

SWE

Zwischenbericht zur Migration von UIS-Centura-Anwendungen nach Java

J. Mäkiö

*Forschungszentrum Informatik
Haid-und-Neu-Str. 10-14
76131 Karlsruhe*

H. Spandl

*Landesanstalt für Umwelt, Messungen und Naturschutz Baden-Württemberg
Griesbachstr. 1
76185 Karlsruhe*

G. Braun von Stumm

*Umweltministerium Baden-Württemberg
Kernerplatz 9
70182 Stuttgart*

1. EINFÜHRUNG	135
2. AUSGANGSSITUATION UND RANDBEDINGUNGEN	135
3. VORGEHENSWEISE BEI DER STUDIE	136
4. MIGRATIONSSTRATEGIEN UND DEREN UMSETZUNG	136
4.1 AUTOMATISCHE CODEUMWANDLUNG	137
4.2 DB-GUI-GENERATOR (PUR).....	137
4.3 STANDARD-JAVA UND WERKZEUGE	138
4.4 DB-GUI-GENERATOR (GEMISCHT)	138
5. ZUSAMMENFASSUNG.....	139
6. LITERATUR.....	139

1. Einführung

Im Informationssystem Wasser, Immissionsschutz, Boden, Abfall, Arbeitsschutz (WIBAS) /1/ des Umweltinformationssystems Baden-Württemberg (UIS BW) sollen in den nächsten Jahren die gegenwärtig eingesetzten, historisch gewachsenen und mit verschiedenen Entwicklungswerkzeugen entwickelten Fachanwendungen für die staatlichen Umweltdienststellen technisch vereinheitlicht werden.

Dieser Bericht stellt Zwischenergebnisse des Projektes „Studie über die Migration von in Centura-Gupta entwickelten Fachanwendungen nach Java“ vor. Die Laufzeit des Projektes reicht vom 05.12.2008 bis zum 31.07.2009. Der Fokus der Studie liegt auf der Analyse der zu migrierenden Fachanwendungen, der Entwicklung von Migrationsstrategien sowie der Untersuchung und Bewertung geeigneter Werkzeuge und deren Anwendung hinsichtlich der Durchführung der Migration unter Berücksichtigung der gegebenen Rahmenbedingungen. Ein wesentliches Ziel der Studie ist eine Entscheidungshilfe für die bevorstehenden Migrationsprojekte.

2. Ausgangssituation und Randbedingungen

In WIBAS gibt es mehrere mit Centura-Gupta entwickelte Fachanwendungen, die nach Java migriert werden sollen. Die Fachanwendungen sind im Wesentlichen Client/Server-Datenbank-Applikationen. Sie ermöglichen die Eingabe, die Pflege und die Einsicht von fachbezogenen Daten, unterstützen zahlreiche Vollzugsaufgaben der Umweltdienststellen sowie die Erstellung verschiedener Reports. Die vorhandene Datenbankstruktur soll von der Migration nicht betroffen sein.

Bedingt durch sich ändernde rechtliche Rahmenbedingungen im Vollzug, zunehmende Berichtsanforderungen von Bund und EU, die zunehmende Vernetzung im Internetzeitalter sowie neue Anwenderanforderungen müssen die Fachanwendungen regelmäßig angepasst werden. Diesen Anforderungen stehen einige technische und nicht-technische Ziele und Randbedingungen bzw. Kriterien gegenüber.

Die Kriterien für die ausgesuchte Lösung können in drei Gruppen unterteilt werden: Randbedingungen („KO-Kriterien“), Kriterien aus Sicht der Entwicklung und des Auftraggebers sowie Kriterien aus Sicht des Endanwenders.

- Die Randbedingungen ergeben sich durch den Einsatz der Fachanwendungen sowie durch strategische Entscheidungen im Vorhaben WIBAS. Die Fachanwendungen sollen in Java bzw. Cadenza /2/ entwickelt werden. Zur Speicherung der Daten wird eine Oracle-Datenbank genutzt. Die Anwendungen werden in einer Client/Server-Umgebung eingesetzt. Die vorhandene UIS-Dienstumgebung einschließlich der UIS-Berechtigungsverwaltung soll beibehalten werden. Die Installation muss in die Standard-UIS-Installation passen.
- Die Kriterien aus Sicht der Entwicklung und des Auftraggebers können u.a. unterteilt werden in *Kosten* (Lizenzmodell und Einarbeitungsaufwand für die Werkzeuge), *Werkzeugeigenschaften* (exogene Kriterien wie Marktstabilität und Verbreitung des Werkzeugs sowie endogene Kriterien wie Integrierbarkeit in den Cadenza-

Anwendungsrahmen und Transparenz hinsichtlich benutzter Drittbibliotheken) und *Programmcode* (Lesbarkeit und Überarbeitbarkeit des erzeugten Codes).

- Die Kriterien aus Sicht des Endanwenders beziehen sich auf benutzerrelevante Eigenschaften der migrierten Fachanwendungen. Dabei spielen z.B. die Performanz der erzeugten Anwendungen sowie die Unterstützung der Arbeitsvorgänge eine wichtige Rolle. Auch Eigenschaften wie Einheitlichkeit der Fachanwendungen sowie Konfigurierbarkeit und Gestaltungsfreiheit der Benutzeroberfläche sind aus der Sicht des Endanwenders von Bedeutung.

3. Vorgehensweise bei der Studie

Die Studie wird in fünf Phasen durchgeführt. Die ersten drei Phasen sind zum Zeitpunkt der Erstellung dieses Dokument bereits abgeschlossen.

1. Phase: Analyse der zu migrierenden Fachanwendungen.
 - Untersuchung der den Anwendungen implizit zugrundeliegenden Workflows, um diese während der Migration möglichst originalgetreu abzubilden.
 - Analyse der in den Fachanwendungen verwendeten GUI-Elemente sowie deren Realisierbarkeit in Java.
 - Aufstellung der zur Bewertung der Migrationsverfahren verwendeten Kriterien.
2. Phase: Entwicklung von Migrationsstrategien sowie Analyse von geeigneten Entwicklungswerkzeugen anhand der Kriterien aus der ersten Phase.
3. Phase: Prototypische Implementierung.
 - Implementierung ausgewählter Teile der Fachanwendungen.
 - Integration in den Cadenza-Anwendungsrahmen.
 - Bewertung der verwendeten Entwicklungswerkzeuge.
4. Phase: Erstellung eines Kriterienkatalogs für Migrationsprojekte, Entwicklung eines Werkzeugs zur Unterstützung bei der Entscheidungsfindung.
5. Phase: Erstellung der schriftlichen Dokumentation der Projektergebnisse.

4. Migrationsstrategien und deren Umsetzung

Die Migrationsstrategien und deren Umsetzung ergeben sich durch die Analyse der Fachanwendungen unter der Berücksichtigung des Bedarfs und übergeordneter Ziele und Kriterien (siehe Abbildung 1).

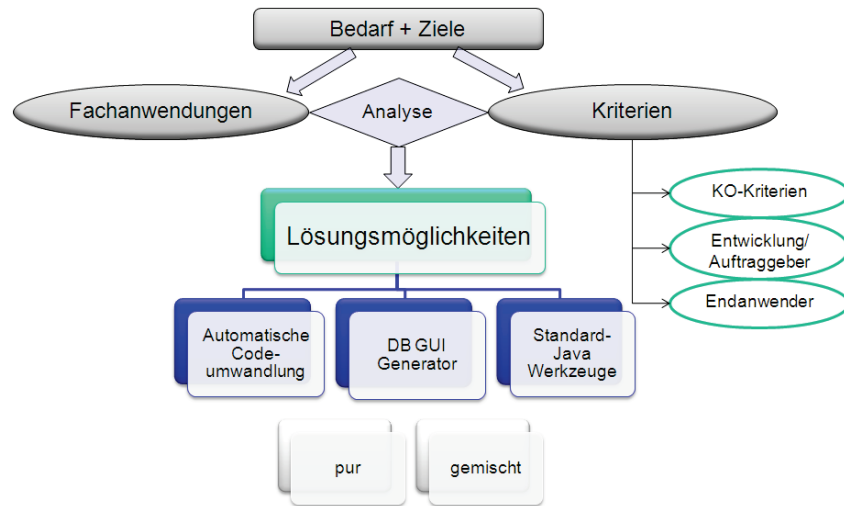


Abbildung 1: Migrationsstrategien und die damit verbundenen Lösungsmöglichkeiten

4.1 Automatische Codeumwandlung

Die Lösungsmöglichkeit „*Automatische Codeumwandlung*“ sieht vor, den vorhandenen Centura-Code in einem ersten Schritt mit einem am Markt erhältlichen Werkzeug automatisch in C#-Code und anschließend mit weiteren frei zugänglichen Werkzeugen in Java umzuwandeln.

Der Vorteil dieser Vorgehensweise ist die automatische Umwandlung des Codes. Dadurch sind erste Ergebnisse schnell verfügbar. Diesem Vorteil stehen jedoch einige Nachteile gegenüber. Centura ist eine prozedurale Programmiersprache. Die Zielsprache der Umwandlung ist eine objektorientierte Sprache, die objektorientiertes Vorgehen in vollem Umfang unterstützt. Die Qualität des erzeugten Codes ist ungewiss, da bei dieser Konstellation zwei vom Grundkonzept her unterschiedliche Programmierparadigmen aufeinandertreffen. Da die Softwarearchitektur des erzeugten Programms vor der Codeumwandlung unbekannt ist, kann der weitere Aufwand, wie z.B. für eine Integration in den Cadenza-Anwendungsrahmen, nicht abgeschätzt werden. Somit sind Aufwand bzw. Kosten dieses Vorgehens kaum kalkulierbar. Daher wurde diese Realisierungsalternative in der Studie nicht weiterverfolgt.

4.2 DB-GUI-Generator (pur)

Bei der Lösungsmöglichkeit „*DB-GUI-Generator (pur)*“ wird die zu migrierende Fachanwendung mit speziellen Werkzeugen für Datenbankanwendungen, die Teile der Anwendung automatisch aus Metadaten erzeugen, nachprogrammiert. Hierbei gibt es zwei Vorgehensweisen. Bei der ersten Variante wird eine graphische Bedienoberfläche, z.B. mittels XML, beschrieben. Anschließend wird zu den Bedienelementen der Bedienoberfläche eine Verbindung zu den einzelnen Spalten aus den betroffenen Datenbanktabellen hergestellt. Bei der zweiten Variante stellen Metainformationen der Datenbank, welche die darin enthaltenen Datenbanktabellen sowie weitere Eigenschaften der Datenbank beschreiben, den Ausgangspunkt dar. Anhand dieser Informationen wird eine graphische Bedienoberfläche erzeugt. Beiden Vorgehensweisen gemeinsam ist, dass die Bedienoberfläche während der Programmlaufzeit durch einen GUI-Generator aus der Beschreibung erzeugt wird.

Die Vorteile dieses Ansatzes sind kurze Entwicklungszeiten, hohe Agilität der Weiterentwicklung, eine weitestgehend standardisierte Bedienoberfläche sowie eine klar vorgegebene Softwarearchitektur. Die Nachteile sind teils mit den verwendeten Werkzeugen, teils allgemein mit diesem Ansatz verbunden. Erstens sind solche Werkzeuge, verglichen mit allgemeinen Programmierwerkzeugen, nicht weit verbreitet. Zweitens sind die Beschreibungen der Bedienoberflächen proprietär und können daher oft nur mit denjenigen Werkzeugen bearbeitet werden, mit denen diese erzeugt wurden. Das kann zu einer Herstellerabhängigkeit führen. Drittens unterstützen diese Werkzeuge nicht alle verfügbaren Bedienelemente einer modernen graphischen Bedienoberfläche. Solche Elemente werden in den meisten der zu migrierenden WIBAS-Applikationen jedoch zentral eingesetzt und sollen nach der Migration auch weiterhin nutzbar sein.

4.3 Standard-Java und Werkzeuge

Bei der Variante „Standard-Java und Werkzeuge“ werden die Fachanwendungen direkt in Java mit Programmierwerkzeugen nachprogrammiert, die von modernen Entwicklungsumgebungen wie Eclipse oder Netbeans in vielfältiger Art und Weise unterstützt werden. Die Bedienoberfläche wird mit Standard-Java-Bausteinen unter Zuhilfenahme von GUI-Werkzeugen erstellt. Für die Kopplung mit der Datenbank werden so genannte Object Relational Mapping (ORM)-Werkzeuge eingesetzt. Diese realisieren eine objektrelationale Abbildung, in der Objekte einer in einer objektorientierten Programmiersprache geschriebenen Anwendung in einer relationalen Datenbank abgelegt werden können. Verglichen mit den beiden ersten Lösungsalternativen sind die ersten Erfolge bei der Entwicklung hier nicht schnell sichtbar. Dies liegt einerseits daran, dass die Softwarearchitektur nicht klar vorgegeben ist und daher erst entworfen werden muss. Andererseits gibt es beim Einsatz von ORM-Werkzeugen keine automatische Zuordnung der in der Datenbank abzulegenden Objekte zu den zugehörigen Datenbanktabellen, was zu Mehraufwand bei der Programmierung führt.

Der Vorteil dieses Ansatzes liegt in seiner Universalität. Die Gestaltung der Anwendungen ist in jeder Hinsicht frei. Für die Migration bedeutet dies, dass die vorhandenen Fachanwendungen originaltreu nachgebaut werden können. Darüber hinaus ist die Entwicklergemeinde sehr groß, was zu einer geringeren Herstellerabhängigkeit führt. Der Nachteil ist, dass dieser Ansatz relativ aufwändig ist.

4.4 DB-GUI-Generator (gemischt)

Die letzte Lösungsmöglichkeit „*DB-GUI-Generator (gemischt)*“ bündelt die beiden zuvor beschriebenen Lösungsmöglichkeiten. Es können Bedienoberflächen für die Teile der Fachanwendungen, die direkt mit den darunterliegenden Datenbanktabellen operieren, mit verhältnismäßig geringem Aufwand erstellt werden. Die Teile der Applikationen, für die mit *DB-GUI-Generator (pur)* hinsichtlich des gewohnten Bedienkomforts keine adäquate Lösung erzeugen lässt, können durch gewöhnliche GUI-Werkzeuge – gegebenenfalls unterstützt durch ORM-Werkzeuge – erzeugt werden. Nachteilig ist, dass in diesem Ansatz zwei unterschiedliche Vorgehensweisen gemischt werden. Das kann zu schlecht wartbarem Code und zu einer Herstellerabhängigkeit führen. Zusätzlich, je nach Anwendung, kann die Bündelung zu einer schlechteren Performanz der Applikation führen.

5. Zusammenfassung

In der vorliegenden Studie werden auf der Grundlage der Kriteriengruppen Randbedingungen („KO-Kriterien“), Kriterien aus der Sicht der Entwicklung und der Auftraggeber sowie Kriterien aus der Sicht des Endanwenders geeignete Migrationslösungen für Centura-Fachanwendungen untersucht und bewertet. Die Studie wird in den fünf Phasen

1. Analyse der zu migrierenden Fachanwendungen,
2. Entwicklung von Migrationsstrategien sowie Analyse von geeigneten Entwicklungswerkzeugen anhand der Kriterien aus der ersten Phase,
3. Prototypische Implementierung,
4. Erstellung eines Kriterienkatalogs für Migrationsprojekte, Entwicklung eines Werkzeugs zur Unterstützung bei der Entscheidungsfindung und
5. Erstellung der schriftlichen Dokumentation der Projektergebnisse

durchgeführt.

Die für die Studie relevanten Migrationsstrategien „Automatische Codeumwandlung“, „DB-GUI-Generatoren“ und Entwicklung mit „Standard-Java und Werkzeugen“ wie GUI-Tools und Object Relational Mapping (ORM) wurden vorgestellt. Da die Studie noch nicht abgeschlossen ist, liegen keine Endergebnisse vor. Absehbar ist, dass außer der automatischen Codeumwandlung alle Varianten zum Einsatz kommen können. Die konkrete Entscheidung bei der Migration einer Fachanwendung hängt von der Gewichtung der einzelnen Kriterien ab. Abhängig von der Schwerpunktsetzung kann die Entscheidung jeweils anders ausfallen. Im restlichen Projektverlauf ist geplant, Entscheidungshilfen für die Migrationsprojekte zu erstellen.

6. Literatur

- /1/ Braun von Stumm, G., Schulz, K.-P., Kaufhold, G., Hrsg. (2006): Konzeption Informationssystem Wasser, Immissionsschutz, Boden, Abfall, Arbeitsschutz (WIBAS) als Teil des ressortübergreifenden Umweltinformationssystems Baden-Württemberg (UIS BW). Konzeption WIBAS 2006. Universitätsverlag Ulm GmbH, ISBN 3-89559-263-3.
- /2/ Tietz, F. et al. (2009): disy Cadenza/GIStern – Weiterentwicklung der Plattform für Berichte, Auswertungen und GIS sowie ihrer Anwendungen bei Partnern. In diesem Bericht.

