellierung des fachlichen Datenmodells bildet bei diesem Ansatz die Betrachtungseinheit. Betrachtungseinheiten werden Gruppen von Individuen genannt, welche mit Hilfe eines gemeinsamen Satzes von Eigenschaften beschrieben werden können. In der Datenbank kann die Betrachtungseinheit dadurch als Tabelle abgebildet werden. Eine typische Betrachtungseinheit im Umweltbereich ist die Messstelle. Die Festlegung der Betrachtungseinheiten richtet sich grundsätzlich nach fachlichen Kriterien und ist daher nicht eindeutig.

Viele Betrachtungseinheiten besitzen Eigenschaften, welche sich nicht in einer einzigen Tabelle abbilden lassen, da die Ausprägungen dieser Eigenschaften in einer 1:n-Beziehung zum Individuum stehen. Schon wenn einige Eigenschaften historisch verwaltet werden müssen, gibt es (zeitlich nacheinander) unterschiedliche Ausprägungen zu einer Eigenschaft. Auch hier können wieder Gruppen von Eigenschaften gebildet werden, die in einer gemeinsamen 1:n-Beziehung zum Individuum stehen. Ein Beispiel hierfür sind historisch verwaltete Stammdaten zu einer Messstelle. Bei der Historisierung von Stammdaten spielt bereits die Abbildung der Zeit im generischen Modell eine Rolle. Jeder Stammdatensatz wird mit einem Gültigkeitsbeginn und einem Gültigkeitsende abgelegt.

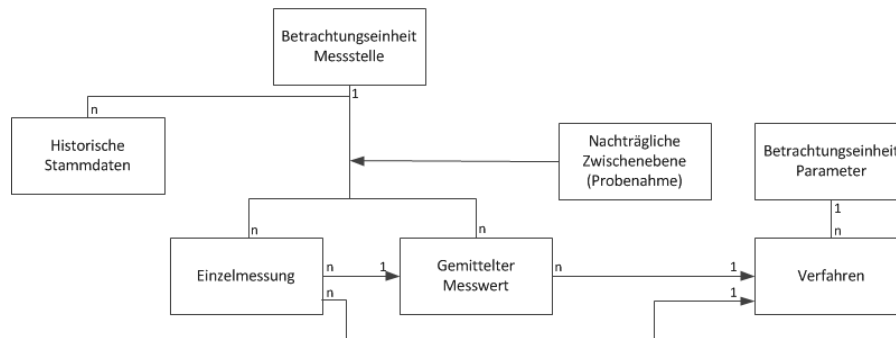


Abbildung 1: Fachdatenobjekt Messstelle mit Relation zum Parameter

Gruppen von Eigenschaften, deren Ausprägungen in einer gemeinsamen 1:n-Beziehung zu einem Individuum einer Betrachtungseinheit stehen, werden als Unterbetrachtungseinheiten bezeichnet. Historisch verwaltete Stammdaten einer Betrachtungseinheit sind folglich bereits eine Unterbetrachtungseinheit. Eine Betrachtungseinheit kann beliebig viele Unterbetrachtungseinheiten besitzen, z. B. auch unterschiedliche Gruppen von Stammdaten mit eigenen Historien. Ein anderer Typ einer Unterbetrachtungseinheit zu einer Messstelle ist der Messwert, wobei zunächst gern vergessen wird, dass zwischen Messstelle und Messwert die Probenahme betrachtet werden sollte. Mit dem vorgestellten Ansatz kann diese Ebene bei Bedarf nachträglich eingeführt werden.

Wird die Überlegung zu den Unterbetrachtungseinheiten systematisch weiter verfolgt, können auch Unterbetrachtungseinheiten ihrerseits Unterbetrachtungseinheiten besitzen. Bei der Unterbetrachtungseinheit Probenahme sind dies die Einzelmessungen bzw. die gemittelten Messwerte zur Probe. Die hierarchische Struktur aus einer Betrachtungseinheit und deren Unterbetrachtungseinheiten wird von uns als Fachdatenobjekt bezeichnet.

Die Eigenschaften der Betrachtungseinheiten besitzen für jedes Individuum, welches durch einen Datensatz in der Tabelle repräsentiert wird, maximal eine Ausprägung. Für diese Ausprägung wird ein Datentyp festgelegt. Neben den Standarddatentypen der gängigen RDBMS verfügt die generische Anwendung über den Datentyp „Zeiger“. Bei einem „Zeiger“ besteht die Ausprägung aus einem Verweis auf einen beliebigen (anderen) Datensatz einer Betrachtungs- oder Unterbetrachtungseinheit. Bei den Messwerten werden z. B. Zeiger auf den Parameter und die Einheit abgelegt, welche eigene Betrachtungseinheiten darstellen. Nur der Zahlenwert ist ein „echtes“ Datum.

Damit liegen bereits die wesentlichen Modellierungsregeln fest. Alle zu verwaltenden Daten müssen als Ausprägungen von Betrachtungs- bzw. Unterbetrachtungseinheiten angegeben werden können. Relationen können über Zeiger innerhalb und über die Grenzen von Fachdatenobjekten hinaus abgelegt werden. Auswahllisten werden über Zeiger auf Betrachtungseinheiten realisiert.

Um der unterschiedlichen inhaltlichen Bedeutung der Betrachtungseinheiten Rechnung zu tragen, werden diese in Gruppen eingeteilt. Übliche Gruppen sind Fachdaten, Kataloge und Auswahllisten, in besonderen Fällen auch gesetzliche Vorgaben.

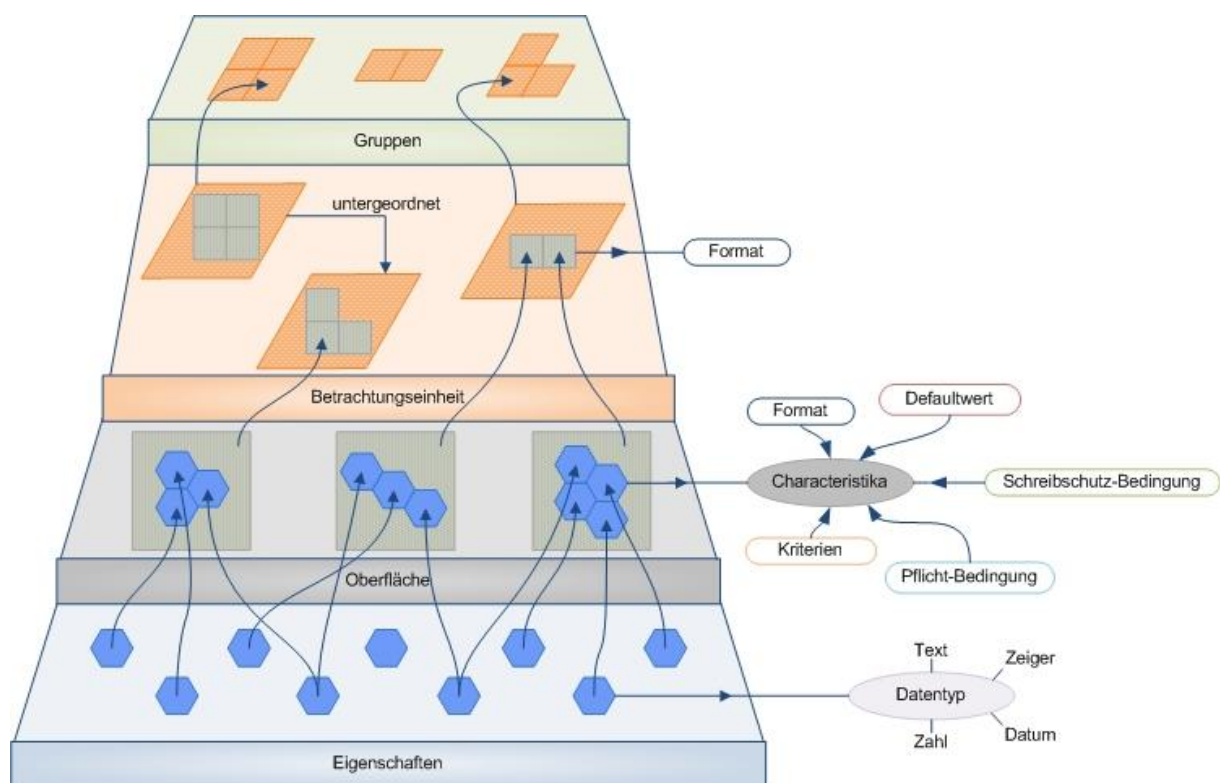


Abbildung 2: Grundsätzlicher Aufbau der generischen Anwendung

Diese Form der Modellierung führt dazu, dass das Tabellenwerk, welches das Fachdatenmodell repräsentiert, absolut transparent ist. Der Tabellename entspricht dem der (Unter-)Betrachtungseinheit. Bis auf die Zeiger sind alle Tabelleninhalte unmittelbar durch den Menschen lesbar.

Diese Eigenschaften des Tabellenwerks erleichtern sowohl die nachträgliche Veränderung des Datenmodells als auch die Datenmigration. Selbst Strukturänderungen (z. B. das nachträgliche Einfügen einer Unterbetrachtungseinheit in ein bestehendes Fachdatenobjekt) ist ohne großen Aufwand möglich. Stellt sich

z. B. nachträglich heraus, dass bestimmte Stammdaten historisch zu verwalten sind, kann diese Unterbetrachtungseinheit in das Fachdatenobjekt eingefügt werden. Ggf. müssen dafür die hierarchisch unmittelbar darunter liegenden Unterbetrachtungseinheiten „umgehängt“ werden. Außerhalb dieses Teils des Datenmodells hat diese Änderung jedoch keine Auswirkungen. Die Datenmigration ist trivial: Für jeden bestehenden Datensatz der Betrachtungseinheit wird ein historisch verwalteter Stammdatensatz mit Gültigkeitsbeginn „offen“ und Gültigkeitsende „offen“ angelegt und an den bestehenden Datensatz angehängt. Anschließend werden die Daten in die Unterbetrachtungseinheit übertragen und die überflüssig gewordenen Eigenschaften (Spalten) der Betrachtungseinheit gelöscht. Zuletzt werden evtl. vorhandene Datensätze der Unterbetrachtungseinheiten, die jetzt unterhalb der historisch verwalteten Stammdaten angesiedelt sind, vom bisherigen Datensatz der Betrachtungseinheit an den neuen Stammdatensatz „umgehängt“. Dieser Vorgang kann völlig automatisch ablaufen.

Die zunächst vorgenommene Einteilung des fachlichen Datenmodells in Fachdatenobjekte stellt nur eine mögliche Sicht auf die Daten dar. Grundsätzlich ist diese Einteilung nicht eindeutig. Für andere Fragestellungen können andere Einteilungen vorgenommen werden. Dies erfolgt, indem die Fachdatenobjekte in einzelne Betrachtungseinheiten „aufgelöst“ werden und eine andere Betrachtungseinheit zum Ausgangspunkt für die Bildung von Fachdatenobjekten gewählt wird. Über das Rückwärtsverfolgen der Zeiger werden so die Unterbetrachtungseinheiten des neuen Fachdatenobjekts ermittelt. Eine Voraussetzung dafür, dass eine Betrachtungseinheit eine Unterbetrachtungseinheit einer anderen Betrachtungseinheit sein kann ist, dass alle Datensätze der potentiellen Unterbetrachtungseinheit einen Zeiger auf die übergeordnete Betrachtungseinheit besitzen. Eine interessante Struktur ergibt sich z. B., wenn anstelle der Messstelle der Parameter als Ausgangspunkt der Bildung von Fachdatenobjekten gewählt wird.

Dadurch, dass sich dieser Vorgang automatisieren lässt, ist es möglich, Fachdatenobjekte durch ein Programm suchen zu lassen und so Sichten zu erzeugen, die bisher nicht betrachtet wurden. Dies ist eine Form der Abfrage, die konventionelle Datenbanken nicht bieten können und die Erkenntnisse bringen kann,

die über eine normale Datenbankabfrage, die mit diesem Modell ebenfalls möglich ist, weit hinausgehen.

2.2 Meta-Datenmodell

Im Meta-Datenmodell wird die Information über den Aufbau des Fachdatenmodells aus Fachdatenobjekten, Betrachtungseinheiten und Eigenschaften sowie deren Datentypen abgelegt. Es enthält ferner die Information über die Interpretation von Zeitangaben. Hierzu können Eigenschaften, die absolute Zeitangaben enthalten, als Beginn und Ende der Gültigkeit eines Datensatzes gekennzeichnet werden. Ferner können absolute Zeitangaben als Ereigniszeitpunkt für den Datensatz gekennzeichnet werden. Eine Betrachtungseinheit kann nur höchstens eine der beiden Angaben enthalten. Auf Basis dieser Metainformationen ist die Verwaltung von Historien (z. B. historisch verwaltete Stammdaten) und von Zeitreihen (z. B. Messungen) generisch implementiert.

2.3 Gesamt-Datenmodell

Neben dem eigentlichen fachlichen Datenmodell und dem dieses beschreibenden Meta-Datenmodell sind Tabellen erforderlich, die die Parameter für die Datenbankanwendung enthalten. Dabei handelt es sich um Tabellen für die Zustands-, Benutzer-, Rollen- und Rechteverwaltung sowie Tabellen für die Verwaltung des Layouts und der Begriffe der Benutzeroberfläche.

3 Zustands-, Status- sowie Rollen- und Rechteverwaltung

Das Konzept sieht vor, dass der gleiche Datensatz in verschiedenen Zuständen betrachtet werden kann. Das bedeutet, dass er in der Datenbank mehrfach in unterschiedlichen Zuständen enthalten sein und dabei jeweils andere Ausprägungen besitzen kann. Diese Eigenschaft kann für unterschiedliche Fragestellungen verwendet werden. Die einfachste ist die Versionenverwaltung, um auf ältere Arbeitsstände zurückgreifen zu können. Um die Zustandsverwaltung jedoch möglichst allgemein verwenden zu können, werden die möglichen Zustände und Zustandsübergänge in einem gerichteten Zustandsgraph abgelegt. Dieser kann z. B. verschiedene Betriebszustände einer Anlage bzw. Umgebungsbedingungen (z. B. Sommer/Winter) oder auch eine Prozesskette abbilden. Beim Login wird

entschieden, von welchem Zustand ausgegangen wird. Bildet der Zustandsgraph eine Prozesskette ab, stellt der Zustand eine Station des Prozesses ab.

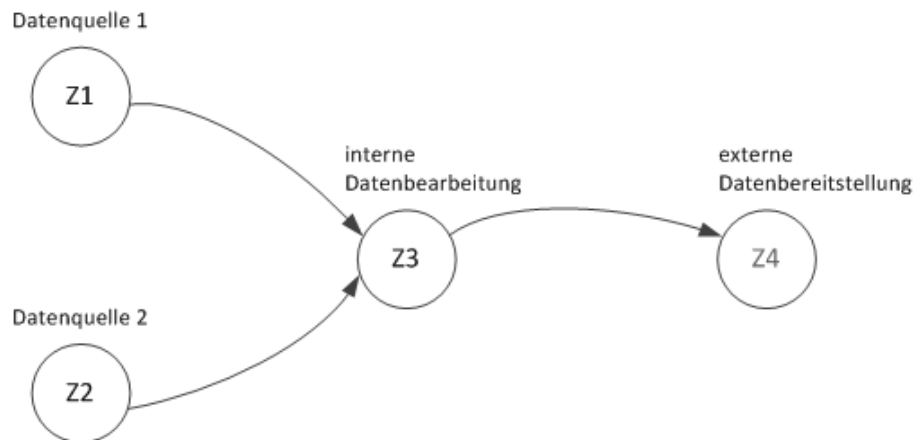


Abbildung 3: Zustandsgraph für eine externe Datenbereitstellung

Ferner sieht das Konzept vor, dass jeder Datensatz (in jedem Zustand) einen Status besitzt. Die Liste der möglichen Status können erweitert werden. Fest vorgesehene Status sind „Aktiv“ und „Gelöscht“, da gelöschte Elemente häufig nicht wieder verwendet werden dürfen und so in der Datenbank verbleiben müssen. Die Löscheinformation kann durch die Statusinformation über Schnittstellen auch zwischen verschiedenen Installationen ausgetauscht werden.

Auch die Rollen- und Rechteverwaltung ist generisch gelöst. Es können beliebig viele Rollen vergeben werden. Die Benutzer werden diesen Rollen zugeordnet.

Auf der anderen Seite werden unabhängig davon Rechte definiert. Die Rechte können einen Parameter besitzen. Der Parameter ermöglicht es, Schreibrechte für bestimmte Ausprägungen einer Eigenschaft zu definieren. Folgende Rechte werden unterschieden:

- Anzahlen ermitteln (Anzahl Treffer einer Abfrage)
- Summenbildung erlaubt (Summe über die Eigenschaften bei einer Abfrage)
- Lesen (einzeln); Ist die Leseberechtigung für eine Tabelle durch eine datensatzabhängige Bedingung definiert, werden dem Benutzer nur die Datensätze angezeigt, die der Bedingung genügen. Besteht eine Leseberechtigung für die Tabelle, jedoch für keines der Attribute, werden leere Datensätze angezeigt. Die Existenz der Datensätze ist damit bekannt.

- Erzeugen (Daten ergänzen, wo bisher keine waren; bei eigenschaftsbezogenen Rechten ist der Parameter der Wert, der erzeugt werden darf)
- Ersetzen (Daten überschreiben; bei eigenschaftsbezogenen Rechten ist der Parameter der zulässige Ersatzwert)
- Löschen (von Datensätzen bzw. bei einzelnen Eigenschaften überschreiben mit NULL)
- Abfragen (über alle Datensätze)
- Exportieren (bezogen auf bestimmte Funktionen und Datenformate)

Die Rechtevergabe kann sich auf die Strukturelemente „Betrachtungseinheit“ (Tabelle) und „Eigenschaft“ (Spalte) beziehen und erfolgt in der Form:

„Wenn <Bedingung erfüllt> dann <Recht> an <Strukturelement>“

Die Bedingung ist eine logische Verknüpfung von Vergleichen, welche nicht nur die Rolle des Benutzers oder den Benutzer selbst, sondern auch Eigenschaften enthalten können. So ist es möglich, auch datensatz- und einzelwertbezogene Rechte zu definieren. Zusammen mit den parametrisierten Rechten ist es so z. B. auch möglich, bestimmten Rollen nur bestimmte Statusübergänge zu erlauben.

4 Datenbankanwendung COODEXX

Die Datenbankanwendung, die das oben beschriebene Konzept umsetzt, nennen wir COODEXX (Configurable ObjectOriented Database Engine with indexEr and XML-Interface).

Das Ziel dieser parametrisierten, d. h. generischen Datenbankanwendung ist es, von Beginn der Entwicklung einer spezifischen Anwendung an mit realen Daten arbeiten zu können. Deshalb besitzt die generische Datenbankanwendung die wesentlichen Funktionen vieler Datenbankanwendungen:

- Datenbrowser,
- Datenformulare,
- Abfragemodul,
- Import- und Exportschnittstellen (xml und tabellenorientiert),

- Druck- und Dokumentationsfunktionen,
- Benutzerverwaltung,
- Rollen und Rechteverwaltung,
- Vorgangsverwaltung.

Das Layout der Benutzeroberfläche wird durch das fachliche Datenmodell und ergänzende Layout-Parameter bestimmt, wobei es eine Grundeinstellung gibt, die es erlaubt, ohne ergänzende Angaben jedes fachliche Datenmodell abzubilden.

Gleichzeitig ist es wichtig, die Begrifflichkeit des Fachgebiets anstelle von Begriffen aus der Informatik zu verwenden. Die Texte der Benutzeroberfläche sind folglich nicht fest, sondern einstellbar. Damit kann die Anwendung auch mehrsprachig eingesetzt werden, wobei dies nicht nur für die Benutzeroberfläche, sondern auch für die Daten gilt. Die Anwendung unterstützt auch die parallele Verwaltung von Daten (beschränkt auf die Textangaben) in mehreren Sprachen, wobei es eine Basissprache geben muss.

Auch wenn die generische Datenbankanwendung schon einen großen Teil der erforderlichen Funktionalität abdeckt, bleiben fast immer Funktionen übrig, die individuell ergänzt werden müssen. Durch das existierende Gerüst einer funktionierenden und mit Daten gefüllten Anwendung ist es jedoch wesentlich einfacher, solche Funktionen zu spezifizieren und umzusetzen als ohne eine solche Umgebung.

Die Datenbankanwendung ist gleichermaßen als Client-Server Anwendung und als Webanwendung einsetzbar. Einzelne Betrachtungseinheiten lassen sich, z. B. zum Zweck der Messwerterfassung, auf mobilen Endgeräten bearbeiten.

5 Grenzen

Die Lösung einer fachlichen Aufgabenstellung mit Hilfe einer generischen Datenbankanwendung kann nur dann erfolgreich sein, wenn sich das fachliche Datenmodell mit den Mitteln des spezifischen generischen Ansatzes effektiv modellieren lässt. Z. B. muss die Modellierung der Zeit für die fachliche Aufgabenstellung ausreichen.

Eine Erweiterung des generischen Ansatzes, um Spezifika eines fachlichen Datenmodells zu implementieren, ist i. d. R. aufwändiger als eine spezifische Lösung.

6 Literaturverzeichnis

[Lüdtke, 2012] Lüdtke, Torsten: Effektive Softwarelösungen für Umwelt- und Anlagendaten an Chemiestandorten. Jahrestagung der NORDOSTCHEMIE, Rheinsberg, 11. Mai 2012

[Hussels, 2006] Hussels, Ulrich: AGXIS – Ein Konzept für eine generische Schnittstellenbeschreibung. Workshop des Arbeitskreises „Umweltdatenbanken“ 6.–7. Juni 2005 in Hannover, Umweltdatenbanken und Netzwerke, Texte 11/06; Dessau, März 2006

[Becker, 2002] Becker, Dierk; Hussels, Ulrich; Nagel, Janet: Vergleich zweier theoretischer Ansätze zur Realisierung generischer Datenbankentwicklungswerkzeuge. In: Simulation in Umwelt- und Geowissenschaften: Workshop Cottbus 2002, Aachen: Shaker Verlag, 2002, S. 137–147, ISBN 3-8322-0733-3

[Hussels, 2001] Hussels, Ulrich; Nagel, Janet: Vorstellung eines generischen Ansatzes zur Erstellung und Pflege von Umweltdatenbanken. In: Wittmann, J.; Bernard, L. (Hrsg.) : Simulation in Umwelt- und Geowissenschaften, Workshop Münster 2001. Aachen, Shaker Verlag, 2001, S. 173–178, ISBN 3-8265-9251-4