



Universität-Gesamthochschule Paderborn

Seminararbeit

**Prototypische Realisierung eines generischen Werkzeugs
zur Migration von Datenstrukturen in
Groupwaredatenbanken**

Prof. Dr. Ludwig Nastansky

Betreuer: Dipl.-Inform. Carsten Huth

Sommersemester 1999

vorgelegt von:

Ingo Erdmann

Zur Warthe 30

33106 Paderborn

Studiengang Wirtschaftsinformatik

Matrikel Nummer: 3269475

Inhaltsverzeichnis

INHALTSVERZEICHNIS	I
ABBILDUNGSVERZEICHNIS	II
TABELLENVERZEICHNIS	II
1. SZENARIO	1
1.1 EINGRENZUNG MIGRATION	1
1.2 EINGRENZUNG FIELD MAPPING	3
2. GRUNDLAGEN MIGRATION	4
2.1 DATENTYPEN.....	4
2.1.1 <i>Basistypen</i>	4
2.1.2 <i>Erweiterte Typen</i>	5
2.2 GRÜNDE FÜR EINE MIGRATION DER DATENSTRUKTUR.....	6
2.2.1 <i>Migrationsprojekte ohne Migration der Datenstruktur</i>	7
2.2.2 <i>Migrationsprojekte mit Migration der Datenstruktur</i>	7
2.3 PHASEN EINER MIGRATION.....	9
2.3.1 <i>Analyse und Planung</i>	9
2.3.2 <i>Durchführung</i>	12
2.3.3 <i>Dokumentation</i>	12
3. ANFORDERUNGEN AN EIN WERKZEUG	14
3.1 ALLGEMEIN	14
3.2 GCC KONTEXT.....	14
3.3 UMSETZUNG IN EINEN PROTOTYPEN	15
4. AUSBLICK	18
5. ZUSAMMENFASSUNG	20
LITERATURVERZEICHNIS	A

Abbildungsverzeichnis

Abb. 1: Ein Dokument besteht aus Note Objekt und Maske.....	2
Abb. 2: Erklärung Field Mapping	3
Abb. 3: Field Mapping im Prototypen	16
Abb. 4: Beispiel für ein Konfigurationsdokument.....	16

Tabellenverzeichnis

Tabelle 1: Datentypen.....	6
----------------------------	---

1. Szenario

In der Vergangenheit wurden am Groupware Competence Center (GCC) der Universität GH Paderborn zahlreiche Projekte durchgeführt, um die Datenstruktur einer Groupwaredatenbank an eine neue Datenbankgestaltung anzupassen. Diese sogenannten Migrationsprojekte wurden manuell durchgeführt, was zu großem zeitlichen Aufwand geführt hat.

In dieser Arbeit sollen die Begrifflichkeiten der Thematik Migration auf Grundlage der Groupwareplattform Lotus Notes eingegrenzt, sowie die Vorgehensweise bei solchen Projekten aufgezeigt werden. Desweiteren ist ein Ziel zu klären, inwieweit die Migration einer Datenbank durch ein generisches Werkzeug unterstützt bzw. durchgeführt werden kann. Grundlage der Arbeit ist im wesentlichen die Erfahrung aus Migrationsprojekten am GCC. Zusätzlich wurde die Hilfe Datenbank der Versionen Lotus Notes 4.6 und 5.0, sowie das IBM Redbook „Lotus Domino R5: A Developer’s Handbook“ verwandt.

1.1 Eingrenzung Migration

Informationen werden in Datenbanken der Groupwareplattform Lotus Notes, im folgenden verkürzt Datenbanken genannt, in Dokumenten gespeichert. Ein Dokument setzt sich grundsätzlich aus zwei Komponenten zusammen, dem Note Objekt und der Maske. Das Note Objekt ist der Datensatz, in dem die Daten eines Dokumentes innerhalb der Datenbank gespeichert werden. Die Maske ist ein Gestaltungselement der Datenbank und bestimmt die Darstellung eines Dokumentes bei der Ausgabe z. B. auf einem Bildschirm oder Drucker. Außerdem stellt sie das Grundgerüst für alle Benutzereingaben zur Verfügung, die entweder durch Eintippen oder durch die Benutzung von Schaltflächen vorgenommen werden können (Eigene Übersetzung aus Collins F. 1999, S. 45). Das bedeutet, eine Maske besteht aus statischen Elementen, Verarbeitungsintelligenz und Feldern. Ein Feld ist ein Element der Maske, das für die Eingabe und die Darstellung von Daten verwendet wird. Das Datum wird nach der Eingabe im Note Objekt in einem Feld gespeichert, das den gleichen Namen trägt wie das Feld in der Maske (Abb. 1).

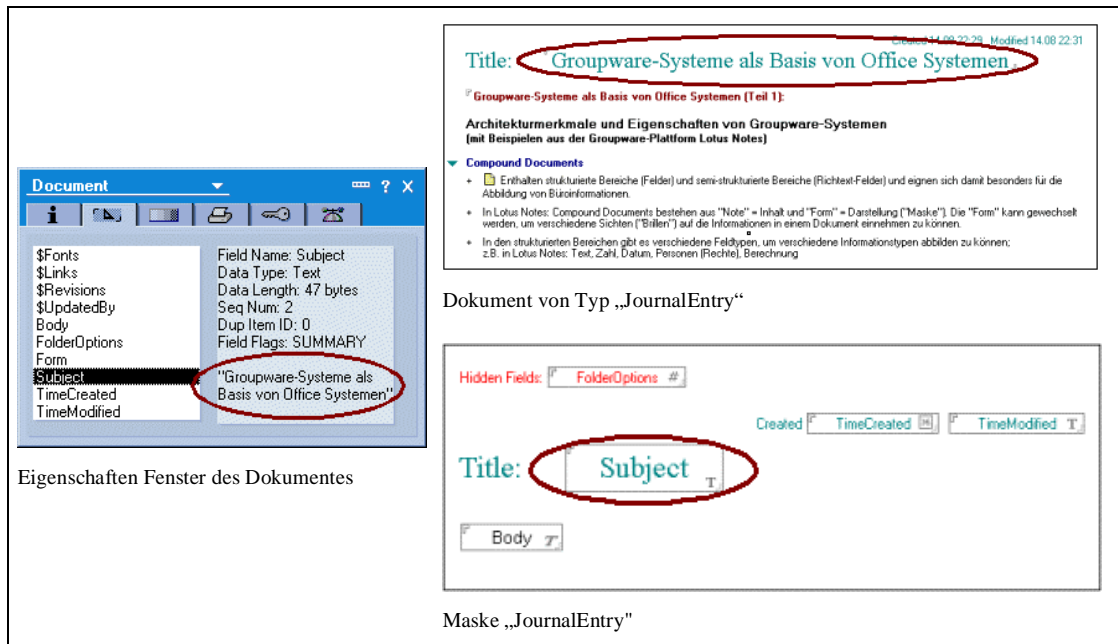


Abb. 1: Ein Dokument besteht aus Note Objekt und Maske

Der Begriff Migration wird für einen Prozeß verwendet, der den Kontext eines Dokumentes konserviert, wenn sich die Feldnamen der Maske ändern. Dabei besteht der Prozeß aus zwei Teilprozessen, der Migration der Datenstruktur, also der Veränderung der Struktur des Note Objektes, und der Migration der Datenbankgestaltung, die das Verändern oder Ersetzen der Maske beinhaltet. Die Migration der Datenbankgestaltung findet in Lotus Notes basierten Groupwareumgebungen durch das Ersetzen der Datenbankgestaltung oder durch direkte Veränderung der Gestaltungselemente statt. Beim Ersetzen der Gestaltung können alle Elemente der Gestaltung, u. a. also auch die Masken, ersetzt werden.

Wenn sich bei einer Migration der Gestaltung einer Datenbank Änderungen an den Feldnamen einer Maske ergeben, ist zur Konservierung des Kontextes der Dokumente vor dem Austausch bzw. der Veränderung der Maske eine Migration der Datenstruktur notwendig. Das bedeutet, daß die Inhalte der Felder in entsprechende neue Felder übertragen werden, deren Namen mit der neuen Maske korrespondieren.

Die Maske, die zur Darstellung eines Dokumentes vor der Migration verwendet wird, wird im Kontext dieser Arbeit als Quellmaske bezeichnet, die entsprechende Maske nach der Migration wird als Zielmaske bezeichnet. Ein Feld im Note Objekt, das den Namen eines Feldes der Quellmaske trägt, wird als Quellfeld bezeichnet. Das Feld, das in der Zielmaske im gleichen Kontext steht wie das Quellfeld wird als Zielfeld bezeichnet.

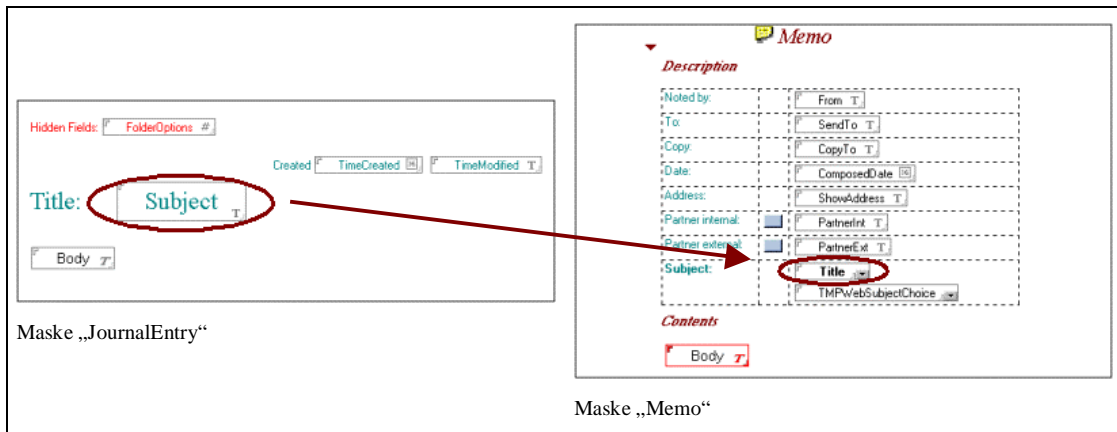


Abb. 2: Erklärung Field Mapping

Abb. 2 zeigt als Quellmaske die Maske mit dem Namen „JournalEntry“ und als Zielmaske die Maske mit dem Namen „Memo“. Ein Quellfeld ist das Feld mit dem Namen „Subject“, ein Zielfeld ist das Feld mit dem Namen „Title“. Die Maske, mit der ein Dokument erstellt wurde, legt den Typ des Dokumentes fest. Ein Dokument vom Typ „JournalEntry“ wurde also unter Verwendung der Maske mit dem Namen „JournalEntry“ erstellt. Die Migration eines Dokumentes vom Typ Quellmaske in ein Dokument vom Typ Zielmaske wird also durchgeführt, indem alle Quellfelder in ihre entsprechenden Zielfelder umgewandelt werden und dem Dokument die Zielmaske zugeordnet wird.

1.2 Eingrenzung Field Mapping

In Vorbereitung auf die Migration der Datenstruktur muß festgelegt werden, welche Quellfelder von ihrer Bedeutung her welchen Zielfeldern am besten entsprechen. Diese Zuordnung von Feldnamen wird als Field Mapping (Mapping) bezeichnet. In Abb. 2 wird das Feld „Subject“ auf das Feld „Title“ abgebildet, es handelt sich also um eine 1:1 Beziehung zwischen Quell- und Zielfeld. Es ist jedoch auch möglich, mehr als ein Quellfeld auf mehr als ein Zielfeld abzubilden, z. B. wenn der Inhalt mehrerer Felder zu einem Feld zusammengeführt werden soll. Daher kann es sich beim Field Mapping um eine n:m Beziehung zwischen Quell- und Zielfeldern handeln, mit $n, m \in \mathbb{N} \geq 0$. Wenn n oder m dem Wert 0 entsprechen liegt ein Sonderfall vor. Ist n gleich 0, existiert also kein Quellfeld, so muß für das Zielfeld ein sinnvoller Vorgabewert gewählt werden. Ist m gleich 0, existiert also kein Zielfeld, ist zu entscheiden, ob das Quellfeld evtl. gelöscht werden kann.

2. Grundlagen Migration

2.1 Datentypen

Jedes Feld in einem Dokument hat einen bestimmten Datentyp. Da i. d. R. ein Dokument erstellt wird, indem in die Felder einer Maske Daten eingetragen werden, wird der Datentyp der Felder des Note Objektes vom Datentyp des Feldes der Maske festgelegt, das den gleichen Namen trägt, so daß deren Datentypen übereinstimmen. Wenn das Feld des Note Objektes einen anderen Datentyp aufweist als das entsprechende Feld der Maske, kann es zu Fehlern bei der Darstellung des Dokumentes kommen. Felder in Masken können nur Daten der Basisdatentypen aufnehmen, daher können sich Felder der erweiterten Typen nicht in einer Maske befinden, d. h. sie kommen nur in Note Objekten vor und können somit nicht in Dokumenten dargestellt werden. Im folgenden wird für Felder der Note Objekte der Begriff „Feld“ und für das entsprechende Feld der Maske der Begriff „Maskenfeld“ verwendet.

2.1.1 Basistypen

Basisdatentypen sind die Typen Text, Zahl, Zeit und Richtext. Ein Feld vom Datentyp Text wird im folgenden als Textfeld bezeichnet. Analog wird mit Feldern der übrigen Datentypen, sowie mit Maskenfeldern, verfahren. Da Maskenfelder nur Daten der Basisdatentypen enthalten können, sind alle Daten, die vom Benutzer eingegeben werden vom Typ eines Basisdatentyps. Ein weiteres Kennzeichen eines Basisdatentyps ist, daß es in vielen Fällen möglich ist, ein Datum eines Basisdatentyps in ein Datum eines anderen Basisdatentyps umzuwandeln. Die Zahl 6 kann z. B. in den Text „6“ umgewandelt werden oder der Text „12.08.1999“ in das Datum 12.08.1999. Richtext bildet eine Sonderstellung, da ein Richtextfeld auch nicht konvertierbare Daten wie z. B. Graphiken, Videos oder Tonaufzeichnungen enthalten kann. Es ist aber möglich, alle anderen Basisdatentypen in Text umzuwandeln und danach die Konvertierung in Richtext vorzunehmen, da Text in Richtext konvertierbar ist. Ebenso ist es möglich, die Textkomponenten aus einem Richtextfeld in ein Datum der anderen Basistypen umzuwandeln. Daher gehören alle Basisdatentypen zur Klasse der konvertierbaren Datentypen.

Der Datentyp Schlüsselwort, der bei Maskenfeldern möglich ist, existiert nicht für Felder in Note Objekten. Maskenfelder vom Datentyp Schlüsselwort, sowie weitere Unterteilungen dieses Typs in Lotus Notes R5, werden in Textfeldern gespeichert.

Der Datentyp Text kann weiter unterteilt werden. Neben dem Standardtyp existiert der Datentyp Namen, der wiederum zusätzlich in den Sonderformen Autoren und Leser auftreten kann. Felder dieser Typen sind Bestandteil der Lotus Notes Sicherheitsarchitektur und regeln den Zugriff auf Dokumente ergänzend zur Zugriffskontrollliste (ZKL) der Datenbank.

Alle Basisdatentypen mit Ausnahme von Richtextfeldern können als Mehrfachwertfelder auftreten, d. h. es können mehrere Werte in einem Feld abgespeichert werden. Diese Felder werden als Listenfelder bezeichnet.

2.1.2 Erweiterte Typen

Erweiterte Datentypen sind zur internen Verwendung in der Programmierung vorgesehen und treten nicht in Maskenfeldern auf, daher können die Inhalte dieser Felder nicht in Dokumenten dargestellt werden. Im folgenden werden diese Datentypen mit dem Namen der zugehörigen Lotus Script Konstante bezeichnet, da keine Deutsche Übersetzung für diese Datentypen existiert. Die Bedeutung der einzelnen Typen wird kurz in Tab. 1 erläutert. Erweiterte Datentypen kommen im Allgemeinen nicht als Liste vor. Einzige Ausnahme bildet der Typ NOTELINKS, der verwendet wird, um Dokumentenverknüpfungen in Richtextfeldern zu verwalten.

Daher kann bei einer Migration davon ausgegangen werden, daß es sich beim Field Mapping von erweiterten Typen um eine 1:n Beziehung handelt, da die Zusammenführung dieser Felder nicht möglich bzw. nicht sinnvoll ist. Dies gilt auch für den Typ NOTELINKS, da Felder dieses Typs automatisch vom Notes Client erzeugt werden und eine Erweiterung der Liste in einem Feld zu Störungen der Inhalte der Richtextfelder bei der Darstellung des Dokumentes führen kann, in denen sich die entsprechenden Verknüpfungen befinden.

Ausnahmen stellen die erweiterten Datentypen HTML, MIME_PART und FORMULA dar. Bei diesen Typen ist es zwar nicht sinnvoll, die Zusammenführung in Form einer Liste durchzuführen, aber es kann durchaus Sinn machen, eine bestehende Formel zu erweitern oder an den Programmcode eines HTML Feldes

etwas anzuhängen. Daher zählen diese Typen zusätzlich zu den Basistypen zur Klasse der konvertierbaren Datentypen, denn ihr Inhalt lässt sich in Text umwandeln.

Datentyp (Name der Konstante)	#Typ	Erklärung	Konvertierung	Liste möglich	Klasse
TEXT	1280	Text	Ja	Ja	Basis
NAMES	1074	Kanonische Namen	Ja	Ja	Basis
READERS	1075	Leser eines Dokumentes	Ja	Ja	Basis
AUTHORS	1076	Autoren eines Dokumentes	Ja	Ja	Basis
NUMBERS	768	Zahlen	Ja	Ja	Basis
DATETIMES	1024	Datums- und Zeitwerte oder Zeiträume	Ja	Ja	Basis
RICHTEXT	1	Reichtext	Ja	Nein	Basis
HTML	21	HTML Programmcode	Ja*	Nein	Erweitert
MIME_PART	25	Feld mit MIME Inhalten (Neu in Notes R5)	Ja	Nein	Erweitert
FORMULA	1536	Notes Formel	Ja*	Nein	Erweitert
ATTACHMENT	1084	Datei Anhang	Nein	Nein	Erweitert
EMBEDDEDOBJECT	1090	Eingebettetes OLE Objekt	Nein	Nein	Erweitert
OTHEROBJECT	1085	Sonstiges Objekt	Nein	Nein	Erweitert
ERRORITEM	256	Fehlerfeld, tritt bei Zuweisung inkompatibler Datentypen auf	Nein	Nein	Erweitert
USERID	1792	Name einer Benutzer ID	Nein	Nein	Erweitert
NOTEREFS	4	Verknüpfung eines Antwortdokumentes zum Hauptdokument	Nein	Nein	Erweitert
ICON	6	Icon	Nein	Nein	Erweitert
NOTELINKS	7	Verknüpfung (Datenbank, Ansicht, Dokument, Anker)	Nein	Ja	Erweitert
SIGNATURE	8	Kryptographische Signatur	Nein	Nein	Erweitert
UNAVAILABLE	512	Nicht verfügbarer Datentyp	Nein	Nein	Erweitert
UNKNOWN	0	Unbekannter Datentyp	Nein	Nein	Erweitert
USERDATA	14	User data	Nein	Nein	Erweitert
QUERYCD	15	Notes Query	Nein	Nein	Erweitert
ACTIONCD	16	Aktion	Nein	Nein	Erweitert
ASSISTANTINFO	17	Assistant Information	Nein	Nein	Erweitert
VIEWMAPDATA	18	Gespeicherter View Map Datensatz	Nein	Nein	Erweitert
VIEWMAPLAYOUT	19	Gespeichertes View Map Layout	Nein	Nein	Erweitert
LSOBJECT	20	Lotus Script Programmcode	Nein	Nein	Erweitert

* nur als Quelle einer Konvertierung. Dieser Typ kann nicht Ziel einer Konvertierung sein.

Tabelle 1: Datentypen (Eigene Übersetzung und Ergänzung aus Lotus Development Corporation, 1999, NotesItem Klasse, Type property)

2.2 Gründe für eine Migration der Datenstruktur

Da die Migration der Datenstruktur einer Datenbank ein aufwendiges Projekt sein kann, sollte im Vorfeld geklärt werden, ob dies wirklich notwendig ist, oder ob das gewünschte Ergebnis nicht auch durch die Anpassung der Zielmaske unter Beibehaltung der Namen der Maskenfelder erreicht werden kann. Die Gründe für eine Migration können daher in zwei Klassen unterteilt werden, die maßgeblich den Aufwand der Migration bestimmen, abhängig davon, ob die Migration mit oder ohne Anpassung der Datenstruktur auskommt. Die im folgenden aufgeführten Beispiele stellen nur eine kleine Auswahl der möglichen Gründe dar und sollen die Erfordernisse an ein Migrationsprojekt klassifizierend veranschaulichen.

2.2.1 Migrationsprojekte ohne Migration der Datenstruktur

Abhängig vom Grund für eine Migration, kommen einige Projekte ohne Anpassung der Datenstruktur aus. Wenn dies nicht notwendig ist, reduziert sich der Aufwand der Migration auf den Austausch bzw. die Anpassung der Gestaltung der Datenbank, insbesondere der Masken. Daher sollte versucht werden, die Notwendigkeit einer Anpassung der Datenstruktur zu vermeiden.

2.2.1.1 *Veränderte Anforderungen an eine Anwendung*

Die Anforderungen an eine Applikation unterliegen ständigen Änderungen. In den meisten Fällen steigen die Anforderungen, so daß neue Maskenfelder hinzugefügt oder die Eigenschaften bestehender Maskenfelder verändert werden müssen. Dies ist z. B. der Fall, wenn ein Textfeld in ein Schlüsselwortfeld umgewandelt werden soll, um die Funktionalitäten der Maske zu optimieren. In diesem Fall ist keine Änderung der Datenstruktur nötig, da sich weder Datentyp des Feldes noch Name des Maskenfeldes ändern. Auch das Hinzufügen neuer Maskenfelder ist nicht mit einer notwendigen Anpassung der Datenstruktur verbunden, da es in der Lotus Notes Architektur ohne weiteres möglich ist, einem Note Objekt weitere Felder hinzuzufügen.

2.2.1.2 *Neuer Produktzyklus*

Die Migration einer Datenbank, deren Gestaltung auf einem kommerziellen Produkt basiert, wird dann notwendig, wenn eine neue Version der Applikation verwandt werden soll, z. B. um Erweiterungen des Funktionsumfangs nutzen, oder Fehlerbereinigungen einbringen zu können. In einem solchen Fall treffen auf die Umstellung auf die Gestaltung der neueren Version der Applikation die gleichen Aussagen zu, die in Punkt 2.2.1.1 getroffen wurden, mit dem Unterschied, daß der Anwender nicht entscheiden kann, ob eine Migration der Datenstruktur vermieden werden soll oder nicht, da diese Entscheidung vom Hersteller getroffen wird. Daher sollte darauf geachtet werden, daß der Hersteller bei der Auslieferung einer neuen Version, sofern die Migration der Datenstruktur notwendig ist, die benötigten Werkzeuge bereitstellt, um die Migration automatisiert durchzuführen.

2.2.2 Migrationsprojekte mit Migration der Datenstruktur

Der Aufwand für Migrationen, bei denen eine Anpassung der Datenstruktur notwendig ist, kann sehr hoch sein. Daher ist ein Ziel dieser Arbeit, diese Art von

Migrationen bestmöglich durch ein generisches Werkzeug zu unterstützen. Änderungen an der Datenstruktur werden u. a. erforderlich, wenn sich der Name eines Maskenfeldes ändert, also das Quellfeld und das Zielfeld unterschiedliche Namen tragen oder Felder, deren Inhalt erhalten bleiben soll, aus der Maske entfernt werden.

2.2.2.1 Wechsel des Produkts

Veränderte Anforderungen, die Einstellung einer Produktlinie oder strategische Entscheidungen gegen den Hersteller der aktuell eingesetzten Applikation können ein Migrationsprojekt erforderlich machen, bei dem die Datenstruktur an das Datenmodell einer neuen Applikation angepaßt werden muß. Da Applikationen verschiedener Hersteller oft eine komplett unterschiedliche Datenstruktur aufweisen, kann in einem solchen Fall der Aufwand für die Migration der Datenstruktur und die Erarbeitung des damit verbundenen Field Mappings sehr groß sein.

2.2.2.2 Kompatibilität zum Standard

Bisher gab es für die Applikationsentwicklung auf Basis der Groupware Plattform Lotus Notes keinen Standard, auf den eine Applikation aufsetzen konnte. Mit der Vorstellung der Lotus Solution Architecture (LSA), einem Datenbankframework zur Lösung von Basisproblemen der Applikationsentwicklung, führte Lotus Development einen Standard ein, auf dem Applikationen basieren können. Um eine Applikation kompatibel zur LSA oder einem anderen Standard zu machen, muß eine Anpassung der vorhandenen Datenstruktur stattfinden und somit eine Migration der bestehenden Datenbanken durchgeführt werden.

2.2.2.3 Zusammenführung von Dokumentenbeständen

Wenn es in einem Unternehmen zu Restrukturierungsmaßnahmen kommt, in Folge dessen Abteilungen zusammengelegt oder aufgeteilt werden, oder Akquisitionen getätigt werden, und die zusammengeführten Abteilungen oder Unternehmen über eigene Datenbestände verfügen, die nach der Zusammenführung als gemeinsame Informationsbasis fortgeführt werden sollen, muß eine der Datenstrukturen angepaßt und die Datenbanken migriert werden.

2.3 Phasen einer Migration

Dieser Abschnitt bezieht sich auf Migrationsprojekte mit Migration der Datenstruktur. Die Migration einer Datenbank ohne Anpassung der Datenstruktur reduziert sich auf das Anpassen oder Ersetzen der Datenbankgestaltung. Zur Reduzierung der Komplexität der Ausführungen wird im folgenden davon ausgegangen, daß nur die Migration eines Dokumententyps durchgeführt wird. Dies kann analog auf mehrere Dokumententypen übertragen werden.

Da bei der Migration Dokumente verändert werden, müssen Projekte dieser Art sehr gut vorbereitet werden, um Datenverlusten vorzubeugen und die operative Umgebung nicht zu gefährden. Daher sollten die Phasen nach Möglichkeit in der hier aufgeführten Reihenfolge durchgeführt werden.

2.3.1 Analyse und Planung

2.3.1.1 *Analyse der vorhandenen Daten*

Bevor mit der Erarbeitung des Field Mappings begonnen werden kann, muß die Struktur der vorhandenen Daten bekannt sein. Wenn diese nicht dokumentiert ist, muß eine Analyse der vorhandenen Daten vorgenommen werden. Ein erster Schritt ist dabei die Analyse der Quellmaske des zu migrierenden Dokumententyps. Die Namen der Maskenfelder und die zugehörigen Datentypen müssen festgestellt werden. Da die Felder der Note Objekte jedoch Datentypen enthalten können, die nicht dem Typ des Maskenfeldes entsprechen, und da auch Felder vorhanden sein können, die nicht in der Maske enthalten sind wie z. B. das Feld mit dem Namen „Form“, sollte auch die Struktur der tatsächlich vorhandenen Dokumente vor der Migration überprüft werden. Da die manuelle Überprüfung aller Dokumente zu aufwendig wäre, sollte dies mit Hilfe eines Werkzeugs geschehen. Wenn für diesen Zweck kein Standardwerkzeug zur Verfügung steht, kann die Analyse der tatsächlich vorhandenen Daten auch auf die manuelle Analyse einzelner ausgewählter Dokumente reduziert werden. Dabei sollte die Tatsache besondere Berücksichtigung finden, daß ein leeres Feld immer vom Typ Text ist, unabhängig davon, welchen Datentyps das zugehörige Maskenfeld ist.

2.3.1.2 *Analyse der Zielmaske*

Die Zielmaske, die das Dokument nach der Migration bei der Ausgabe darstellt bestimmt die Datentypen der Zielfelder. Die Zielfelder müssen jedoch nicht nur auf

ihren Datentyp hin überprüft werden, sondern es muß zusätzlich festgestellt werden, ob für ein Feld evtl. ein Vorgabewert gesetzt werden muß, wenn das Quellfeld keinen oder einen unzulässigen Wert enthält, oder wenn kein entsprechendes Quellfeld vorhanden ist, der Inhalt des Feldes aber für die Funktionalität der Applikation benötigt wird.

2.3.1.3 *Erarbeitung des Field Mapping*

Nachdem durch die Analyse festgestellt worden ist, welche Quell- und Zielfelder existieren und welche Datentypen diese haben bzw. haben sollen, muß die Zuordnung der Quell- zu den Zielfeldern erfolgen. Dabei muß der Datentyp des Zielfeldes festgelegt werden, den man als Zieldatentyp bezeichnet. Bei der Festlegung des Zieldatentyps wird zwischen impliziter und expliziter Festlegung unterschieden. Wird der Zieldatentyp explizit angegeben, ist eine Konvertierung des Datentyps des Quellfeldes nötig, wenn dieser nicht dem Zieldatentyp entspricht. Da die erweiterten Datentypen nicht Ziel einer Konvertierung sein können, kann in diesem Fall nur eine implizite Angabe des Typs vorgenommen werden, d. h. der Typ des Zielfeldes ist mit dem des Quellfeldes in jedem Fall identisch. Eine explizite Angabe des Zieldatentyps ist also nur bei Basisdatentypen möglich.

Bei der Erarbeitung des Field Mapping sollte berücksichtigt werden, daß die Zusammenführung mehrerer Felder nur dann möglich ist, wenn es sich bei den Datentypen der Quellfelder um Basistypen handelt (siehe auch 2.1.2). Eine Zusammenführung ist dabei entweder möglich, indem das Zielfeld eine Liste mit allen Inhalten der einzelnen Quellfelder enthält, oder beim Datentyp Text als zusammengesetzter Text und beim Datentyp Zahl als Summe. Wenn die Quellfelder dabei unterschiedliche Datentypen aufweisen, muß eine Konvertierung der Datentypen erfolgen. Das Zuweisen eines Quellfeldes zu mehreren Zielfeldern ist aber in jedem Fall für alle Datentypen möglich.

Zusätzlich ist zu berücksichtigen, daß evtl. Sonderfälle auftreten, die einer speziellen Behandlung bedürfen. Dies ist immer dann der Fall, wenn der Inhalt des Zielfeldes nicht identisch mit, sondern abhängig vom Inhalt des Quellfeldes ist. Soll z. B. das Field Mapping für ein Schlüsselwortfeld erarbeitet werden, wobei die Schlüsselworte in eine andere Sprache übersetzt werden sollen, ergibt sich eine Bedingung nach dem Muster: Wenn das Quellfeld das Wort „Keyword“ enthält, soll das Zielfeld das Wort

„Schlüsselwort“ enthalten. Dies kann z. B. durch eine @Funktion der Lotus Notes Makrosprache ausgedrückt werden. Sie würde für das vorliegende Beispiel lauten:

```
@If(Quellfeld = "Keyword"; @SetField("Zielfeld"; "Schlüsselwort");")
```

Ein weiteres Beispiel für die Notwendigkeit einer Veränderung der Inhalte des Quellfeldes ist, wenn an die bestehenden Inhalte eine Konstante angefügt, z. B. aus „Automobilzulieferer“ der Text „Automobilzulieferer (Motor)“ werden soll. Auch dieses Beispiel kann mit einer Formel gelöst werden.

Zum Abschluß der Erarbeitung des Field Mapping muß entschieden werden, welche Quellfelder nach der Erstellung der Zielfelder gelöscht werden sollen, da sie im Regelfall nicht mehr benötigt werden, weil sich ihre Inhalte in den Zielfeldern befinden.

2.3.1.4 Erstellung eines Migrationsagenten

Da die Umstellung der Datenstruktur nicht manuell durchgeführt werden kann, muß hierfür ein Werkzeug entwickelt werden, das in der Terminologie der Groupware Plattform Lotus Notes als Agent bezeichnet wird. Der Agent ist i. d. R. auf das aktuelle Migrationsprojekt zugeschnitten und paßt anhand des erarbeiteten Field Mappings die Datenstruktur an. Er kopiert dabei den Inhalt eines oder mehrerer Quellfelder unter Berücksichtigung der inhaltlichen Abhängigkeiten, die im Field Mapping spezifiziert wurden (siehe 2.3.1.3), in das oder die Zielfelder. Danach werden ggf. die Quellfelder gelöscht.

Bei der Programmierung des Agenten ist zu berücksichtigen, daß Mechanismen vorgesehen werden sollten, um die Veränderungen an den Daten der Dokumente nachvollziehen, und aufgetretene Fehler protokollieren zu können.

2.3.1.5 Simulation

Bei der Durchführung der Migration können Fehler auftreten. Wenn bei der Analyse der vorhandenen Daten (2.3.1.1) alle Dokumente überprüft wurden, werden die möglichen Fehlerquellen zwar reduziert, trotzdem kann jedoch der Programmcode selbst noch Fehler enthalten. Daher muß die Migration der operativen Umgebung zuvor simuliert werden. Dazu wird eine Replik der Datenbank angelegt und ein Testlauf des Migrationsagenten durchgeführt. Das Protokoll des Agenten und die migrierten Daten müssen danach analysiert und ausgewertet werden. Nur wenn die

Migration der Testumgebung erfolgreich war, sollte die Migration der operativen Umgebung durchgeführt werden.

2.3.1.6 Anpassung

Ist die Simulation nicht erfolgreich verlaufen, muß eine Anpassung des Field Mappings und als Folge dessen eine Anpassung des Migrationsagenten erfolgen. Die Anpassung des Field Mapping sowie die Durchführung und Auswertung der Simulation muß solange wiederholt werden, bis der Testlauf erfolgreich war.

2.3.2 Durchführung

2.3.2.1 Sicherung

Vor der eigentlichen Durchführung der Migration sollte eine Sicherungsreplik erstellt werden. Außerdem sollte vor Erstellung der Replik nach Möglichkeit in der Zugriffskontrollliste der Zugang zur Datenbank für die Benutzer auf lesenden Zugriff beschränkt werden, damit keine neuen Dokumente mehr angelegt werden können, die nicht migriert würden. Danach sollte die Datenbank mit allen weiteren vorhandenen Repliken repliziert werden, um die veränderte Zugriffskontrollliste zu übertragen.

2.3.2.2 Migration

Die Migration sollte auf einer weiteren lokalen Replik ausgeführt werden. Dies hat den Zweck, bei Abstürzen des Rechners oder des Notes Clients die operative Umgebung nicht zu gefährden, sowie keine Serverlast durch den Migrationsagenten zu erzeugen. Wenn die Migration erfolgreich verlaufen ist, wird die Datenbankgestaltung durch die neue Gestaltung ersetzt. Danach sollte eine Funktionsprüfung der Applikation stattfinden. Wenn diese positiv ausgefallen ist, kann die migrierte Datenbank in die operative Umgebung repliziert werden. Danach wird die Zugriffskontrollliste ggf. in den ursprünglichen Zustand zurückversetzt und mit allen vorhandenen Repliken repliziert.

2.3.3 Dokumentation

Nach Abschluß der Migration muß die Planung und Durchführung dokumentiert werden, damit bei später auftauchenden Problemen nachvollzogen werden kann, was bei der Migration an der Datenbank verändert worden ist. Daher sollte das Field Mapping festgehalten werden. Außerdem sollte protokolliert werden, welche

Änderungen an den Dokumenten vom Migrationsagenten durchgeführt worden sind, um diese von nachträglich manuell durchgeführten Änderungen unterscheiden zu können.

3. Anforderungen an ein Werkzeug

3.1 Allgemein

Aus den Phasen einer Migration lassen sich Anforderungen an ein generisches Werkzeug ableiten. Der Unterschied zu einem Migrationsagenten liegt dabei in erster Linie darin, daß die Anwendung des Werkzeugs nicht speziell auf einen Migrationsfall beschränkt ist, sondern daß es sich bei jedem beliebigen Projekt verwenden läßt. Außerdem sollte das Werkzeug wenn möglich alle Phasen der Migration unterstützen. Da der größte Aufwand bei einem Migrationsprojekt darin besteht, das Field Mapping zu erarbeiten und den Migrationsagenten zu programmieren, sollte der Schwerpunkt der Unterstützung auf diesen Phasen liegen.

Insbesondere kann gerade die Programmierung des Migrationsagenten ein Problem darstellen, da z. B. Richtextfelder in Lotus Notes Version 4.x nur mit Hilfe von Lotus Script migriert werden können. Daher sollte das Werkzeug benutzungsfreundlich sein und die meisten Standardfälle einer Migration ohne Programmierkenntnisse durchführen können. Da aber jedes Projekt spezielle Anforderungen hat, sollte das Werkzeug skalierbar sein, damit auch komplexere Aufgaben gelöst werden können. Es sollte überdies Schnittstellen enthalten, die es leicht erweiterbar machen. Dies gilt sowohl für den Programmcode, der die Migration durchführt, den sogenannten Migrationskern, als auch für die Benutzungsschnittstelle.

3.2 GCC Kontext

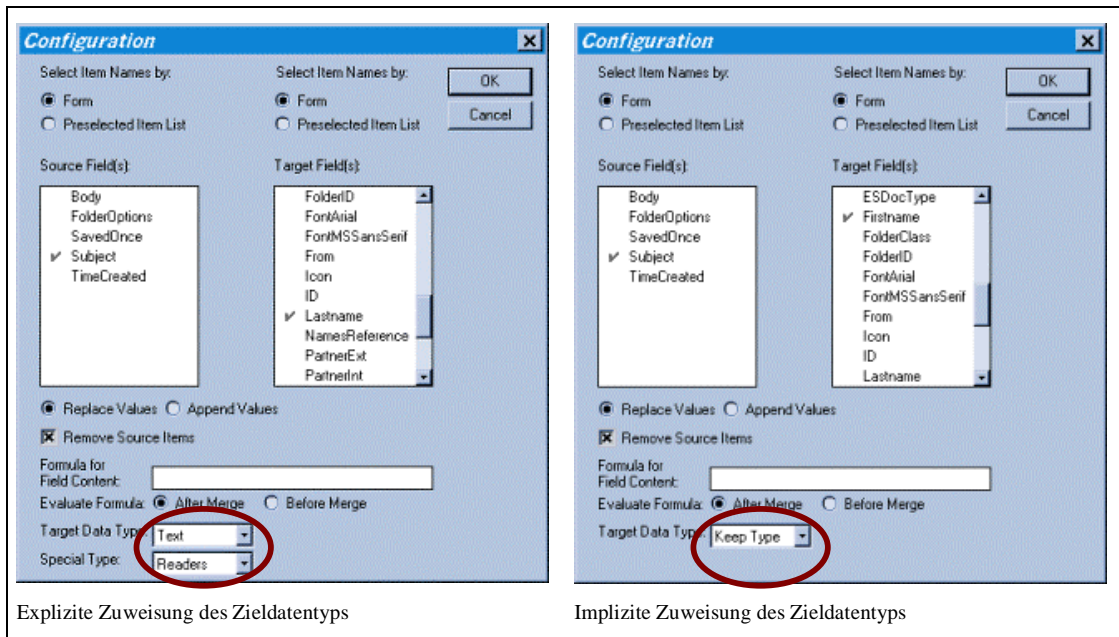
Aus der Erfahrung aus zahlreichen Migrationsprojekten am GCC haben sich spezielle Anforderungen an ein generisches Werkzeug ergeben. So ist neben der Unterstützung der Migration von Richtextfeldern wichtig, daß neue Felder, die nicht in der Zielmaske enthalten sind, vom Benutzer erzeugt werden können, wenn diese für die neue Applikation notwendig sind. Um die Erstellung des Field Mappings benutzungsfreundlich zu gestalten, sollte zudem die Auswahlliste der Feldnamen aus den Quell- bzw. Zielmasken generiert werden. Da aber beim Field Mapping auch die Felder zugeordnet werden können müssen, zu denen keine entsprechenden Maskenfelder existieren, sollten die Feldnamen ebenso aus einer Liste von selektierten Beispieldokumenten ausgelesen werden können. Diese Methode stellt sicher, daß alle vorhandenen Felder erfaßt und ausgewählt werden können.

Es sollte außerdem möglich sein, für einen fortgeschrittenen Benutzer die Verwendung von Formeln der Lotus Notes Makrosprache ohne Anpassung des Migrationskerns, sowie implizite und explizite Zuweisung des Zieldatentyps zu ermöglichen. Daraus resultiert die Forderung, daß das Werkzeug eine automatische Erkennung des Zieldatentyps durchführen kann, wenn der Zieldatentyp implizit zugewiesen wurde. Dieses Problem ist trivial bei der Migration eines einzelnen Feldes oder bei der Zusammenführung mehrerer Felder gleichen Datentyps. Sollen jedoch Felder unterschiedlichen Datentyps zusammengeführt werden, muß ein geeigneter Zieldatentyp automatisch ausgewählt werden. Außerdem sollte die Zusammenführung als Liste zusätzlich durch die Möglichkeit ergänzt werden, eine Zusammenführung als Summe bzw. als zusammengesetzter Text zu ermöglichen.

3.3 Umsetzung in einen Prototypen

Dieses Kapitel ist eine kurze, generelle Funktionsbeschreibung und keine technische Dokumentation. Der Prototyp ist eine eigenständige Datenbank, in der alle notwendigen Konfigurationen vorgenommen werden können, die für die Migration einer Datenbank notwendig sind. Er ist unter Lotus Notes ab der Version 4.5 lauffähig und in der Lage, Datenbanken aller Lotus Notes Versionen zu migrieren, auf die der verwendete Client zugreifen kann. Der Prototyp erfüllt alle in 3.1 und 3.2 aufgeführten Anforderungen.

In der Datenbank wird für jeden zu migrierenden Dokumententyp ein Konfigurationsdokument angelegt. In diesem wird der Server und der Pfad der zu migrierenden Datenbank, sowie der Name der Maske, also der Dokumententyp, angegeben. Damit das Auslesen der Felder der Zielmaske möglich ist, muß auch der Pfad des Templates angegeben werden, das die zukünftige Gestaltung der Datenbank enthält. Über eine Aktionsschaltfläche ist es möglich, der Liste von Field Mappings ein neues Mapping hinzuzufügen. Über ein Dialogfenster (Abb. 3) wird das Field Mapping eingegeben. Wenn dies abgeschlossen ist, kann die Migration direkt über den Druck auf eine Schaltfläche durchgeführt werden. Dabei besteht die Möglichkeit, vor der physikalischen Migration erst eine Simulation durchzuführen. Der Vorteil dabei ist, daß das Field Mapping zwecks Fehlererkennung ausprobiert werden kann, ohne für jede Anpassungsphase erneut eine Replik erstellen zu müssen.



Explizite Zuweisung des Zieldatentyps

Implizite Zuweisung des Zieldatentyps

Abb. 3: Field Mapping im Prototypen

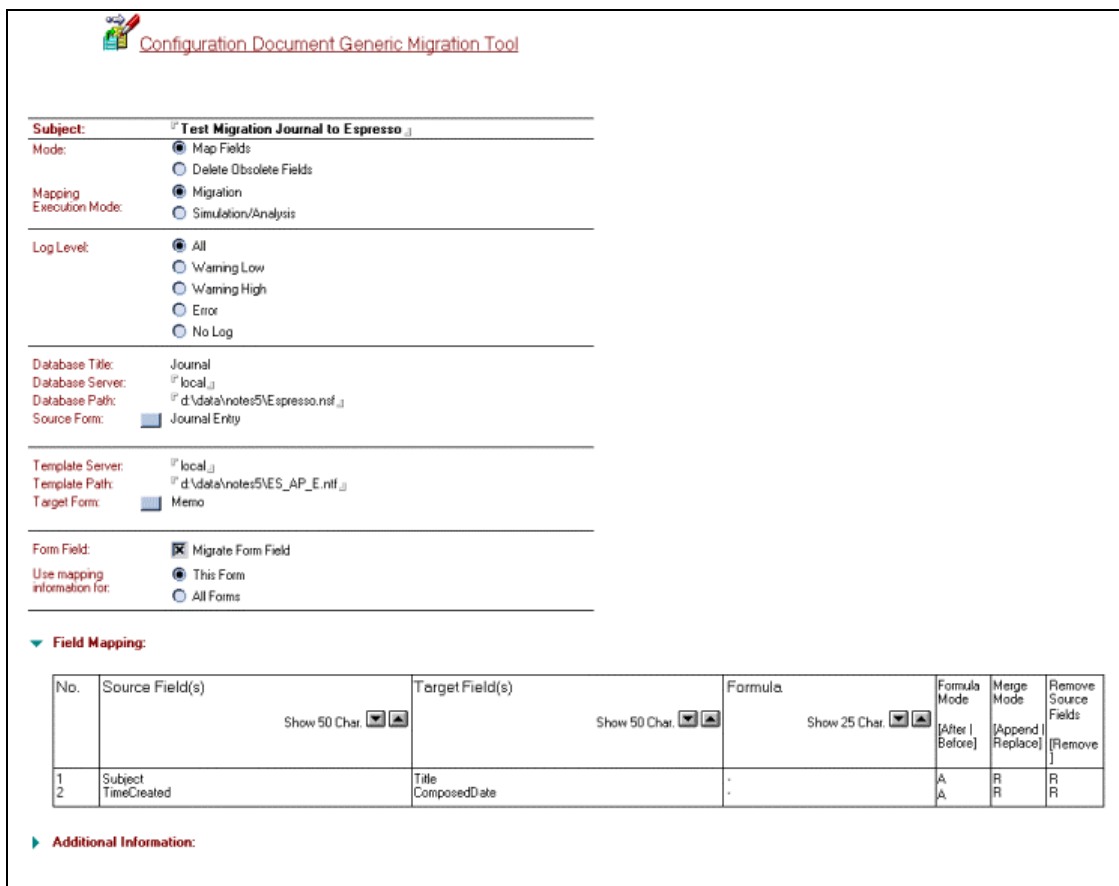


Abb. 4: Beispiel für ein Konfigurationsdokument

Unabhängig davon, ob eine Migration oder eine Simulation durchgeführt wird, wird ein Protokolldokument erstellt, in dem abhängig von einer zuvor wählbaren Genauigkeitsstufe, die Ereignisse und Fehler, die während der Migration aufgetreten sind, aufgezeichnet werden. Dieses Protokolldokument wird als Antwortdokument

dem Konfigurationsdokument zugeordnet und nach Abschluß der Migration automatisch geöffnet. Da im Laufe des Anpassungsprozesses des Mappings und der Simulationen mehrere Protokolle erstellt werden, läßt sich so die Entwicklung des Field Mapping im Laufe des Projektes nachvollziehen.

4. Ausblick

Das Werkzeug kann in verschiedenen Punkten ausgebaut und verbessert werden. So wird für die Analyse der vorhandenen Daten zwar die Möglichkeit unterstützt, die Feldnamen aus Dokumenten auszulesen, doch für eine vollständige Analyse wäre es nötig, auch Datentypen der Felder festzustellen sowie Dokumente aufzuzeigen, in denen sich Felder befinden, die einen Datentyp aufweisen, der nicht dem erwarteten Typ entspricht. Dies ist wichtig, um mögliche Fehlerquellen z. B. bei der Konvertierung von Daten, frühzeitig zu erkennen. Auch wäre eine Anzeige des aktuellen Datentyps neben dem Namen des Feldes beim Erstellen des Field Mappings hilfreich, um die Ist-Situation des Dokumentenbestandes besser einschätzen zu können.

Das Benutzerinterface könnte dahingehend verbessert werden, daß Standardprobleme wie die Migration von Schlüsselwortfeldern, das Setzen von Vorgabewerten für leere Felder oder das Anhängen von Konstanten an Feldwerte nicht über Formeln, sondern über eigene Eingabefelder vorgenommen werden kann. Außerdem würde ein graphischer Zusatz, bei dem sich die Felder von den Masken direkt auf die Zielmaske ziehen lassen, das Werkzeug intuitiver benutzbar machen.

Die Simulationskomponente unterstützt zur Zeit nur die Simulation von Mapping Kombinationen, bei denen ein Mapping nicht auf dem Ergebnis eines früher auf dem gleichen Dokument ausgeführten Mappings basiert. Ist dies trotzdem der Fall, liefert die Simulation ein anderes, also falsches Ergebnis, als die Migration. Dieses Problem kann in einer zukünftigen Version dadurch gelöst werden, daß während einer Simulation alle Veränderungen der Daten auf einem virtuellen Dokument ausgeführt werden.

Die Protokolle der Migration sollten das Mapping, das für die protokollierte Durchführung der Migration verwandt wurde enthalten, damit sich die Entwicklung des Mappings im Laufe eines Projektes noch besser verfolgen lassen könnte.

Auch die Performance bietet u. a. aufgrund der objektorientierten Programmierweise Möglichkeiten zur Verbesserung. Da bei jedem Mapping unabhängig von der Komplexität alle Sonderfälle berücksichtigt werden, wird die Performance zusätzlich beeinträchtigt. Sondermodule für die am häufigsten auftretenden Migrationsfälle könnten die Performance stark verbessern. Dabei sollte auch die Funktion

„MineType“ verbessert werden, die für die Auswahl des Zieldatentyps bei der Zusammenführung mehrerer Felder unterschiedlicher Datentypen bei impliziter Zuweisung des Zieldatentyps (siehe auch 3.2) zuständig ist, denn diese Funktion wählt in der aktuellen Version den ersten konvertierbaren Datentyp als Zieldatentyp aus, was in einigen Fällen nicht optimal sein kann.

Da das Werkzeug die Datenstruktur von Dokumenten migriert und in Lotus Notes Datenbanken Gestaltungselemente in speziellen Dokumenten abgelegt werden, ist es unter Umständen in einer kommenden Version sogar möglich, auch die Migration der Datenbankgestaltung zu unterstützen.

5. Zusammenfassung

Die Migration einer Applikation kann sehr aufwendig sein. Hier kann ein generisches Werkzeug den Prozeß, der für eine Migration notwendig ist, unterstützen, die Qualität der Dokumentation verbessern und die Zeit, die für das Projekt benötigt wird, verkürzen. Der Prototyp zeigt außerdem, daß es möglich ist, eine Migration in allen Phasen durch das Werkzeug zu unterstützen und insbesondere den Aufwand für die Programmierung eines Migrationsagenten einzusparen.

Da viele Gründe für eine Migration existieren (siehe auch 1.2) und daher Migrationen häufig durchzuführen sind, ist die Verfügbarkeit eines generischen Werkzeugs wichtig und hilfreich. Es kann sogar die Möglichkeiten erhöhen, die ein Unternehmen bei der Auswahl einer Applikation hat. Denn wenn ein Unternehmen z. B. zu einem Konkurrenzprodukt wechseln möchte, wird die Entscheidung für den Umstieg auf das jeweils am besten geeignete Produkt wesentlich erleichtert, wenn Migrationen mit dem Werkzeug einfacher, sicherer und schneller durchgeführt werden können als bisher.

Wie sich aus dem umfangreichen Ausblick ergibt, hat der Prototyp bis zu einem benutzungsfreundlichen Standardprodukt jedoch noch einen langen Entwicklungsprozeß vor sich.

Literaturverzeichnis

Lotus Development Corporation (1999): Domino 5 Designer Help, Version 5.3.10.9; aus http://doc.notes.net/domino_notes/5.0/help5_designer.nsf am 01.08.1999

Collins, F. (1999): Lotus Domino R5: A Developer's Handbook, First Edition, IBM Direct Publishing, 1999