Muster zur Migration betrieblicher Informationssysteme (Erfahrungsbericht)

Wilhelm Hasselbring
Universität Oldenburg & OFFIS
se.informatik.uni-oldenburg.de

Achim Büdenbender
IBS AG
www.ibs-ag.de

Stefan Grasmann
Zühlke Engineering GmbH
www.zuehlke.com

Stefan Krieghoff
Kommunale Datenverarbeitung Oldenburg (KDO)
www.kdo.de

Joachim Marz
CeWe Color AG & Co. OHG
www.cewecolor.de

Es gibt verschiedene Möglichkeiten ein Altsystem in eine neue Architektur zu migrieren. Altsysteme stellen wichtige Investitionen dar, die nicht einfach außer Betrieb genommen werden können. Der Betrieb muss während des Übergangs weitergehen. Ein Unternehmen kann nicht – nur zur Einführung einer neuen Software-Architektur – mehrere Monate oder auch Jahre lang seinen Betrieb einstellen. Die Entwicklung eines neuen Systems erfordert einen signifikanten Zeitraum, einschließlich einer ausreichend langen Phase der Stabilisierung des neuen Systems durch den praktischen Einsatz. Während dieses Gesamtzeitraums muss das Altsystem i.d.R. ebenfalls weitergepflegt werden, wodurch zeitgleich Kosten sowohl für das Alt- als auch das Neusystem entstehen. Folglich sind sanfte Migrationspfade und die Integration von Alt- und Neusystemen essenziell für die Praxis der Integration von Informationssystemen [BS95, Has00].

Entwurfsmuster bieten bewährte Lösungsstrukturen für immer wiederkehrende Probleme. Für den objektorientierten Entwurf und auch die Strukturierung auf Softwarearchitekturebene existieren bereits viele Kataloge (siehe z.B. http://www.architekturmuster.de/). Während die Bedeutung von modularen, mehrschichtigen Architekturen für Unternehmensinformationssysteme allgemein akzeptiert ist und deren Vorteile ausführlich publiziert wurden, ist die systematische Migration von Altsystemen (so genannte Legacy-Systeme) hin zu neuen Architekturen nur in einem wesentlich geringeren Maße erforscht. Insbesondere gibt es nur wenig publizierte Muster für diesen Problembereich.

Im Folgenden berichten wir über unsere Erfahrungen aus drei unterschiedlichen Migrationsprojekten, um daraus verallgemeinerte Muster abzuleiten:

KDO: Die Migration kommunaler Informationssysteme von Informix 4GL hin zur Java Enterprise Edition. Die KDO stellt Standardsoftware für kommunale Verwaltungen her, insbesondere auch als Application Service Providing.

IBS: Die Migration von Produktionsmanagementsystemen, die mit dem Gupta Teamdeveloper (mit C++ und Visual Basic) entwickelt wurden, hin zu .NET. Die IBS AG stellt Standardsoftware für die Industrieproduktion her.

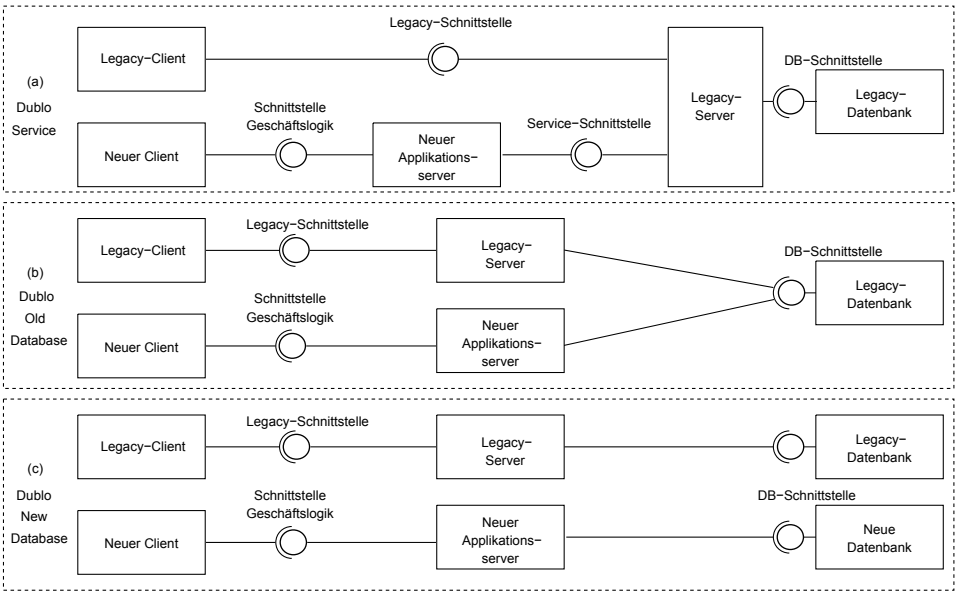

Abbildung 1: Die Varianten des Dublo-Musters.

Cewe Color: Die Migration eines Auspreisungssystems von COBOL hin zur Oracle-Plattform. CeWe Color ist in Europa Marktführer für die Fotoentwicklung, wobei die Auspreisungssoftware die Preisetiketten für die Fototaschen erstellt.

In diesen Projekten traten und treten immer wiederkehrende Probleme auf, insbesondere im Bereich der Datenbankintegration [CHKT05]. Aus den Erfahrungen des KDO-Projektes, welches bereits seit 2002 läuft, wurde schon zuvor das Dublo-Muster [HRJ+04] abgeleitet. Das Dublo-Muster basiert auf der teilweisen Duplikation der Geschäftslogik zwischen Alt- und Neusystem. Inzwischen können wir auf einen — insbesondere durch die Betrachtung mehrerer industrieller Projektkontexte — erheblich erweiterten Erfahrungsschatz zurückgreifen.

Unsere Erfahrungen werden in diesem Papier durch die Beschreibung der Vor- und Nachteile unterschiedlicher Varianten des Dublo-Musters verallgemeinert, so dass sie für ähnliche Aufgaben der Migration von Architekturen wiederverwendet werden können. Wir konnten drei Varianten des Dublo-Musters identifizieren:

Dublo Service Die Dublo-Service-Lösungsstruktur ist in Abbildung 1(a) dargestellt. Die Grundidee besteht in der Entwicklung der Geschäftslogik in der neuen Geschäftslogikschicht, der Erstellung eines Legacy-Adapters für den Zugriff der neuen Geschäftslogik auf die existierende Legacy-Geschäftslogik und der Benutzung dieses Adapters für den Datenzugriff. Folglich wird auf die Datenbank nur indirekt durch den vorhandenen Legacy-Code zugegriffen. Der vorhandene Code dient als dienstbasierte Zugriffsebene für die Datenbank. Auf Funktionalität, die in der neuen Geschäftslogikschicht entwickelt wird, erfolgt der Zugriff durch eine ebenfalls

neue Präsentationsschicht. Dies stellt die Variante des Dublo-Musters dar, wie es in [KHRS08] als Migrationsstrategie hin zu einer Service-orientierten Architektur (SOA) beschrieben wird.

Da die Entwicklung der neuen Präsentations- und Geschäftslogik von der Funktion des alten Systems getrennt wird, ist eine sanfte Migration möglich. Bei dem Dublo-Muster können alte Geschäftslogik und vorhandene Nutzungsschnittstellen so lange wiederverwendet werden, wie sie Funktionalität bereitstellen, die in dem neuen Anwendungskontext sinnvoll ist. Die alte Logik kann Schritt für Schritt durch eine neue Geschäftslogikschicht ersetzt werden.

Dublo Database Old Die Dublo-Database-Old-Lösungstruktur ist in Abbildung 1(b) dargestellt. Die Grundidee besteht darin, dass im Gegensatz zum Dublo-Service-Muster direkt auf die Legacy-Datenbank zugegriffen wird.

Diese Strategie erhält die alte Datenbank und ersetzt die alte Kombination aus Präsentations-, Geschäfts- und Datenzugriffsebene durch getrennte Präsentations- und Geschäftslogikebenen mit dem Vorteil, dass diese Strategie sofort eine Drei-Schichten-Architektur mit den gut getrennten Aspekten Präsentation, Geschäftslogik und Datenzugriff liefert.

Falls das Altsystem ein DBMS verwendet, das den direkten Zugriff erlaubt, bleibt die Möglichkeit des Direktzugriffs auf die Datenbank ohne Verwendung von Legacy-Code, wie es diese Variante vorsieht.

Dublo Database New Die Dublo-Database-New-Lösungstruktur ist in Abbildung 1(c) dargestellt. Die Grundidee besteht darin, parallel zum Altsystem eine ganz neue Infrastruktur aufzubauen.

Die Anwendung des Dublo-Service- und des Dublo-Database-Old-Musters ist sinnvoll, wenn ein inkrementeller Austausch alter Geschäftslogik- und Client-Software durch neue Geschäftslogik in der Mittelschicht angestrebt wird. Da keine zusätzliche Datenbank eingeführt wird, entstehen keine Konsistenz- oder Abgleichsprobleme zwischen neuer und alter Datenbank. Mit dem Dublo-Service-Muster ist es für die neuen Clients transparent ob Geschäftslogik bereits in der neuen Mittelschicht oder noch im alten Legacy-Code implemetiert ist.

Bei der KDO hatten wir uns ursprünglich für die Dublo-Service-Variante entschieden. Der erste Ansatz sah Dublo-Database-Old-Variante vor. Die gewachsenen Datenbankstrukturen in den vorhandenen Altsystemen offenbaren aber nicht die für eine korrekte Benutzung erforderliche Semantik der gespeicherten Datenbankobjekte, welche in den Informix 4GL Funktionen codiert wurde. Inzwischen wurden einige Komponenten auch entsprechend der Die Dublo-Database-New-Variante migriert. Es ist also nicht notwendig, sich in einem Kontext auf eine Variante zu beschränken. Für die jeweiligen Komponenten müssen die entsprechenden Kriterien erfüllt sein (siehe Tabelle 1).

Bei IBS sieht die Situation anders aus. Durch die eingesetzte Implementierungstechnologie (Zugriff auf Oracle und SQLServer aus C++ und Visual Basic Programmen heraus) sind die Datenbankstrukturen nicht direkt mit den oberen Schichten verknüpft, wie es bei

	Einsatzkriterien	Erfolgsfaktoren
Dublo Service	**Pro:** Das Datenbankschema beschreibt die Semantik der Daten nicht ausreichend **Pro:** Ein Parallelbetrieb von Alt- und Neusystem ist erforderlich **Contra:** Aus Performance-Gründen soll direkt auf die Datenbank zugegriffen werden	■ Möglichkeiten zur Generierung von Web Services für Dienste des Altsystems ■ Die Geschäftslogik ist (relativ) stabil ■ Eine serviceorientierte Architektur ist Teil des Leitbilds der IT-Strategie
Dublo Database Old	**Pro:** Geschäftslogik und Datenbank sind gut getrennt **Pro:** Ein Parallelbetrieb von Alt- und Neusystem ist erforderlich **Contra:** Das Datenbankschema beschreibt die Semantik der Daten nicht ausreichend	■ Das Datenbankschema beschreibt die Semantik der Daten ausreichend ■ Das DBMS wird vom Hersteller weitergepflegt oder kann (relativ) einfach ersetzt werden
Dublo Database New	**Pro:** Das Datenbankschema beschreibt die Semantik der Daten nicht ausreichend **Pro:** Das DBMS wird vom Hersteller nicht weitergepflegt **Contra:** Ein Parallelbetrieb von Alt- und Neusystem ist erforderlich	■ Die neue Technologie ist bekannt oder wird umfassend geschult ■ Kein Parallelbetrieb mit alter Datenbank nötig

Tabelle 1: Kriterien für die Varianten des Dublo-Musters.

4GL-Systemen der Fall ist [WW90]. Somit kann hier das Dublo-Database-Old-Muster umgesetzt werden.

Bei CeWe Color besteht der Ansatz darin, die alte COBOL-basierte Datenhaltung komplett abzulösen, somit das Dublo-Database-New-Muster umzusetzen. Dieser Ansatz hat zumeist den Nachteil, Konsistenzmechanismen zu benötigen, die die Daten zwischen alter und neuer Datenbank replizieren [Has97]. Im Falle der zu migrierenden Auspreisungssoftware, die in den Fotolaboren zum Einsatz kommt, werden Alt- und Neu-Systeme jedoch nicht parallel betrieben. Somit kommt dieser Nachteil hier nicht zum Tragen.

Tabelle 1 fasst die Einsatzkriterien und Erfolgsfaktoren für die Varianten des Dublo-Musters zusammen.

Generell gilt für alle Varianten des Dublo-Musters – verglichen mit einem abrupten Übergang der gesamten Geschäftslogik (Big Bang) – dass es in der Übergangszeit zu redundantem Code und Zusatzaufwand kommt. Da neue Systemanforderungen erheblichen Einfluss auf den Entwurf der Geschäftslogik haben können, kann es passieren, dass alter Legacy-Code nicht wiederverwendet werden kann und in der neuen Geschäftslogikschicht sofort reimplementiert werden muss. Dieser letzte Aspekt ist aus vielen Legacy-Integrationsprojekten bekannt. Infolgedessen reduziert dies die Anwendbarkeit des Dublo-Musters auf Bereiche, in denen sich die Geschäftsprozesse während der Übergangszeit nicht zu häufig ändern.

Zusammenfassung Altsysteme stellen wichtige Investitionen dar, die nicht einfach außer Betrieb genommen werden können. Der Betrieb muss während einer Migration weitergehen. Folglich sind sanfte Migrationspfade essenziell für die Praxis der Integration von

Informationssystemen. Es gibt verschiedene Möglichkeiten ein Altsystem in eine neue Architektur zu migrieren. Die Auswahl einer Migrationsstrategie hin zu einer identifizierten Zielarchitektur gehört zu den ersten Schritten in einem Migrationsprojekt. Die in Tabelle 1 zusammengefassten Einsatzkriterien und Erfolgsfaktoren bieten für neue Migrationsprojekte eine Handreichung zur Architekturauswahl, damit Softwareingenieure in ähnlichen Projektkontexten von unseren Erfahrungen profitieren können.

Literatur

[BS95] M.L. Brodie und M. Stonebraker. *Migrating Legacy Systems – Gateways, Interfaces and The Incremental Approach*. Morgan Kaufmann, San Francisco, CA, USA, 1995.

[CHKT05] S. Conrad, W. Hasselbring, A. Koschel und R. Tritsch. *Enterprise Application Integration*. Spektrum Akademischer Verlag, 2005.

[Has97] W. Hasselbring. Federated Integration of Replicated Information within Hospitals. *International Journal on Digital Libraries*, 1(3):192–208, November 1997.

[Has00] W. Hasselbring. Information System Integration. *Communications of the ACM*, 43(6):33–38, 2000.

[HRJ+04] W. Hasselbring, R. Reussner, H. Jaekel, J. Schlegelmilch, T. Teschke und S. Krieghoff. The Dublo Architecture Pattern for Smooth Migration of Business Information Systems. In *Proc. 26th International Conference on Software Engeneering (ICSE 2004)*, Seiten 117–126, Edinburgh, Scotland, UK, Mai 2004. IEEE Computer Society Press.

[KHRS08] S. Krieghoff, W. Hasselbring, R. Reussner und N. Streekmann. Migration von Altsystemen zu serviceorientierten Architekturen. In *Handbuch der Software-Architektur*. dPunkt Verlag, 2.. Auflage, 2008.

[WW90] W.G. Wojtkowski und W. Wojtkowski. *Applications Software Programming With Fourth-Generation Languages*. Wadsworth Publishing, 1990.