



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Sahabettin Alkan

Design eines generischen Archivierungstools für
SAP-Anwendungen

Sahabettin Alkan
Design eines generischen Archivierungstools für
SAP-Anwendungen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Wolfgang Gerken
Zweitgutachter : Prof. Dr. Olaf Zukunft

Abgegeben am 25. März 2009

Sahabettin Alkan

Thema der Bachelorarbeit

Design eines generischen Archivierungstools für SAP-Anwendungen

Stichworte

Archiv, Archivierung, SAP, ADK, Archive Engine

Kurzzusammenfassung

In dieser Arbeit sollen die Möglichkeiten aufgezeigt werden, eine allgemeine Datenarchivierungslösung unter SAP zu erstellen. Dabei wird ein Design eines generischen Systems entwickelt, in dem die Daten in Datenbanktabellen stehen, und die Anwendung nur zur Laufzeit weiß, welche Daten welcher Tabellen archiviert werden. Anhand eines Prototyps werden auch die Punkte, wie z.B. Zugriffssicherheit, Revisionsicherheit usw., erläutert, die eine Archivierungsanwendung mindestens erfüllen muss, damit die Daten ihre Beweiskraft in einer Rechtssituation nicht verlieren.

Sahabettin Alkan

Title of the paper

Design of a generic Archivingtool for SAP-Applications

Keywords

Archive, Archiving, SAP, ADK, Archive Engine

Abstract

In this thesis the possibilities to implement a general data archiving solution under SAP are to be illustrated. The design of a generic system will be developed which stores the data in database tables, and the application knows only at runtime which data of which tables are being archived. Based on a prototype points, such as access security, revision security etc., will also be explained which an archiving application must minimal fulfills, so that the data do not lose their evidential value in a court proceeding.

Danksagung

Als erstes möchte ich allgemein Paricon Products GmbH danken, das mir ermöglicht hat, die Bachelorarbeit anzufertigen.

Mein größter Dank geht an Achim von Lowtzow, der mich kompetent betreut und auch bei der schriftlichen Ausarbeitung wertvolle Anregungen gegeben hat.

Mein herzlicher Dank gilt auch weiteren Mitarbeitern von Paricon Products GmbH, Christian Holtz, der mir allgemein geholfen und Mittel zur Verfügung gestellt hat, und dann Johannes Bohn, der mir bei fachlichen Sachen Hilfestellungen gegeben hat.

Für die Betreuung seitens der Hochschule für Angewandte Wissenschaften Hamburg danke ich Herrn Prof. Dr. rer. nat. Wolfgang Gerken und Prof. Dr. Olaf Zukunft.

Ich danke auch allen Kommilitonen, insbesondere Herrn Ahmet Inci, Herrn Ali Kilic, Herrn André Schmer, Herrn Dennis Winter, Herrn Fahim Aleaf, Herrn Mahmut Cevik, Herrn Mehdi Afridi, Herrn Remzi Musici für die tolle Zusammenarbeit während meines Studiums.

Last but not least möchte ich auch meiner Familie danken, die mich in allen Bereichen unterstützt haben.

Sahabettin Alkan, März 2009

Inhaltsverzeichnis

Abbildungsverzeichnis	7
1 Einleitung	8
1.1 Motivation	8
1.2 Problemstellung	8
1.3 Zielsetzung	9
1.4 Aufbau der Arbeit	9
2 Elektronische Archivierung	11
2.1 Elektronische Langzeitarchivierung	12
2.2 Revisions sichere elektronische Archivierung	12
2.2.1 Gesetzliche Anforderungen an die Langzeitarchivierung	13
2.3 Grundlegende Arten der Archivierung	17
2.4 Ansätze zur elektronischen Archivierung	18
2.4.1 Strategien zur Langzeitverfügbarkeit	18
2.4.2 Speichertechnologien	23
3 Analyse	24
3.1 Datenarchivierung im SAP	24
3.1.1 ADK - Archive Development Kit	24
3.1.2 XML-Archivierung	25
3.1.3 Fazit	27
3.2 Beschreibung des vorhandenen Systems	27
3.2.1 STS - Standard Trafoschicht	27
3.2.2 STS-Archivierung	28
3.3 Gründe für eine Neuentwicklung	31
3.3.1 Marktanalyse	31
3.4 Anforderungsanalyse	32
3.4.1 Funktionale Anforderungen	32
3.4.2 Nichtfunktionale Anforderungen	34

4 Design	35
4.1 Allgemeine Designansätze	35
4.1.1 Dynamische Programmierung	35
4.1.2 Programmgenerierung	36
4.2 Ablauf der Archivierung	37
4.3 Archivierungsobjekte	38
4.4 Datenmodell	39
4.4.1 Customizingtabellen	42
4.4.2 Laufinfotabellen	44
4.4.3 Texttabellen	45
4.5 Klassen	46
4.6 GUI	47
4.7 Archivierungsformat	49
4.7.1 Metadaten	49
4.7.2 Anzeigen der Daten	50
4.7.3 Export der Daten	50
5 Realisierung	51
5.1 Archivieren mit dynamischer Programmierung	51
5.1.1 Speichern der Daten für Tabellenbeziehungen	53
5.1.2 Anordnen der Archivierungsobjekte	54
5.1.3 Dynamische Where-Bedingung	55
5.2 Archivieren mit Programmgenerierung	56
5.3 Gruppierung der Daten	57
5.4 Archivieren von generierten Tabellen	59
5.5 Fazit	61
6 Schluss	62
6.1 Zusammenfassung	62
6.2 Fazit	63
6.3 Ausblick	63
Literaturverzeichnis	65
Glossar	66
Index	68

Abbildungsverzeichnis

3.1	Komponenten der XML-Archivierung (Quelle: SAP)	26
3.2	Paricon Data Management Suite	28
3.3	Ablauf der STS-Datenversorgung	29
4.1	Ablauf der Archivierung	38
4.2	ERD-Diagramm	40
4.3	Klassendiagramm der Anwendung	46
4.4	STS-Archivierung	48
5.1	Ablauf der Archivierung	52
5.2	Fremdschlüsseltabelle	53
5.3	Klasse zur Programmgenerierung	56
5.4	Ablauf der Generierung	58
5.5	Architektur von BAdI (Quelle:SAP)	60

1 Einleitung

1.1 Motivation

Seit der Erfindung der Schriftrolle möchte der Mensch die wichtigsten Dokumente aufbewahren, um später auf die Schriften zugreifen zu können. Dafür hat die Menschheit in seiner Geschichte viele Lösungen entwickelt. Und heutzutage, das wir Informationszeitalter bezeichnen, ist es notwendig geworden, die Informationen bzw. Daten, die mit der Hand nicht zugreifbar sind und ausschließlich in Bits und Bytes dargestellt werden, über einen längeren Zeitraum aufzubewahren und später in einem von Menschen lesbarer Form zu bringen.

Unter Archivierung versteht man dabei das Verschieben der Daten von selten benötigten betriebswirtschaftlichen Objekten aus den produktiven Datenbanksystemen auf kostengünstigere Tertiärspeichersysteme. Dadurch werden die Datenbasis der produktiven Datenbank verkleinert, die Leistung der Datenbank erhöht und letztlich Kosten gesenkt.

1.2 Problemstellung

Unter SAP existieren unzählige Anwendungen, die speziell für verschiedene Aufgabenbereiche definiert sind. Grundlage dieser Anwendungen oder besser gesagt des SAP-Systems bildet ein Datenbanksystem, das die Daten in Tabellenform speichert.

Wenn man von einem Unternehmen ausgeht, stellen wir uns z.B. eine Bank vor, wo täglich tausende Überweisungen und andere Bewegungen auf den Kontos stattfinden, kommt man sehr schnell insgesamt Millionen von Einträgen in den Tabellen.

Ein allgemeines Problem bei Datenbanken ist, dass es keine unendlich großen Tabellen gibt, sondern die Größe der Tabellen durch Verfügbarkeit, Ressourcen und Antwortzeiten beschränkt sind. Im Laufe der Zeit kann die Datenbankgröße im Terabyte-Bereich liegen, was zu Performanceproblemen beim Datenbanksystem und abhängig davon bei den Anwendungssystemen.

Um das Problem zu lösen, dürfen die alten Daten, die für das Tagesgeschäft nicht mehr relevant sind, wegen der gesetzlichen Vorschriften auch nicht gelöscht werden. Die müssen

mindestens sechs und bei einigen sogar zehn Jahre lang aufbewahrt werden, und der Zugriff auf die Daten muss in diesem Zeitraum jederzeit möglich sein.

1.3 Zielsetzung

Ausgehend von obiger Problemstellung soll diese Bachelorarbeit Wege und Implementierungsvorschläge für eine allgemeine Archivierungslösung für beliebige Komponenten im SAP-System aufzeigen.

Grundlage der Implementierung bildet eine Archivierungslösung vom Unternehmen Paricon, die aber speziell für eine Komponente implementiert ist. Die Anwendung bietet keine Möglichkeit durch Einstellungen oder ähnliches andere SAP-Komponenten zu archivieren.

Kern dieser Arbeit ist nicht die Entwicklung einer voll funktionsfähigen Archivierungsanwendung, sondern ist ein konzeptioneller Entwurf eines Prototyps. Dabei soll gezeigt werden, welche Möglichkeiten es unter SAP gibt oder welche Tools schon SAP bietet, um eine allgemeine Archivierungslösung zu entwickeln. Bei diesen Tools soll auch auf Vor- und Nachteile eingegangen werden.

Wie der Titel dieser Bachelorarbeit schon sagt, soll der Prototyp generisch sein, das heißt, der Benutzer soll über Einstellungen festlegen können, welche Anwendungen und welche dazugehörigen Tabellen in SAP archiviert werden sollen oder nicht. Die Archivierungsanwendung würde sozusagen erst zur Laufzeit wissen, welche Daten sie lesen soll. Dafür bietet ABAP, die Programmiersprache der SAP, Techniken, mit den das Problem gelöst werden kann. In dieser Arbeit sollen auch diese Techniken vorgestellt, und die Probleme, die dabei entstehen, aufgezeigt werden.

1.4 Aufbau der Arbeit

Kapitel 2 beschäftigt sich mit den Grundlagen der elektronischen Archivierung und gibt einen kleinen Überblick über die Archivierungsarten. Dabei wird der Begriff Revisionssicherheit diskutiert und Aspekte der Langzeitarchivierung aufgezählt.

Nach dem die Grundlagen vorgestellt wurden, werden in der Analysephase (Kapitel 3) die Systeme SAP und die aktuelle Archivierungsanwendung untersucht. Es wird überprüft, welche Techniken und Methoden SAP zur Lösung der Aufgabe anbietet. Ausgehend von der aktuellen Archivierungslösung werden die Anforderungen für die neue Anwendung aufgestellt.

In Kapitel 4 (Design) folgt, basierend auf den in Kapitel 3 identifizierten Anforderungen, die schrittweise Herleitung und Beschreibung der Softwarearchitektur sowie die Vorstellung der verwendeten Technologien.

Auf Einzelheiten der Implementierung wird in Kapitel 5 (Realisierung) eingegangen. Die wichtigsten Abläufe und Funktionen werden hier mit Hilfe von UML–Diagrammen eingehend beschrieben.

Im letzten Kapitel erfolgt dann eine Zusammenfassung der Arbeit sowie ein Ausblick auf mögliche Weiterentwicklungen des Prototyps bis hin zum Produktivsystem.

2 Elektronische Archivierung

Der Begriff Archiv kommt aus dem Griechischen dem Wort *Archeion*. Damals in der Antike wurden wichtige Dokumente der Stadt und ihrer Bürger im Archeion, dem Rathaus, verwahrt.

Allgemein wird unter Archiv als eine Einrichtung, eine Institution oder Organisationseinheit zur Sammlung, Ordnung, Aufbewahrung, Verwaltung und Nutzung von meist schriftlichen und auch anderen Überresten aus dem Bereich der Verwaltung oder auch anderen öffentlichen oder privaten Institutionen verstanden. Im übertragenen Sinne bezeichnet es auch das Gebäude.

Für den Begriff findet man noch weitere unterschiedliche Erklärungen, die für das jeweilige Gebiet relevanten Schwerpunkte hervorbringen. Als größter gemeinsame Teiler vom ganzen ist zu verstehen, dass die Aufgabe vom Archiv die langfristige Aufbewahrung von Dokumenten und Informationen unabhängig von der Art und Funktionsweise der Aufbewahrung ist.

Die Aufgabe der elektronischen Archivierung besteht darin die digitalen Dokumente und Informationen unveränderbar über einen langen Zeitraum aufzubewahren und zu jeder Zeit frei zugänglich zu machen. An dieser stellt sich aber die Frage, was sich die Archivierung von der Datensicherung unterscheidet. Denn heutzutage werden fast in jedem Unternehmen wöchentlich und sogar in einigen täglich Datensicherungen durchgeführt. Reicht eine Datensicherung nicht aus? Ist die Archivierung nicht ein zusätzlicher Aufwand?

Um die Frage beantworten zu können, muss es untersucht werden, was das Ziel bei beiden Methoden eigentlich ist. Danach ist es klar erkennbar, dass bei beiden verschiedene Strategien verfolgt wird. Der Sinn und Zweck des Backups ist die Vermeidung des Datenverlustes bei HW- oder SW-Fehler oder bei versehentlichem Löschen oder Überschreiben. Die Dauer der Aufbewahrung ist ziemlich kurz, d.h. täglich bis einigen Monaten. Anhand der Aufbewahrung kann man schon erkennen, dass ein Backup die gesetzlichen Anforderungen an Aufbewahrungsfristen nicht erfüllt und daher als Ersatz der Archivierung nicht in Frage kommt.

Die Motivation bei der elektronischen Archivierung ist das mittel- und langfristige Aufheben von Daten, Dateien, Dokumente und Informationen. Heutzutage entstehen sehr viele Dokumente in elektronischer Form und durch den Einsatz elektronischer Signaturen erhalten elektronische Dokumente den gleichen Rechtscharakter wie ursprünglich manuell unterzeichne-

te Schriftstücke. Der Ausdruck auf Papier dient nur noch zur Repräsentation. Da die Daten auch wegen der gesetzlichen Aufbewahrungsfristen nicht einfach gelöscht werden dürfen, um bei Volumen- und Performanceproblemen Abhilfe zu schaffen, ist in der Regel ein Zwang zur Archivierung gegeben.

Laut Kampffmeyer umfasst der Begriff Elektronische Archivierung im Deutschen unterschiedliche Komponenten zusammen, die im angloamerikanischen Sprachgebrauch separat als „Records Management“, „Storage“ und „Preservation“ bezeichnet werden. [Kam05]

In den Beschreibungen findet man zwei Begriffe, die von vielen Autoren synonym zueinander benutzt werden: Elektronische Langzeitarchivierung und Revisions sichere elektronische Archivierung.

2.1 Elektronische Langzeitarchivierung

Es gibt keine allgemeingültige Größe, wie viele Jahre man unter dem Begriff „Langzeit“ versteht. Man orientiert sich dabei an die Gesetze der jeweiligen Branche.

Man spricht von Langzeitarchivierung, wenn die Informationen mindestens 10 Jahre und länger aufbewahrt und zugreifbar gehalten werden. [Kam05] Die Aussage von Herrn Kampffmeyer ist sofern richtig, weil im kaufmännischen Bereich die steuerrechtlich wichtigen Dokumente 10 Jahre lang aufbewahrt werden müssen. Im Versicherungsbereich beträgt die Aufbewahrungsfrist mindestens 30 Jahre und in historischen Archiven spricht man sogar von 100, 200 oder gar 300 Jahre.

Entsprechend unserer Aussage, dass man bei der Definition auf die Gesetzestexte bezieht, kann man bei der Definition von Brigitta Nimz sehr gut erkennen: „... unter dem Begriff „Langzeitarchivierung“ wird eigentlich nur die revisions sichere Aufbewahrung der Daten entsprechend den gesetzlichen Aufbewahrungsfristen verstanden.“ [Nim00]

Nicht nur Nimz sondern auch viele andere Autoren unterscheiden nicht zwischen der Langzeitarchivierung und revisions sichere Archivierung. Wenn sie von Langzeitarchivierung sprechen, meinen die damit die revisions sichere Archivierung.

2.2 Revisions sichere elektronische Archivierung

Wenn kaufmännische Daten und Dokumente zur digitalen Archivierung anstehen, müssen diese so aufbereitet sein, dass eine nachträgliche Veränderung ausgeschlossen ist. Aber nicht nur die Daten selbst, sondern auch die erforderlichen Strukturinformationen müssen gesichert werden und vor allem maschinell auswertbar zur Verfügung stehen.

Der Verband Organisations- und Informationssysteme e.V. (VOI) hat in seinem Code of Practice „Grundsätze der elektronischen Archivierung“ folgende Definitionen aufgeführt:

Revisionssichere Archivierung: Unter „revisionssicherer Archivierung“ versteht man Archivsysteme, die nach den Vorgaben der Allgemeinen Abgabenordnung (HGB AO) und der GoBS Daten und Dokumente sicher, unverändert, vollständig, ordnungsgemäß, verlustfrei reproduzierbar und datenbankgestützt recherchierbar verwalten.

Dabei ist zu erwähnen, dass der Begriff „revisionssicher“ in keinem Gesetz und keiner Verordnung vorkommt. Er hat sich in den 90er Jahren auf Basis eines Vorschlages des Autors etabliert und dient zur Beschreibung nicht nur von Hard- und Softwarekomponenten sondern sicherer Gesamtverfahren der elektronischen Archivierung.

Man findet heutzutage fast für alle Bereiche in der IT Archivierungsprodukte, sei es zur Entlastung des Emailkontos oder ERP-Systeme oder nur für Datenbanksysteme. Die Anbieter werben zwar mit dem Begriff „revisionssicher“ und die Produkte werden auch als „revisionssicheres“ Produkt verkauft, ich konnte aber keine Behörde oder Institution finden, die die Revisionssicherheit des einzelnen Produktes zertifizieren. Sondern die Zertifizierung erfolgt durch den Wirtschaftsprüfer oder einer Zertifizierungsstelle für IT wie TÜVIT auf die individuelle Gesamtlösung im Einsatz beim Anwender. Hierbei ist zu berücksichtigen, dass es nicht nur um die Systemkomponenten geht, sondern der gesamte organisatorische Rahmen, der Betrieb der Lösung und die Prozesse im Sinne eines Gesamtverfahrens werden zeitpunktbezogen geprüft und zertifiziert.

2.2.1 Gesetzliche Anforderungen an die Langzeitarchivierung

Beim Einsatz von elektronischen Archivierungssystemen müssen eine Reihe von gesetzlichen Vorschriften beachtet werden. Bei dieser Bachelorarbeit geht es zwar nur um Datenarchivierung, aber um Vollständigkeitshalber werden alle Gesetze und Vorschriften, die die elektronische Archivierung im Allgemeinen betreffen, aufgeführt.

Neben den allgemeinen Vorschriften existieren gesetzliche Vorschriften und Standards in bestimmten Branchen, wie z.B. die Bestimmungen des Sozialgesetzbuches für Sozialversicherungsträger, besondere Aufbewahrungs-, Vertraulichkeits- und Sicherheitsvorschriften für die Patientenakten in Krankenhäusern, Regeln für Forschungs-, Produktions- und Antragsdokumentationen in der pharmazeutischen Industrie oder Standards im militärischen Umfeld.[Kam97]

Schriftliche Form und elektronische Signatur

Bei bestimmten Rechtsgeschäften ist die schriftliche Form gesetzlich vorgeschrieben. Nach § 126 BGB muss der Aussteller ein Dokument eigenhändig durch Namensunterschrift oder mittels notariell beglaubigtem Handzeichen unterzeichnen.

Neben einer Textform nach § 126b BGB wurde eine elektronische Form § 126a BGB als Sonderfall der Schriftform geschaffen. Elektronische Form kann nach § 126 Abs.3 die Schriftform ersetzen. Die elektronische Form bedarf des Einsatzes von sog. elektronischer Signatur, welche unter anderem auch die Unverfälschbarkeit des elektronischen Geschäftsverkehrs gewährleisten kann. Die Rahmenbedingungen für elektronische Signatur ergeben sich aus dem Signaturgesetz.

Darüberhinaus bestehen aber noch einige Ausnahmen, dass bei bestimmten Dokumenten die elektronische Signatur nicht gültig wird. In diesem Fall müssen die Dokumente weiter noch in Papierform aufbewahrt werden.

Im Falle eines revisionssicheren elektronischen Archivs müssen die Dokumente oder das Archiv nicht nochmal signiert werden, da das revisionssichere Archiv die Unveränderbarkeit der im Archiv gehaltenen Dokumente garantiert.

In einem Zivilprozess haben elektronische Dokumente im Sinne des §416 ZPO (Zivilprozessordnung) keine Urkundenqualität. Die Rechtsprechung spricht dann nach §371 ZPO von einem „Objekt des Augenscheins“. Es unterliegt nach § 286 ZPO dem freien Willen des Richters, ob er das Dokument als Beweismittel anerkennt oder nicht.

Anforderungen an die elektronische Archivierung

Die §§ 239, 257 HGB regeln die grundsätzlichen Voraussetzungen für die Archivierung von kaufmännischen Dokumenten, unabhängig davon, ob in Papier oder in elektronischer Form.

Die grundsätzlichen Anforderungen lassen sich wie folgt zusammenfassen:

- Ordnungsmäßigkeit
- Vollständigkeit
- Sicherheit des Gesamtverfahrens
- Schutz vor Veränderung und Verfälschung
- Sicherung vor Verlust
- Nutzung nur durch Berechtigte

- Einhaltung der Aufbewahrungsfristen
- Dokumentation des Verfahrens
- Nachvollziehbarkeit
- Prüfbarkeit

Anforderungen und Fristen an die Aufbewahrung

In der Abgabenordnung §§ 146, 147 AO (Abgabenordnung) und §§ 238, 239, 257, 261 HGB (Handelsgesetzbuch) sind die Anforderungen an die Aufbewahrung und die Prüfung von kaufmännischen Dokumenten aufgeführt.

- Bücher und Aufzeichnungen, Jahresabschlüsse, Lageberichte, Eröffnungsbilanz, Arbeitsanweisungen, Organisationsunterlagen und Buchhaltungsbelege müssen 10 Jahre und
- Handels- oder Geschäftsbriefe 6 Jahre aufbewahrt werden.

Digitale Betriebsprüfung (GDPdU)

Seit der Neufassung §§ 146, 147 AO (Abgabenordnung) und dem Schreiben des Bundesfinanzministeriums „Die Grundsätze zum Datenzugriff und zur Prüfbarkeit digitaler Unterlagen“ (GDPdU) vom 16.07.2001 darf der Steuerprüfer die EDV des Unternehmens für die Prüfung nutzen oder die Übergabe eines Datenträgers mit den steuerrelevanten Daten verlangen.

Die Änderungen in der AO führten zur digitalen Steuerprüfung:

- Während der Aufbewahrungsfristen müssen Daten jederzeit verfügbar sein, unverzüglich lesbar gemacht werden und maschinell ausgewertet werden können
- Sind Daten mit einem DV-System erzeugt worden, hat die Finanzbehörde das Recht, Einsicht zu nehmen und das System zur Prüfung zu nutzen
- Die Daten müssen maschinell auswertbar sein und auf Anforderung auf Datenträgern zur Verfügung gestellt werden

Hiernach sind elektronische, steuerrelevante Dokumente auf maschinell verwertbaren Datenträgern zu archivieren. Eine Aufzeichnung auf Mikrofilm reicht daher nicht aus, ebenso wenig die Speicherung in maschinell nicht lesbaren PDF-Dateien.

Das BMF-Schreiben zu den GDPdU verweist an mehreren Stellen auf die „Grundsätze ordnungsmäßiger DV-gestützter Buchführungssysteme“ (GoBS), die die handelsrechtlichen Anforderungen an die elektronische Buchführung konkretisieren. Die GoBS formulieren einen Standard für Aufbau und Ausgestaltung eines elektronischen Buchführungssystems, seine organisatorische und seine IT-technische Einbettung in das Unternehmen. Außerdem stellen sie Regeln für die laufende Überwachung und Dokumentation von Buchungsvorgängen und ihrer Verarbeitung auf. Anders als die GDPdU fordern die GoBS keine maschinelle Auswertbarkeit elektronisch gespeicherter Buchführungsunterlagen. [BIT06]

Personenbezogene Daten

Sofern die personenbezogenen Daten mit archiviert werden, muss auch das Bundesdatenschutzgesetz (BDSG) beachtet werden.

Dabei sind besonders die Rechte der Betroffenen, über die Daten und Dokumente gespeichert sind, von Interesse, speziell die Anforderungen an die Löschung und Sperrung von Daten. Denn revisionssichere Archive und digitale optische Speichermedien sind nämlich darauf ausgelegt, Löschungen und Veränderungen von Daten und Dokumenten auszuschließen. § 20 BDSG sieht aber vor die nicht mehr benötigten Daten zu löschen bzw. zu sperren.

Sonstige Vorschriften

Abgesehen von den Vorgaben des Gesetzgebers gibt es von Fachverbänden, Initiativen und Normungsgremien weitere Vorgaben und Empfehlungen zur rechtlichen Anerkennung der Archivierung. Hierzu gehören die zehn Merksätze des VOI für eine sichere, ordnungsgemäße Archivierung:

1. Jedes Dokument muss unveränderbar archiviert werden.
2. Es darf kein Dokument auf dem Weg ins Archiv oder im Archiv selbst verloren gehen.
3. Jedes Dokument muss mit geeigneten Retrievaltechniken wiederauffindbar sein.
4. Es muss genau das Dokument wiedergefunden werden, das gesucht worden ist.
5. Kein Dokument darf während seiner vorgesehenen Lebenszeit zerstört werden können.

6. Jedes Dokument muss in genau der gleichen Form, wie es erfasst wurde, wieder angezeigt und gedruckt werden können.
7. Jedes Dokument muss zeitnah wiedergefunden werden können.
8. Alle Aktionen im Archiv, die Veränderungen in der Organisation und Struktur bewirken, sind derart zu protokollieren, dass die Wiederherstellung des ursprünglichen Zustandes möglich ist.
9. Elektronische Archive sind so auszulegen, dass eine Migration auf neue Plattformen, Medien, Softwareversionen und Komponenten ohne Informationsverlust möglich ist.
10. Das System muss dem Anwender die Möglichkeit bieten, die gesetzlichen Bestimmungen (BDSG, HGB/AO etc.) sowie die betrieblichen Bestimmungen des Anwenders hinsichtlich Datensicherheit und Datenschutz über die Lebensdauer des Archivs sicherzustellen. [[Kam97](#)]

2.3 Grundlegende Arten der Archivierung

Am Anfang des Kapitels wurde erläutert, dass der Zweck der elektronischen Archivierung das Auslagern und die langzeitige Aufbewahrung der elektronischen Informationen bzw. Dokumente sind. Deswegen gibt es heutzutage verschiedene Lösungen für die Systeme, wo es massenhaft Daten anfallen. Das sind Systeme wie Email-, Datenbank- und Dokumenten Management Systeme (DMS). Dem entsprechend gibt es die Begriffe Daten-, Dokumenten-, Email- und Datenbankarchivierung.

Grundlegend gibt es zwei Archivierungsarten nämlich Daten- und Dokumentenarchivierung.

Unter der Dokumentenarchivierung wird die Ablage und langfristige Aufbewahrung der Dokumente in einem elektronischen Archivierungssystem verstanden. [[Sch01](#)]

Als Dokument werden hier alle in Papierform liegende Dokumente verstanden, das können Briefe, Faxe, Rechnungen, Lieferscheine, Zeichnungen und andere beliebige Dokumente sein. Heutzutage ist die Dokumentenarchivierung als eine Funktion in Dokumentenmanagementsystemen (DMS) eingebettet. Neben der revisionssicheren Archivierung bieten solche Systeme die Umwandlung von Papier in beliebigen elektronischen Formaten, Indizierung und Volltextsuche auf Dokumenten.

Das Prinzip sowohl bei der Datenarchivierung und auch bei der Datenbankarchivierung ist gleich. In beiden ist das Ziel die Datensätze aus den Datenbanktabellen, die nicht mehr für das Tagesgeschäft relevant sind, auszulagern, um das System zu entlasten. Denn bei jedem

Geschäftsprozess entstehen eine Vielzahl von Daten, die in der Datenbank gespeichert werden. Im Laufe der Zeit kann das Datenbankvolumen so zunehmen, dass die Performanz und Verfügbarkeit des Systems dadurch negativ beeinflusst wird.

Die Datenbankarchivierung wird im Zusammenhang mit den Datenbanksystemen gesprochen, und die Lösungen dafür sind dementsprechend Datenbanksystemabhängig. Es wird sozusagen direkt auf der Datenbank gearbeitet.

In der Datenarchivierung wird kein Unterschied zwischen den Datenbanken gemacht. Denn auf die Datenbank wird über eine Anwendungsschicht zugegriffen. Den Datenbankzugriff übernimmt die in dieser Schicht liegende Anwendung. Deswegen ist es unerheblich, welches Datenbanksystem in der darunterliegenden Schicht eingesetzt wird.

In den bisherigen Abschnitten wurden keine Unterschiede zwischen den Archivierungsarten gemacht. Denn bei allen gelten die gleichen Vorschriften und Richtlinien.

2.4 Ansätze zur elektronischen Archivierung

2.4.1 Strategien zur Langzeitverfügbarkeit

Das Hauptproblem bei der Langzeitarchivierung besteht darin die Daten über einen längeren Zeitraum, mindestens zwischen sechs und 10 Jahren, verfügbar zu halten. Die Hard- und Software befinden sich in einem ständigen Änderungsprozess und sogar die Unternehmensstruktur und die Gesetze ändern sich mit der Zeit. Dadurch werden die Systeme zusammengefasst, Hardware wird geändert oder die Software wird an die neue Technologie angepasst.

Die Archivierung muss aus diesen Gründen konsistent gegenüber den Veränderungen sein. Wie es schon am Anfang der Kapitel erwähnt wurde, ist die Hauptaufgabe der Archivierung die digitalen Dokumente und Informationen unveränderbar über einen langen Zeitraum aufzubewahren und zu jeder Zeit frei zugänglich zu machen.

Zur Lösung des Problems existieren mehrere Strategien, die sich grundlegend voneinander unterscheiden:

Verwendung von Standardformaten

Die Aufbewahrung und die Verfügbarmachung der Dokumente oder Informationen über die vielen Jahre stellt die Archivierungssoftware vor einer schwierigen Entscheidung, die Aus-

wahl des *richtigen* Datenformats. Durch die große Anzahl von Software existiert mittlerweile eine fast unüberschaubare Auswahl an verschiedenen Formaten.

Deswegen sollte bei der Auswahl des *richtigen* Datenformats untersucht werden, wofür die Dokumente oder Daten verwendet werden und welche Arten von Dokumenten bzw. Informationen existieren.

Wichtigstes Kriterium ist dabei die Einhaltung von Standards. Denn durch die rasante technologische Entwicklung sind nicht nur die Hardware sondern auch die Software und das dafür verwendete Dateiformat dem Alterungsprozess ausgesetzt. Daneben benutzen einige Anwendungen zur Datenspeicherung ihre eigenen Dateiformate, welche speziell an diese Programme gebunden sind und von anderen Anwendungen meist nicht verarbeitet werden können, wie z. B. Dateiformate der Microsoft-Office Anwendungen.

Es sollte daher ein Format gewählt werden, welches von einer internationalen Standardisierungsinstitution wie International Organisation of Standardization (ISO), American National Standard for Information Sciences (ANSI) oder das World Wide Web Consortium (W3C), anerkannt ist. Dadurch kann auf Veränderungen effektiver reagiert werden.

Im Folgenden werden kurz die Standarddatenformate aufgeführt, die für die Datenarchivierung relevant sind:

- **ASCII & Co.**

American Standard Code for Information Interchange (ASCII) wurde zum ersten Mal im Jahr 1963 als 7-Bit-Zeichenkodierung entstanden und diente vorerst zur Darstellung der englischen Sprache. ASCII wurde im Jahr 1967 erstmals als Standard veröffentlicht und im Jahr 1986 zuletzt aktualisiert.

Das Problem bei ASCII war, dass es für die englische Sprache konzipiert war. Deswegen wurden im Laufe der Jahre neue länderspezifische ISO-Standardzeichensätze entstanden, bei den als Grundlage der ASCII-Zeichensatz genommen wurden und daher kompatibel zu ASCII sind. Daneben existieren auch systemspezifische Codes. Der Codepage 437 genannte Code war lange Zeit der am weitesten verbreitete, er kam auf dem IBM-PC unter MS-DOS vor. Was noch im Großrechnerbereich verwendet wird, ist das EBCDIC, was inkompatibel zu ASCII Kodierung ist.

Ein anderer Zeichensatz, was auf vielen Systemen und Anwendungen und in den meisten Webseiten als Standard verwendet wird, ist der Unicode und davon speziell UTF-8. Unicode verwendet 32 Bit pro Zeichen und könnte somit mehrere Milliarden verschieden Zeichen unterscheiden. UTF-8 ist dagegen eine 8-Bit-Kodierung von Unicode, die zu ASCII abwärtskompatibel ist.

Bei der Archivierung sollte aufgrund der Kompatibilität das verwendete Zeichensatz beachtet und bekannt gegeben werden. Gerade bei der Migration kann es sein, dass

bei einem neuen System ein anderer Zeichensatz verwendet wird, und entsprechend müssen auch die Daten vorher in das Zeichensatz konvertiert werden.

Comma Separated Values (CSV) ist eine Variante des ASCII-Textes und dient zur Speicherung von tabellarischen Daten. Dabei werden die einzelnen Werte der Tabelle entweder durch Komma, Semikolon, Doppelpunkt, Leerzeichen oder Tabulator getrennt. Ein allgemeiner Standard für das Dateiformat CSV existiert nicht. Es wird jedoch im RFC 4180 grundlegend beschrieben.

Zur Langzeitarchivierung ist das CSV-Format aufgrund seiner Standardisierung und einfachen Konvertierungsmöglichkeiten gut geeignet. Allerdings ist es gegenüber dem anderen Formaten mit Nachteilen verbunden.

- Jede Zeile in der Datei steht für ein Tupel der Tabelle. Das Zeichen zum Zeilenbruch wird aber von Betriebssystem zu Betriebssystem anders dargestellt. So ist die Plattformunabhängigkeit nicht gegeben.
- Welches Zeichen zum Trennen der Werte benutzt wird, sollte vorher vor der Implementierung festgelegt werden.
- Die Angabe von Metainformationen für eine Tabelle ist schwierig.
- Das verwendete Zeichencode ist nicht erkennbar, deswegen muss es festgelegt werden oder den anderen bekannt gegeben werden.
- Es gibt verschiedene Formate für Datum- und Zeit- oder auch für Dezimalzahlen und Zahlen mit Tausendertrennzeichen. Deswegen muss das auch vor der Implementierung festgelegt oder bekanntgegeben werden.

Aus oben genannten Gründen muss es schon vor der Implementierung der Archivierungslösung schriftlich festgelegt werden, wie eine CSV-Datei formatiert wird, weil die auch entsprechend programmiert werden müssen.

- **SGML**

Standard Generalized Markup Language (SGML) ist eine international normierte Metasprache, die die logische Struktur und den Inhalt von elektronischen Dokumenten beschreibt.

Der Vorteil von SGML liegt darin, dass die Dokumentenstruktur und deren Semantik in einem sogenannten DTD (Document Type Definition) vom Anwender separat frei definiert werden kann. Aufgrund der Komplexität der Sprache konnte sich SGML aber nicht durchsetzen.

Dennoch kann SGML aufgrund der Standardisierung für die Langzeitarchivierung benutzt werden. Es gibt aber zurzeit sehr wenige Entwickler, die sich mit SGML gut auskennen oder es muss noch zusätzlich für Schulungen investiert werden.

- **XML**

Extensible Markup Language (XML) ist zurzeit de-facto-Standard unter den Dokumentenbeschreibungssprachen. XML wurde von W3C entwickelt und kann als eine Teilmenge von SGML angesehen werden.

Die Struktur und deren Semantik eines Dokuments kann wie bei SGML in einem DTD oder auch in einem XML-Schema, das selbst in XML-Format liegt, definiert werden.

Heutzutage wird XML fast in allen Bereichen der IT eingesetzt. Der Grund dafür ist die Existenz von zahlreichen XML-Parsern und die einfache Syntax. Die Definition umfasst etwa 30 Seiten im Gegensatz von etwa 500 Seiten bei SGML.

Da XML ein kostenfreies, offenes, standardisiertes und zunehmend weit verbreitetes Format ist, erfüllt es sämtliche Voraussetzungen zur Langzeitarchivierung.

Als Fazit ist für die Datenarchivierung XML zu empfehlen. Gegenüber SGML hat XML sehr hohe Bekanntheit und auch leicht erlernbar. Gegenüber CSV können sehr viele Informationen wie Zeichenkode und Metadaten der Datenbanktabellen schon in der XML-Datei gespeichert werden. Zusätzlich kann die Struktur einer XML-Datei mit DTD oder XML-Schema beschrieben und dadurch die Gültigkeit der Datei überprüft werden, was in CSV nicht möglich ist. Der Programmieraufwand wird auch verringert, weil es schon sehr viele gute XML-Parser auf dem Markt gibt und der Parser nicht neu programmiert werden muss.

Migration

Allgemein wird unter Migration die Umstellung einer Software auf ein neues System oder Konvertierung der bestehenden Formate der Daten in neuere Formate, sowie die Umstellung von Hardware einer alten Technologie in eine neue Technologie verstanden.

Da sich der technologische Fortschritt ständig in Bewegung befindet, muss auch die Migration in periodischen Abständen stattfinden. Dabei muss beachtet werden, dass bei der Migration das Original nicht verändert, oder gelöscht werden. Die Integrität der Daten muss gewährleistet sein.

Bei diesem Transfer von Aufzeichnungen in ein neues System müssen möglicherweise Metadaten anders strukturiert werden oder können nicht mehr erhalten werden. Es entsteht ein mittlerer bis hoher Informationsverlust. Zu beachten ist vor allem auch der enorme Zeitaufwand für Migrationen beim entsprechenden Datenvolumen.

Technikmuseum

Bei diesem Verfahren werden ein Exemplar von den verwendeten Hard- und Softwareumgebungen über die Jahre bereitgehalten. Der Vorteil dabei ist, dass die Originaldokumente oder Daten unverändert erhalten bleiben. Dagegen sprechen aber die enormen Erhaltungskosten, weil nach einigen Jahren immer neue Produkte gekauft werden und der Support für die älteren Maschinen und Programme nicht gewährleistet werden kann.

Emulation

Das Originalsystem wird durch eine eigens zu erstellende Software imitiert. Künftige Computer verhalten sich dann wie die ursprünglich verwendeten und sind in der Lage, die gleichfalls archivierte Software zu lesen. Diese Archivierungsform ist theoretisch am besten geeignet, da sie nicht nur die Wiedergabe der Inhalte, sondern auch deren ursprüngliche Darstellung gewährleisten würde.

Fazit

In der Praxis wird für die Langzeitarchivierung allgemein die Migration bevorzugt und im Bereich SAP ausschließlich die Migration. Aufgrund der zahlreichen Probleme und enormen Erhaltungskosten wird der Technikmuseumsansatz nicht mal angedeutet. Denn es ist wegen des Alterungsprozesses nicht sichergestellt, dass das System nach jahrelanger Lagerung noch funktionsfähig ist und auch Ersatzteile für das System beschafft werden kann. Außerdem wird es schwierig sein Personen zu finden, die das System bedienen können.

Bei der Emulation werden sehr gute Kenntnisse über das System wie interne Funktionsweise und Spezifikationen benötigt. Gerade bei einem gewaltigen System wie SAP ist es fast unmöglich einen Emulator zu schreiben oder es ist mit sehr hohen Kosten verbunden, womit auch die eventuellen Lizenzkosten nicht ausgeschlossen werden können.

Bei beiden Verfahren ist der Vorteil, dass die Daten und Dokumente originalgetreu erhalten bleiben und keine Datenverluste entstehen. Dagegen können bei der Migration wegen der Konvertierung Datenverluste entstehen. Dafür sind die Daten aber immer aktuell und jederzeit zugreifbar. Es werden natürlich auch bei der Migration Kosten entstehen, hängt aber von der Menge der Daten ab, die migriert werden. Daher wird bei der Langzeitarchivierung fast ausschließlich die Migration bevorzugt.

2.4.2 Speichertechnologien

Durch die gesetzliche Vorschriften oder Richtlinien ist es nicht festgelegt, welche spezielle Speichermedien benutzt werden sollen. Die verlangen lediglich, dass das Datenverarbeitungssystem die Unveränderbarkeit des Datenbestandes gewährleisten muss. Es liegt also im Ermessen des Steuerpflichtigen, welche Speichermedien zur Erfüllung der gesetzlichen Anforderungen und der betrieblichen Aufgaben einzusetzen sind. [BIT06]

Bei den Autoren gibt es verschiedene Auffassungen, welche Speichermedien für die revisionssichere Archivierung geeignet oder ungeeignet sind. Wenn nur das einzelne Medium und nicht das gesamte Archivierungssystem betrachtet wird, ist die Antwort eindeutig:

Ungeeignet für die revisionssichere elektronische Archivierung sind die herkömmlichen magnetischen Speichermedien wie z.B. Magnetbänder und Festplatten, die von Betriebssystemen dynamisch verwaltet werden. Bei beiden Typen kann der Inhalt des Datenbestandes geändert und gelöscht werden.[Kam05]

Geeignete Medien für die elektronische revisionssichere Archivierung sind in digital-optische Speichermedien, die in ihrem Laufwerk mit einem Laser berührungsfrei abgetastet und nur einmal beschrieben werden können. Diese Speichertechnologie wird als Hardware WORM („Write Once, Read Multiple Times“) bezeichnet. Die WORM Speichermedien selbst sind durch ihre physikalischen Eigenschaften gegen Veränderungen geschützt. WORM Medien können weder abgeändert, noch überschrieben, noch gelöscht werden.

Weil aber bei der Revisionssicherheit das gesamte System betrachtet wird, können zur Archivierung die Festplattensysteme genauso gut genutzt werden. Das System muss natürlich über entsprechende Mechanismen verfügen, die das Löschen und Veränderung der Daten und den unerlaubten Zugriff verhindern.

Aufgrund der großen Wichtigkeit, muss es noch erwähnt werden, dass bei der elektronischen Langzeitarchivierung egal welches Speichermedium benutzt wird, die konsequente Einplanung und regelmäßige Durchführung von „Refreshing“, also Umkopieren der Daten auf ein neues Medium, unerlässlich ist. Denn bei allen Medien gibt es unterschiedliche Risiken und Gefahren, dass bei langfristigem Aufbewahren auf die Daten nicht zugegriffen werden kann.

3 Analyse

3.1 Datenarchivierung im SAP

Die Basis eines SAP-Systems bildet eine Datenbank, in der die bei jedem Geschäftsvorgang erzeugten Daten gespeichert werden. Über die Jahre vergrößert sich auch das Datenvolumen, was auf die Performance, Ressourcenverbrauch und Verfügbarkeit des Systems auswirkt. In Folge dessen steigen auf der anderen Seite auch die Kosten. Wie es in dem vorigen Kapitel erläutert wurde, dürfen einige Daten wegen der gesetzlichen Aufbewahrungspflicht nicht einfach gelöscht werden.

Aus diesen Gründen sollten daher Daten zu abgeschlossenen Geschäftsvorgängen, die für das Tagesgeschäft nicht mehr relevant sind, aus der Datenbank entfernt werden. Die Daten müssen deshalb so aus der Datenbank ausgelagert und abgelegt werden, dass bei Bedarf jederzeit lesend darauf zugegriffen werden kann. Diese Aufgabe übernimmt die SAP-Datenarchivierung. [\[SAP\]](#)

In diesem Kapitel werden die Technologien vorgestellt, die das SAP-System standardmäßig anbietet:

3.1.1 ADK - Archive Development Kit

Das ADK stellt die Laufzeitumgebung für die Datenarchivierung bereit und bildet die Grundlage für die Erstellung von Datenarchivierungslösungen. Es bietet Schnittstellen zwischen den Anwendungen, der Datenbank und dem Archiv, in dem die Daten abgelegt sind.

Im Mittelpunkt der SAP-Datenarchivierung steht das Archivierungsobjekt. In ihm wird festgelegt, was und wie archiviert wird. Konkret bedeutet das, welche Programme für die Archivierung benötigt werden, also die Programme zum Erzeugen und Rückladen des Archivs, zum Löschen der Datenbanktabellen und eventuell auch zum Vor- und Nachlauf. Die Angabe der zu archivierenden Datenbanktabellen können nicht gemacht werden. Die müssen im Programmcode angegeben werden.

Zur Entwicklung der Programme und zur Verwaltung stellt das ADK zwei Funktionsbereiche:

- Archivadministration
- Anwendungsprogrammierschnittstelle (API) zur Entwicklung eigener Archivierungsprogramme

Von der Archivadministration aus ist es möglich sowohl die zu einem Archivierungsobjekt gehörenden Archivierungsprogramme wie Schreib-, Lösch-, Lese- und Rückladeprogramme auszuführen bzw. als Hintergrundjob einzuplanen als auch Customizing-Einstellungen vorzunehmen und in die Verwaltung zu wechseln. Darüber hinaus können über die Archivadministration Archivierungsläufe verwaltet und Statistiken zu den Archivierungsläufen angesehen werden. [Boh05]

Nach der Definition des Archivierungsobjektes erfolgt die Implementation der für das Archivieren notwendigen Programme mindestens zum Schreiben, Löschen und Rückladen. Dafür bietet das ADK eine Anwendungsprogrammierschnittstelle(API), in der die benötigten Programme und Funktionsbausteine definiert sind. Die Aufgabe des Entwicklers besteht darin die richtigen Daten zu selektieren und zu einem betriebswirtschaftlich abgeschlossenen Objekt zusammenzufassen und es an ADK zum Archivieren weiterzugeben. Deswegen muss derjenige sehr gute Kenntnisse über die Semantik der Daten der jeweiligen Anwendung, für die ein Archivierungsprogramm geschrieben wird, haben.

3.1.2 XML-Archivierung

Die andere Technik, die auch in den neuen SAP-Systemen mit dem SAP Web Application Server 6.40 (SAP Web AS) ausgeliefert wird, ist die XML-Archivierung.

Das Herzstück bildet hier wieder das Archivierungsobjekt, das mit dem gleichen Programm wie bei Archivierungsobjekten für ADK definiert und verwaltet wird. Der einzige Unterschied bei der Definition ist, dass diese Objekte als XML-Archivierungsobjekte gekennzeichnet werden.

Der technische Aufbau und die Funktionsweise der XML-Archivierung ist ganz anders als bei ADK. Die Abbildung 3.1 zeigt die beteiligten Komponenten der XML-Archivierung.

Der größte Unterschied zum ADK ist, während ADK im Anwendungssystem eingebettet ist, hat man bei der XML-Archivierung zwei logisch getrennte Systeme, die in der Regel auch physisch getrennt sein können. Das eine ist das Anwendungssystem, dessen Datenbank entlastet werden soll bzw. die zu archivierenden Daten bereitstellt und das archivierende System, in dem sogenannte XML-Datenarchivierungsservice, kurz XML-DAS, läuft.

Das XML-DAS ist ein Teil der SAP J2EE Engine, wo die eigentliche Archivierung und die Administration der XML-Archivierung stattfindet. Der Entwickler bzw. Administrator muss zur

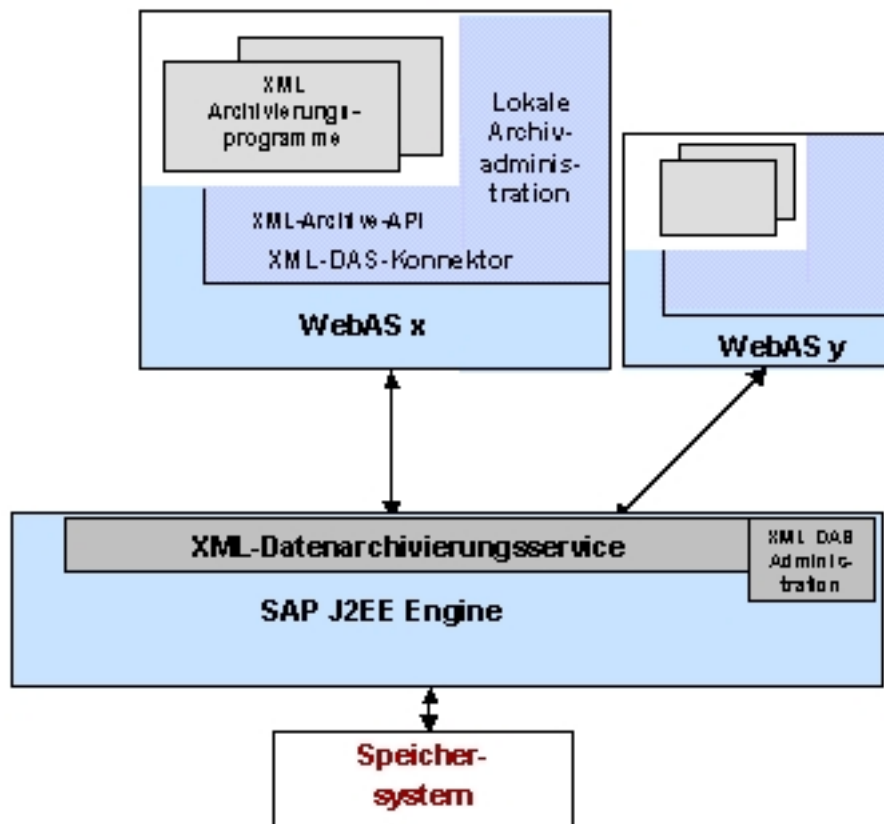


Abbildung 3.1: Komponenten der XML-Archivierung (Quelle: SAP)

Administration beide Umgebungen beherrschen, einerseits die, die zur ADK-Administration benutzt wird und andererseits das XML-DAS.

Die Entwicklung und die Ausführung der Archivierungsprogramme finden wie bei ADK auf dem Anwendungssystem statt. Die Kommunikation zwischen dem Anwendungssystem und dem XML-DAS findet per HTTP(s)-Protokoll mittels XML-DAS-Konnektor statt.

Der Vorteil bei der Technologie ist der Einsatz von Standards wie JAVA und XML. Außerdem können jetzt auch JAVA-Anwendungen mit archiviert werden, im Gegensatz zu ADK.

3.1.3 Fazit

Beide Technologien haben ihre Vor- und Nachteile. Deswegen sollte es, bevor ein Archivierungssystem entwickelt wird, analysiert werden, was und welche Daten archiviert werden und auf welchen SAP-Systemen die Archivierungslösung eingesetzt wird.

Der Hauptunterschied ist, dass die XML-Archivierung relativ neu ist und deswegen auf älteren SAP-Systemen nicht existiert. Ab dem SAP Web Application Server 6.4 gibt's die XML-Archivierung. Gerade bei Großunternehmen dauert es aber über Jahrzehnten, bis das ältere System auf einem neuen migriert wird.

Auf dem Markt gibt es sehr viele Archivierungsprodukte, die auf ADK basieren, aber nur ein paar, die mit der neuen Technologie implementiert sind. Deswegen wird es sehr schwierig sein, jemanden zu finden, der sich mit der Technologie auskennt. Außer der SAP-Hilfe gibt es zurzeit keine Dokumentation und keine Beschreibung über die XML-Archivierung, so dass man bei Problemen auf sich allein gestellt ist.

Gegenüber dem ADK ist die XML-Archivierung mit sehr vielem Administrationsaufwand verbunden. Denn außer dem Anwendungssystem muss auch das JAVA-Engine eingerichtet und gewartet werden, was bei vielen Kunden nicht der Fall ist. Bei Problemen wird es schwierig sein, die Fehlerquelle zu finden, weil ja beide Systeme überprüft werden muss.

Außer einer Marketingstrategie sehe ich noch keine Notwendigkeit eine Archivierungslösung in dieser neuen Technologie zu implementieren, außer man hat spezielle JAVA-Anwendungen, die man archivieren will. Weiterhin ist es in ADK genauso leicht die Daten als XML zu archivieren.

3.2 Beschreibung des vorhandenen Systems

3.2.1 STS - Standard Trafoschicht

STS ist der Bestandteil der paricon Data Management Suite, das neben STS noch Multi Extraction Framework (MXF) und Validation Framework (VF) beinhaltet. Als Datenversorungskomponente entstand die Lösung als Ergebnis einer Zusammenarbeit zwischen der DZ BANK und der Paricon AG. Die STS ist von SAP zertifiziert, als Standardlösung neutral aufgebaut und läuft auf allen SAP Lösungen ab 6.20 Basis.

Wie in der Abbildung 3.2 zu sehen ist, stellt ein Bindeglied zwischen Vorkomponente oder Middleware und den von SAP zur Verfügung gestellten Schnittstellen dar.

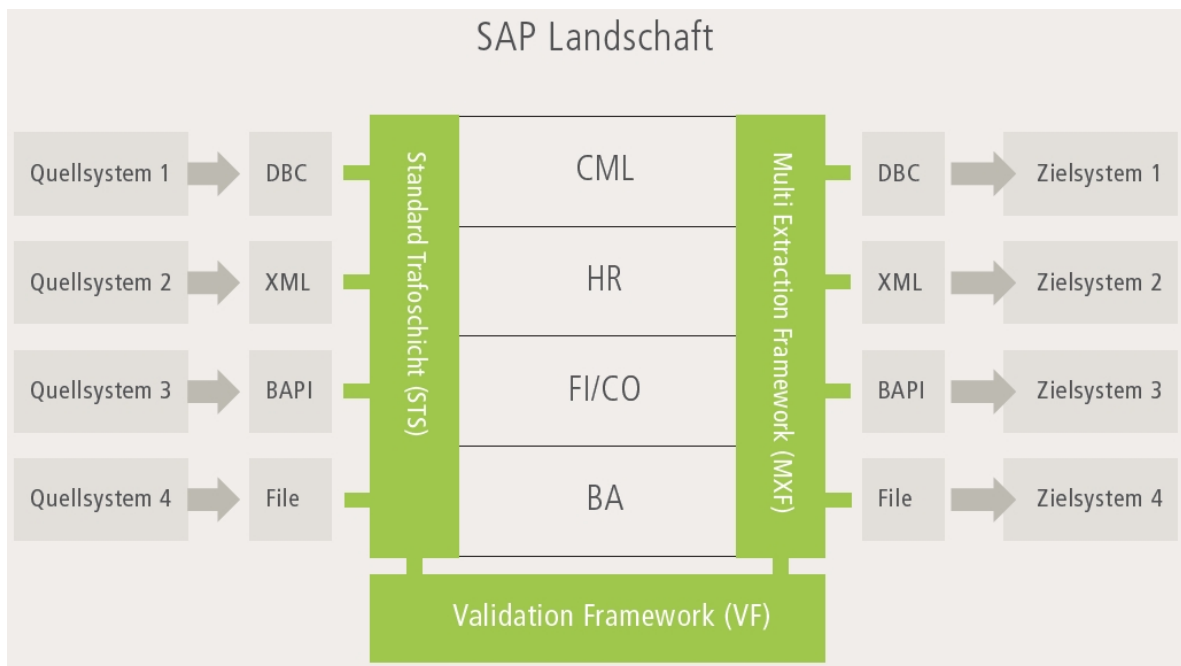


Abbildung 3.2: Paricon Data Management Suite

Der Ablauf der Datenübergabe erfolgt folgendermaßen, beginnend mit einem Vorverarbeitungslauf, in dem die Daten grundsätzlich für die Migration vorbereitet und angepasst werden, erfolgt nach der Pufferung der Daten die eigentliche Verarbeitung. In der parallelisierten Verarbeitung werden die Daten plausibilisiert, ggf. korrigiert. Abschließend wird jeder Datensatz im Zielsystem zuverlässig verbucht. Funktioniert dies nicht, ist eine automatische Nachbuchung oder eine manuelle Nachbuchung zu einem späteren Zeitpunkt möglich. Die Abbildung 3.3 veranschaulicht den Ablauf.

3.2.2 STS-Archivierung

Bei der Datenvorbereitung und Verbuchung entstehen in STS an verschiedenen Stellen (vgl. Abbildung 3.3) Daten, die wiederum in Form von Datenbanktabellen gespeichert werden. Man kann sich vorstellen, dass die Anzahl der Datensätze in einer Bank im Millionenbereich liegen, und an Datenvolumen liegt die Größe im Gigabyte- und kann sogar im Terabytebereich liegen. Aufgrund gesetzlicher Bestimmungen und Richtlinien dürfen die Daten auch im STS nicht einfach gelöscht werden. Denn der Steuerprüfer muss später nachvollziehen können, wie die Daten in dem neuen System entstanden sind.

Im Laufe der Zeit kam die Anforderung von Helaba (Landesbank Hessen-Thüringen), bei den auch STS eingesetzt wurde, eine Archivierungslösung dafür zu entwickeln. So entstand



Abbildung 3.3: Ablauf der STS-Datenversorgung

die STS-Archivierung als Add-On-Lösung. Aufgrund der Systemkompatibilität mit den SAP-Systemen der Bank basiert sie auf ADK.

Die STS stellt mit dem Archivierungs-Add-on sicher, dass alle relevanten Daten der Datenversorgungsprozesse revisionssicher archiviert und rekonstruiert werden können. Sie stellt sicher, dass nur archivierte Läufe tatsächlich physikalisch gelöscht werden und damit die Datensicherheit gewährleistet ist.

Archivierungsobjekte

Weil auch die neu zu entwickelnde allgemeine Archivierungslösung kompatibel mit STS-Archivierung sein soll, ist es wichtig zu untersuchen, welche Archivierungsobjekte für STS gibt.

Laut der Funktionsbeschreibung kann STS in drei Bereiche unterteilt werden, bei den auch bei laufendem Betrieb oder vorher Daten erzeugt werden. Die Daten werden in den Datenbanktabellen gespeichert, die für den jeweiligen Bereich konzipiert wurden:

1. Customizing
2. Laufinformationen
3. Content

Im Customizing-Bereich befinden sich alle erforderlichen Einstellungen samt Produkteinstellungen, die für einen reibungslosen Betrieb der STS benötigt werden.

Im Laufinformationen-Bereich sind alle STS-Laufinformationen gespeichert. Die Informationen geben einen Überblick darüber, zu welchem Produkt gehören die Eingangsdaten,

welche Eingangsdaten überhaupt verarbeitet wurden, sowie Protokolldaten des gesamten Ablaufs der Verarbeitung von Eingangsdaten.

Im Content-Bereich werden die eigentlichen Eingangsdaten gehalten. Mögliche Fehler der Daten und deren Verarbeitung werden in diesem Bereich protokolliert bzw. korrigiert.

Entsprechend der Funktionsbereiche existieren auch drei Archivierungsobjekte, zu den die entsprechenden Datenbanktabellen zugeordnet sind:

1. `_STS_ARC1` - Archivierungsobjekt für Customizing-Tabellen
2. `_STS_ARC2` - Archivierungsobjekt für Laufinfo-Tabellen
3. `_STS_ARC3` - Archivierungsobjekt für Content-Tabellen

Datenformat

Zum Speichern der Tabellen hat man sich für XML entschieden. Denn die Struktur und Inhalte sind sehr gut mit Hilfe der Beschreibungssprache XML beschreibbar. SAP bietet eine ganz einfache Methode um eine Struktur oder Tabelle in XML umzuwandeln und wieder zurückzuwandeln.

In STS-Archivierung werden die selektierten Daten vorher in eine Struktur gepackt. Diese Struktur enthält Daten wie Mandant, Objektschlüssel und die eigentliche XML-Darstellung des Objekts. Danach wird die Transformation „ID“ aufgerufen. Als Ergebnis hat man ein String als XML. Dieser String wird dann zum Archivieren an ADK weitergegeben.

3.3 Gründe für eine Neuentwicklung

In dem vorigen Kapitel wurde eine Lösung vorgestellt, die im produktiven Betrieb eingesetzt und auch weiter noch von Paricon Products GmbH weiterentwickelt und gewartet wird. Allerdings wie schon erwähnt, ist die Anwendung in seinem Umfeld beschränkt einsetzbar, und daher alleinige Vermarktung ist nicht möglich.

Bevor auf die technischen Unterschiede eingegangen wird, sollte auch die Hintergründe untersucht werden, warum ein Unternehmen die Neuentwicklung vorzieht, als ein fertiges Produkt zu kaufen.

Paricon Products GmbH ist ein Softwareunternehmen, das sein Kerngeschäft Entwicklung und Vermarktung der Produkte im SAP-Bereich ist. Das Unternehmen hat sich Anfang 2009 von Paricon AG losgelöst. Neben der Produktentwicklung zählte damals auch die Unternehmensberatung zum Kerngeschäft und so konnten sich die Umsätze der beiden Bereiche gegenseitig kompensieren. Jetzt ist das Unternehmen aber abhängig von der Anzahl der Installationen der eigenen Produkte und deren Wartung.

Ein Unternehmen, das nur vom Vertrieb der eigenen Produkte lebt, muss immer neue Innovationen auf den Markt bringen, damit es sich im Markt etablieren zu können. Eines davon ist eine Archivierungslösung. Mit der STS-Archivierung waren die Grundlagen da und für den eigenen Produkt STS waren die Anforderungen erfüllt. Aber für die anderen Produkte war es nicht möglich, die STS-Archivierung auf den einzusetzen. Die Lösung war viel zu spezifisch.

Im Hinblick auf die Zukunft sollte die Lösung auch für die Fremdprodukte und Module in SAP eingesetzt werden. Das heißt, die Lösung sollte alleine vertrieben werden. Wie die STS-Archivierung sollte erst mal nur die Tabellen archiviert werden. Für die Zukunft ist aber schon möglich aufbauend auf diese Lösung auch Dokumente zu archivieren.

3.3.1 Marktanalyse

Bevor ein Produkt neu entwickelt wird, sollte auch der Markt untersucht werden, ob schon ähnliche Produkte gibt und welchen Funktionsumfang sie haben. Außerdem sollten Stärken und Schwächen der jeweiligen Produkte untersucht werden, um überhaupt ein konkurrenzfähiges Produkt zu entwickeln.

Auf dem Markt gibt es zurzeit unzählige Produkte für SAP-Bereich und nicht-SAP-Bereich, dann speziell Daten- und Dokumentenarchivierung und zuletzt im Bereich der E-Mail-Archivierung. Daher wurden nur die SAP-Produkte untersucht und darunter die sich in die Datenarchivierung spezialisiert haben.

Nach der Untersuchung wurde aber festgestellt, dass es keine reine Datenarchivierungslösung für SAP gibt, sondern eine Kombination von Dokumenten- und Datenarchivierung. Dies könnte folgende Ursachen haben:

- Die Dokumente wie Rechnungen, Lieferscheine oder ähnliche beanspruchen mehr Speicherplatz als reine Daten. Deswegen ist der Bedarf zum Auslagern solcher Dokumente größer als Auslagern von Tabellen.
- SAP bietet für die eigenen Anwendungen schon standardmäßig Programme zur Datenarchivierung. Allerdings sind die allgemeinen Lösungen für den eigenen Betrieb teilweise nicht brauchbar. Deswegen müssen externe Berater und Entwickler herangezogen werden und für Beratungsfirmen bringt das richtig viel Geld. Aus diesem Grund gibt es unzählige Beratungshäuser, die den Service anbieten.

3.4 Anforderungsanalyse

Ausgehend von der STS–Archivierung und der Marktanalyse werden in diesem Kapitel die funktionalen- und nichtfunktionalen Anforderungen aufgestellt. Dabei wurden alle Anforderungen berücksichtigt, die ein fertiges Produkt erfüllen sollte.

3.4.1 Funktionale Anforderungen

- Archivieren von Daten:
 - Das System soll beliebige Anwendungen im SAP-System archivieren können.
 - Für jede Anwendung soll man beliebig viele Anzahl von Datenbanktabellen definieren können.
 - Das System soll für eine Anwendung Daten von mehreren Mandanten archivieren können.
 - Das System soll neben Datenbanktabellen auch generierte Tabellen archivieren können. Generierte Tabellen sind die, die eine Anwendung zur Laufzeit generiert. Deswegen kann die Tabelle beliebig heißen und beliebige Struktur haben. Dafür sollte das System eine Schnittstelle anbieten, über die dann eine Anwendung die generierten Tabellen und deren Daten übergeben kann.
 - Man muss verschiedene Tabellentypen definieren und die Tabellen diesen Typen zuordnen können.

-
- Bei den Typen muss man wenigstens archivierbar, löscher und rückladbar auswählen können.
 - Die Archivierung soll als Hintergrundjob und manuell gestartet werden können.
 - Man muss auswählen können, welche Anwendungen man archivieren will.
 - Man muss selektieren können, welche Daten man archivieren will.
 - Die Tabellen sollen als XML-Struktur gespeichert werden.
 - Rückladen der Archivierung:
 - Das System soll die archivierten Daten zurückladen können.
 - Das System soll die Möglichkeit bieten, die gesamten Daten oder nur einzelne Tabellen oder Objekte von einem Archivierungslauf zurückzuladen.
 - In Bezug auf den vorigen Punkt soll die Integrität der Daten gewährleistet sein. Wenn eine Abhängigkeit der Daten zwischen den Tabellen gibt, sollen die abhängigen Daten automatisch zurückgeladen werden.
 - Anzeigen der Archivierung:
 - Das System soll die archivierten Daten anzeigen können. Dabei werden die Daten extern in einem Browser angezeigt.
 - Das System soll zu einem Archivierungslauf bestimmte Selektionskriterien anbieten, mit den der Benutzer bestimmte Daten zum Anzeigen auswählen kann.
 - Löschen der archivierten Daten:
 - Das System soll die Daten bei erfolgreicher Archivierung löschen.
 - Protokollierung:
 - Das System muss alle Archivierungsläufe mit Startzeit, Endzeit, User und Status anzeigen können.
 - Konfigurierbarkeit (Customizing)
 - Das System muss Möglichkeiten bieten es individuell zu konfigurieren.

3.4.2 Nichtfunktionale Anforderungen

- Wartbarkeit
 - Analysierbarkeit
 - * Bei der Programmierung müssen die vorgegebenen Programmierrichtlinien eingehalten werden.
 - * Die zusammengehörigen Teile des Systems müssen in Paketen zusammengefasst werden.
 - Modifizierbarkeit
 - * Das System soll modular aufgebaut werden, so dass man ohne großen Eingriff in den bestehenden Code neue Funktionalitäten hinzufügen kann.
- Technische Dokumentation
 - Die Dokumentation wird in SAP erstellt.
 - Alle erstellten Module müssen dokumentiert werden, darunter Klassen und dessen Methoden, Funktionsbausteine und Programme.
 - Darüber hinaus muss wenigstens eine Klassendiagramm und ein ERD-Diagramm existieren, die alle verwendeten Klassen und Tabellen beinhalten.

4 Design

Die STS–Archivierung und auch ADK dient in diesem Kapitel als Grundlage. Daher werden die Abläufe an vielen Stellen gleich sein, lediglich der Programmcode wird anders aussehen, weil die neue Lösung ja allgemein gültig sein muss. So wird es keine statischen Definitionen mehr für die Datenbanktabellen geben, sofern auf dynamische Programmiermethoden zugegriffen wird oder voll statisch sein, wenn die Programme generiert werden.

4.1 Allgemeine Designansätze

Wie der Titel dieser Bachelorarbeit sagt, die zu entwickelnde Anwendung muss generisch sein. Das heißt, die Anwendung sollte alle Arten von Tabellen archivieren können, ohne den Code ändern und neu kompilieren zu müssen.

Mit der Programmiersprache ABAP bietet SAP schon Techniken, die diese Anforderung erfüllen. Die eine Möglichkeit ist die dynamische Programmierung und die andere Programmgenerierung.

4.1.1 Dynamische Programmierung

Es soll hier nicht die ganze Syntax und alle Sprachelemente der Sprache ABAP erklärt werden, das würde auch nicht möglich sein, sondern die wichtigsten Sprachelemente für die dynamische Programmierung erläutert werden.

- **Interne Tabellen**

Interne Tabellen sind ein Mittel, um dynamische Datenmengen im Arbeitsspeicher von ABAP abzuspeichern. Dabei werden die Daten zeilenweise im Speicher abgelegt, wobei jede Zeile die gleiche Struktur hat. Interne Tabellen bieten in ABAP die Funktionalität von Arrays. Ein wichtiges Einsatzgebiet ist z.B. die programminterne Speicherung und Aufbereitung von Inhalten aus Datenbanktabellen.

- **Feldsymbole**
Feldsymbole sind Platzhalter bzw. symbolische Namen für bestehende Felder. Ein Feldsymbol reserviert keinen physischen Platz für ein Feld, sondern zeigt auf ein Datenobjekt. Feldsymbole sind vergleichbar mit dereferenzierten Zeigern in der Programmiersprache C (d. h. Zeiger auf die der Inhaltsoperator * angewendet wird).
- **Datenreferenzen: REF TO DATA REF TO DATA** ermöglicht Deklaration von Datenreferenzvariablen. Somit können später mit der Referenzvariable ein Datenobjekt dynamisch erzeugt oder eine Referenz auf ein Datenobjekt beschafft werden.
- **Generische Datentypen: ANY, TABLE**
Die generischen Datentypen ANY und TABLE werden bei Feldsymbolen benutzt. Bei ANY werden alle Typen von Datenobjekten akzeptiert. Das Feldsymbol übernimmt alle Attribute des Datenobjekts. Die Angabe TABLE besagt, dass das Datenobjekt eine interne Standardtabelle ist.
- **RTTS - Run Time Type Services**
Die RTTS werden durch eine Hierarchie von Typklassen realisiert, die Methoden für RTTC (Run Time Type Creation) und RTTI (Run Time Type Identification) enthalten. Mit Hilfe dieser Systemklassen ist es möglich,
 - Typinformationen vorhandener Instanzen und Typnamen des ABAP-Typsystems zur Laufzeit zu bestimmen,
 - neue Datentypen zur Laufzeit zu definieren.

Es wurde hier bewusst keine konkrete Syntax und Beispiele angegeben, weil die Benutzung dieser Techniken sehr komplex und schwierig sind. Sie würden den Rahmen hier sprengen.

4.1.2 Programmgenerierung

Neben der statischen Programmentwicklung bietet ABAP Anweisungen zum Generieren von Programmen. Auf der einen Seite bietet diese Art der Programmentwicklung zwar eine leistungsstarke und flexible Möglichkeit der Programmierung, auf der anderen Seite ist diese Art der Programmerstellung äußerst aufwändig. Aus diesem Grund wird empfohlen, in Anwendungsprogrammen erst die anderen Möglichkeiten der dynamischen Programmierung auszuschöpfen, bevor die Programmgenerierung verwendet wird.

Außerdem werden unter ABAP-Referenz oder SAP-Hilfe nicht alle Möglichkeiten aufgezählt, sondern nur einige davon. Alle Anwendungen unter SAP und selbst die Entwicklungsumgebung sind in ABAP geschrieben und können debuggt werden. So können zum Beispiel

rausgefunden werden, welche Funktionen oder Programme bei der Erzeugung von Klassen, Datenbanktabellen, Funktionsbausteine oder Dictionary–Objekten ausgeführt werden.

Bei dieser Methode sind die Entwickler gefordert, lesbaren Code zu schreiben. Denn der generierte Code wird später nicht gepflegt und gewartet, sondern nur der Code, also der Generator, wird gewartet und weiterentwickelt. Das führt dazu, dass auf die Strukturierung und den Programmierrichtlinien nicht beachtet wird und der generierte Code am Ende schwer lesbar ist.

Im Falle der Archivierung gibt es zwei Schritte, bevor die Anwendung ausgeführt werden kann. Der erste Schritt ist die Programmgenerierung und der zweite die Programmausführung. Jedesmal wenn in Customizing etwas geändert wird, müssen alle Programme neu generiert werden. Da der Benutzer das vergessen kann, sollte die Generierung nach dem Speichern der Einstellungen automatisch ausgeführt werden. Denn die alten Programme werden dabei gelöscht und zur gleichen Zeit kann ein Archivierungsjob im Hintergrund ausgeführt werden. Deswegen muss dieser Fall besonders berücksichtigt werden.

4.2 Ablauf der Archivierung

Die Abbildung 4.1 zeigt den grundsätzlichen Ablauf der Archivierung. Dabei kann ein Archivierungsablauf grob in vier Schritten unterteilt werden.

1. Vorlauf (optional):

Laut der SAP–Dokumentation erfordern nur einige Archivierungsobjekte den Vorlauf. Daher ist er optional. In diesem Schritt sollen die Daten für die Archivierung vorbereitet werden, z.B. indem es ein Archivierungskennzeichen gesetzt wird. In diesem Schritt werden die Daten nicht gelöscht. Deswegen sollte dieser Schritt weiter untersucht werden, ob der wirklich notwendig ist. Daher kann der in der Realisierung erst mal weggelassen werden:

2. Schreiben:

Der aufwendigste Schritt ist das Schreiben und sollte auch in weitere Schritte unterteilt werden.

- a) Es erfolgt zuerst Selektion der zu archivierenden Daten,
- b) in dem nächsten Schritt werden die Daten auf Archivierbarkeit überprüft,
- c) danach folgt die Umwandlung der Daten in das gewünschte Format,
- d) erst dann erfolgt das Erzeugen der Archivdatei,
- e) und zuallerletzt das Schreiben der Daten in das Archiv

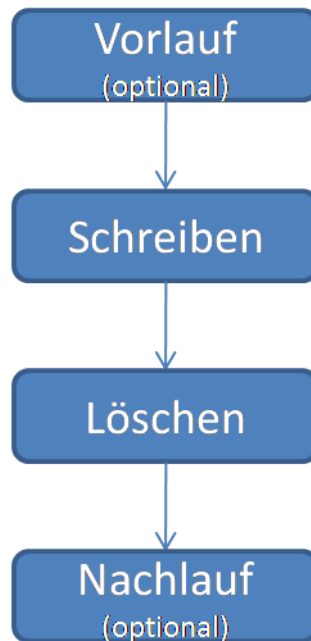


Abbildung 4.1: Ablauf der Archivierung

Zu diesem Zeitpunkt erfolgt jedoch noch kein Löschen der Daten aus der Datenbank.

3. Löschen:

Nach dem erfolgreichen Schreiben folgt das Löschen der Datenbank. Normalerweise wird diese Aktion in diesem Schritt ausgeführt. Aber es kann auch sein, dass die archivierten Daten in der Datenbank nur ein Löschkennzeichen erhalten.

4. Nachlauf (optional):

Wie der Vorlauf ist dieser Schritt auch optional. Falls die Daten in der Löschphase noch nicht aus der Datenbank gelöscht worden sind, geschieht dies durch den Nachlauf. Ausserdem werden die Sperren in diesem Schritt wieder aufgehoben. Die genannten Aufgaben können eigentlich zum Löschen verlagert werden. Deswegen wird dieser Schritt bei der Realisierung erst mal weggelassen.

4.3 Archivierungsobjekte

Im Kapitel Analyse wurde die Wichtigkeit der Archivierungsobjekte erläutert und darüberhinaus wurden auf die Archivierungsobjekte der STS-Archivierung eingegangen. Dabei wurde

auch erklärt, warum für eine Anwendung mehrere Archivierungsobjekte notwendig sind. Daher wird in diesem Kapitel die allgemeine Vorgehensweise bei der Definition von Archivierungsobjekten erläutert.

Bevor die Archivierungsobjekte für eine Anwendung definiert werden, sind genaue Kenntnisse über die Daten erforderlich. Für die Anwendung muss untersucht werden, wo die Daten erzeugt werden, welche Datenbanktabellen es gibt und inwieweit die untereinander zusammenhängen. Danach erfolgt die Zusammenfassung der Tabellen, wie in der STS-Archivierung, nach Arten. Erst danach erfolgt die Definitionen der Archivierungsobjekte. Folgende Gruppierung kann vorgeschlagen werden:

- Customizingtabellen
- Laufinfotabellen
- Datentabellen
- Generierte Tabellen

Die Tabellenarten können natürlich je nach Anwendung anders sein. Nicht jede Anwendung generiert zum Beispiel Datenbanktabellen zur Laufzeit. An dieser Stelle kann eine Frage aufkommen, warum die Customizingtabellen überhaupt archiviert werden sollen. Gerade bei solchen Anwendungen wie bei STS von Paricon, wo die Datenkonvertierungen stattfinden, wichtig zu wissen oder überhaupt später der Steuerprüfer nachvollziehen kann, wie die neuen Daten entstanden sind. Es empfiehlt sich daher in solchen Fällen zusammen mit den normalen Daten auch die Tabellen vom Customizing zu archivieren.

In der Analysephase wurde auch erklärt, dass für jedes Objekt mindestens ein Schreib-, Löscho- und Rückladeprogramm angegeben werden muss. Weil die zu entwickelnde Archivierungsanwendung eine allgemeingültige Lösung sein wird, können bei allen Archivierungsobjekten die gleichen Programme für Schreiben, Löschen und Rückladen angegeben werden. Diese Programme werden eigentlich nur für ADK benötigt und für die neue Archivierungsanwendung sind die nicht relevant. Aber damit die Archivierung auch über ADK funktioniert, sollte die neue Anwendung Schnittstellen dafür bieten.

4.4 Datenmodell

Im Datenmodell wurden auch die Beziehungen zu den SAP-eigene Tabellen angegeben. Die Tabellen von ADK sind mit (SAP) gekennzeichnet. Zur besseren Übersicht wurden die Nicht-Schlüssel-Attribute bei diesen Tabellen weggelassen und nur die Primär- und Fremdschlüssel angegeben.

Das Archive Development Kit (ADK) von SAP ist so ausgelegt, dass es dort keine Mandantenabhängigkeit definiert ist. Alle Einstellungen, die dort betätigt werden, gelten mandantenübergreifend. Daher existiert in den SAP-Tabellen, die in dem Datenmodell angegeben sind, das Feld *MANDT* nicht.

Die zu entwickelnde Anwendung wird mandantenabhängig sein, da die zu archivierenden Tabellen durch generierte Tabellen unterschiedlich sein kann. Nicht alle Tabelleninhalte sind für alle Mandanten gleich wichtig, so dass bestimmte Daten nicht auf allen Mandaten gleich behandelt werden sollen.

In der Anwendung werden grundsätzlich zwei Arten von Tabellen unterschieden, Customizing- und Laufinfotabellen. Zusätzlich gibt es noch Texttabellen, die zur Mehrsprachenfähigkeit dienen.

In den Customizingtabellen werden die Einstellungen festgelegt, die für den Anwendungslauf benötigt werden. z.B. es kann festgelegt werden, für welche Anwendung welche Tabellen man archivieren möchte und über welchen Felder man die Daten selektieren möchte. Die Customizingtabellen sind:

- /ARC/APPS
- /ARC/APP_SELEC
- /ARC/ARCH_OBJS
- /ARC/TABLES
- /ARC/TAB_SELEC
- /ARC/TAB_TYPES
- ARCH_OBJ (SAP)

Die Laufinfotabellen dienen dazu die zur Laufzeit erzeugten Daten zu speichern, insbesondere die Protokolleinträge oder Informationen zu dem Archivierungslauf wie Status oder die von SAP-System erzeugten Archivierungslaufnummern. Die Laufinfotabellen sind:

- /ARC/JOBS
- /ARC/LOGS
- /ARC/RUNS
- /ARC/RUNS_ADK
- /ARC/ARCH_INDEX
- ADMI_RUN (SAP)

- ADMI_JOBS (SAP)

SAP ist ein mehrsprachenfähiges System. Durch die Definition von Texttabellen kann die eigene Anwendung im SAP-System auch mehrsprachig sein. Die Texttabellen sind:

- /ARC/APPS_T
- /ARC/TAB_TYPES_T

4.4.1 Customizingtabellen

/ARC/APPS Das ist die Haupttabelle der Anwendung. Hier wird der Name der Anwendung, die archiviert werden soll und eine Starttabelle, mit der die Archivierungsanwendung mit der Selektion beginnt, angegeben. Das sagt aber nicht aus, ob diese Tabelle auch archiviert wird oder nicht. Das wird in der Tabelle */ARC/TABLES* festgelegt. Im Gegensatz zu den Tabellen in */ARC/TABLES* kann die Starttabelle auch ein Datenbankview sein. Die Views können nur gelesen, aber nicht geschrieben oder gelöscht werden.

/ARC/APP_SELEC In der Tabelle gibt man die Felder von der Starttabelle an. Später kann der Endbenutzer für diese eingegebene Felder Selektionskriterien angeben und anhand dieser Kriterien werden dann die Daten von der Starttabelle selektiert. Die einzelnen Tupel bestimmen, welche Daten von den anderen Tabellen, die in */ARC/TABLES* definiert sind, selektiert und archiviert werden.

/ARC/ARCH_OBJS Für eine zu archivierende Anwendung kann man hier mehrere Archivierungsobjekte angeben. Die Tabelle hat eine n:1 Beziehung zu der ADK-Tabelle *ARCH_OBJ*. Deswegen müssen diese Archivierungsobjekte schon über ADK-Anwendung definiert worden sein.

/ARC/TAB_TYPES Hier können verschiedene Typen für die Tabellen definiert werden und über die Flags angeben, ob bei dem Typ eine Tabelle archivierbar, zurückladbar, löscherbar oder generiert ist. Zum Beispiel sollen zu einer Anwendung häufig die Customizingtabellen mitgespeichert, aber nicht gelöscht werden.

/ARC/TABLES In dieser Tabelle werden für jedes Archivierungsobjekt die Tabellen angegeben, die archiviert werden sollen. Außerdem müssen die Tabellen zu einem Typ, das in der Tabelle */ARC/TAB_TYPES* definiert ist, zugeordnet werden. Wenn die Starttabelle aus */ARC/APPS* auch archiviert werden soll, dann wird die auch hier eingetragen.

Zusätzlich existiert noch ein Feld *flg_starttable*. Das Feld bestimmt, ob die eingegebene Tabelle als Schlüsseltable benutzt wird. Die Felder zu der Tabelle werden in */ARC/TAB_SELEC* über das Feld *flg_keyfield* bestimmt. Die Einträge dienen zur

Schlüsselfindung. Nach diesen Schlüssel können Einzelsätze oder Pakete selektiert werden.

/ARC/TAB_SELEC Wenn Beziehungen zwischen den zu archivierenden Tabellen existieren, werden die hier festgelegt. Man muss natürlich sehr genaue Kenntnisse über das Datenmodell der zu archivierender Anwendung haben. Anhand der Angaben in dieser Tabelle werden die Daten selektiert und archiviert. Bei fehlerhaften Angaben kann es sein, dass entweder nicht die gewünschten oder keine oder aber auch alle Daten einer Tabelle archiviert werden. Folgende Besonderheiten muss man bei der Eingabe noch beachten:

- Wenn eine Tabelle eine Beziehung zu der Startabelle der Anwendung hat, trägt man in die Primärschlüsselfelder das Archivierungsobjekt, den Tabellennamen und das entsprechende Feld der jeweiligen Tabelle, aktiviert das Feld *flg_field_start* also gibt ein „X“ ein und gibt den Feldnamen von der Startabelle in das Feld *ref_field* ein. SAP kennt kein Boolean als Datentyp, es wird immer ein Char1-Feld mit einem großen „X,“ für true verwendet.
- Wenn für ein bestimmtes Feld Index aufgebaut werden soll, wird einfach nach der Eingabe der benötigten Daten in die Primärschlüsselfelder das „X“ in das Feld *flg_indexfield* eingegeben. Wenn aber für das jeweilige Feld einen Eintrag in dieser Tabelle schon existiert, trägt man nachträglich das „X“ ein.
- Das Feld *flg_keyfield* legt fest, welche Felder zur Gruppierung benutzt werden. Es können auch mehrere Felder angegeben werden, so dass bei sehr vielen Tabellen bzw. Daten die Datenpakete handhabbar bleiben. Natürlich sollte die hier angegebene Tabelle in */ARC/TABLES* als Schlüsseltable definiert worden sein.

Ansonsten definiert man eine Beziehung, indem man die Daten für die Quelltable in die Primärschlüsselfelder *arch_obj*, *tab_name*, *field_name* einträgt und von der Zieltabelle in die Felder *ref_arch_obj*, *ref_table*, *ref_field*.

ARCH_OBJ (SAP) Die im Customizing vom Benutzer erzeugten Archivierungsobjekte und die eingegebenen Einstellungen zu diesem Objekt werden in dieser Tabelle gespeichert. Deswegen muss es, bevor in unserer Archivierungslösung die Einstellungen gemacht werden, das Archivierungsobjekt über ADK-Framework erstellt und die notwendigen Parameter gepflegt sein.

4.4.2 Laufinfotabellen

Wie oben erwähnt, werden die Einträge in diesen Tabellen zur Laufzeit erzeugt und auch gelesen.

/ARC/RUNS Wenn ein Archivierungslauf für eine Anwendung gestartet wird, werden aktuelle Laufnummer, Startdatum mit Uhrzeit und der Benutzer in dieser Tabelle eingetragen. Die Felder für Status wird während des Programmlaufes immer aktualisiert und beim Beenden wird letztendlich auch das Enddatum mit der Uhrzeit in dem entsprechenden Feld gespeichert. Beim Löschvorgang wird diese Tabelle nochmal benutzt. Wenn der Löschvorgang automatisch gestartet wird, wird anhand der Laufnummer und des Status das Archiv geholt und die archivierten Daten aus den DB-Tabellen gelöscht. Beim manuellem Löschvorgang kann der Benutzer die löschbaren Archivierungsläufe auswählen und den Vorgang starten.

/ARC/RUNS_ADK In dieser Tabelle werden die vom ADK erzeugten Laufnummern gespeichert. In ADK wird zu jedem Archivierungsobjekt ein Lauf gestartet. Da aber laut dem Datenmodell für eine Anwendung mehrere Archivierungsobjekte existieren können, wird in der Archivierungsanwendung in der gleichen Anzahl von Archivierungsobjekten ADK-Läufe gestartet. Diese Nummern werden wieder zur Auswahl der richtigen Archive beim Löschen und Zurückladen benötigt.

/ARC/LOGS Wie der Name schon sagt, wird diese Tabelle zu Loggingzwecken benutzt. Während des Archivierungslaufs werden alle Statusmeldungen oder Fehlermeldungen in dieser Tabelle speichern. In erster Linie ist diese Tabelle für die Programmierer gedacht. Als Beispiel wäre, wenn das Programm im Hintergrund läuft und das Programm abbricht, kann es nicht festgestellt werden, wo es abgebrochen wurde und welche Aktion vorher ausgeführt wurde. Durch Debugausgaben in dieser Tabelle können die Fehlerstellen sehr leicht gefunden werden.

/ARC/ARCH_INDEX Diese Tabelle bildet die Grundlage der Indextabellen in der Archivierungsanwendung. Wenn der Benutzer in der Anwendung Einstellungen vornimmt, wird zu jedem Archivierungsobjekt eine Indextabelle generiert, sofern sie nicht vorhanden ist. Folgende Felder werden für jede Tabelle erstellt:

- Primärschlüsselfelder:
 - *mandt* Mandant
 - *arch_obj* das Archivierungsobjekt
 - *run_no* Unsere Archivierungslaufnummer
 - *document* die Archivierungslaufnummer von ADK

- *object_id* wird aus den Feldern, die in */ARC/TAB_SELEC* als Schlüsselfeld deklariert sind, erzeugt
- Nichtschlüselfelder:
 - „*selected_fields*“ alle Felder, die in der Tabelle */ARC/TAB_SELEC* als Indexfeld definiert sind
 - *archivekey* Name der Archivdatei
 - *obj_offset* Offset des Datenobjekts in der Archivdatei
 - *reloaded* Kennzeichen, ob der betroffener Datensatz zurückgeladen wurde

ADMI_RUN (SAP) Die Tabelle beinhaltet Informationen zu einem Archivierungslauf von ADK, wie Datum, Uhrzeit, Status, Benutzer usw. Zur besseren Übersicht wurde allerdings nur der Primärschlüssel *document*, wo vom ADK-vergebene Laufnummer gespeichert wird, angegeben.

ADMI_JOBS (SAP) Wenn ein Archivierungslauf im Hintergrund ausgeführt wird, werden die Informationen in der Tabelle gespeichert. Über die Tabelle kann festgestellt werden, welche Läufe als Hintergrundjob ausgeführt wurden.

4.4.3 Texttabellen

Die Tabellen, die das System die Eigenschaft Mehrsprachigkeit hinzufügen, heißen Texttabellen. Diese Tabellen beinhalten nur bestimmte Felder. Wenn eine Tabelle ein Textfeld zur Beschreibung haben soll, wird eine neue Tabelle erstellt und das Beschreibungsfeld in diese hinzugefügt. Außerdem bekommt die Tabelle ein Feld namens *LANGU*, wo der Sprachenschlüssel eingetragen wird. Später bei der Anzeige wählt SAP automatisch die entsprechende Zeile, wo der gleiche Schlüssel im System eingestellt ist. Zusammen mit dem Feld *LANGU* werden die Primärschlüssel von der eigentlichen Tabelle, die aber in der Texttabelle als Fremdschlüssel definiert sind, als Primärschlüssel deklariert.

In den mehrsprachigen Systemen wie SAP, sollte der Entwickler oder Datenmodellierer schon in der Designphase seiner Anwendung dies berücksichtigen, weil es von vornherein nicht bekannt ist, mit welcher Sprache der Endbenutzer später auf dem System arbeitet.

Der einzige zusätzlicher Aufwand besteht darin, die Daten in den Tabellen zu bearbeiten, also die Beschreibungen in anderen Sprachen zu übersetzen.

4.5 Klassen

Anhand der funktionalen Anforderungen in Kapitel 3.4.1 und dem Funktionsumfang der STS–Archivierung konnte festgestellt werden, dass die neue Archivierungsanwendung in fünf Funktionsbereiche unterteilt werden:

1. Archivierung der Datenbanktabellen
2. Lesen des Archivs
3. Löschen der Datenbank
4. Rückladen des Archivs
5. Protokollierung

Jeder Bereich arbeitet unabhängig voneinander. Daher ist es sinnvoll für jeden Aufgabenbereich eine Klasse zu bilden und die benötigten Funktionen in dieser Klasse zu kapseln. Die Abbildung 4.3 zeigt den ersten Entwurf für das Klassendiagramm.

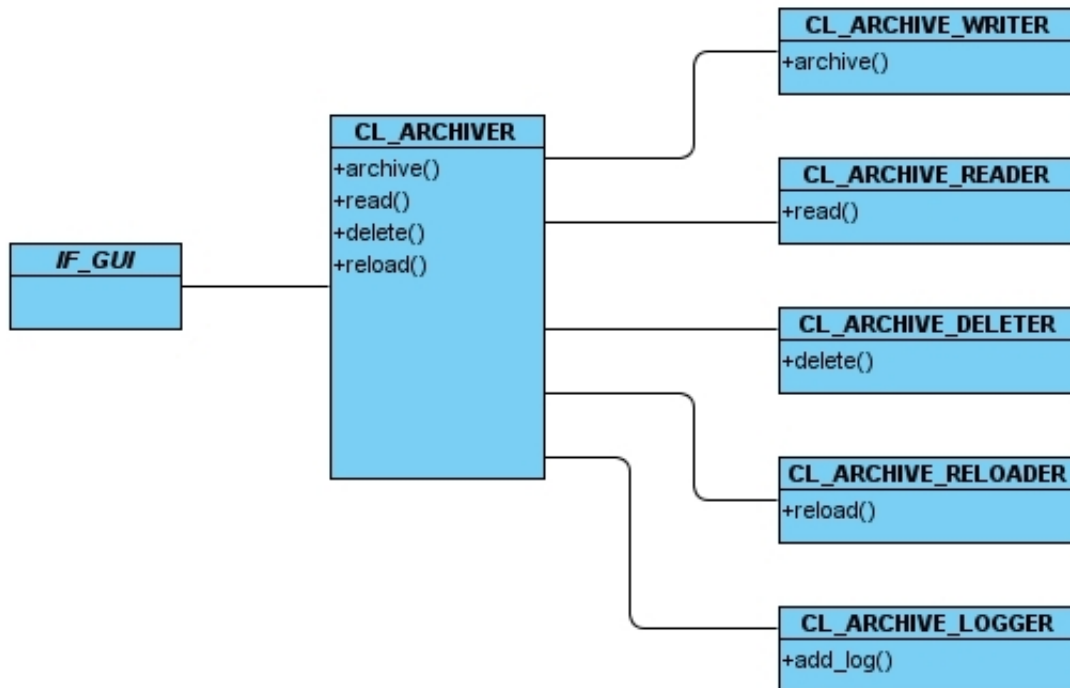


Abbildung 4.3: Klassendiagramm der Anwendung

Nur beim Anzeigen, Rückladen und Löschen können Gemeinsamkeiten auftreten, weil es in den Bereichen erst das Archiv gelesen wird und dann die für den Bereich spezifische Methoden aufgerufen werden. Eventuell können die Funktionen zum Lesen des Archivs zusammengefasst und in eine Oberklasse ausgelagert werden. Daher müsste diese Anforderung in der Realisierung weiter untersucht werden, wie umfangreich das Lesen ist.

Um die Anwendung von der GUI abzukoppeln, bietet sich eine Schnittstelle, in der die für das GUI benötigten Methoden definiert werden. Die Klasse `CL_ARCHIVER` dient als Controllerklasse und leitet die Methodenaufrufe an die entsprechenden Klassen. So ist die Anwendung besser wartbar und erweiterungsfähig, wenn zum Beispiel später die Anforderung kommt, dass die Anwendung später eine Web-Schnittstelle anbieten soll, muss an dem bestehenden Code nichts geändert werden.

4.6 GUI

Die Archivierung sollte wie die STS-Archivierung eine grafische Benutzeroberfläche bieten. Dadurch wird erreicht, dass Benutzer von einem Punkt aus die ganze Archivierung bequem durchführen und kontrollieren können.

Springen System Hilfe

STS Archivierungslauf

Einplanen Job Status Zur Zeit keine Aktion vorhanden.

Archivieren Löschen Anzeigen & Rückladen Archivierungsläufe

Suchen

Produkt ID bis
 Laufnummer bis
 BWL Datum 19.09.2006 bis
 Anlagedatum 19.09.2006 bis

Archivierungslauf starten

Laufnr	Produkt-ID	BWL-Datum	Anlagedatum	Username	Status
20367	2100	19.09.2006	19.09.2006	HEMEYER	○○○
20368	4600	19.09.2006	19.09.2006	HMEYER	○○○
20369	4600	19.09.2006	19.09.2006	HMEYER	○○○
20370	4600	19.09.2006	19.09.2006	HMEYER	○○○
20371	4600	19.09.2006	19.09.2006	HMEYER	○○○
20372	4600	19.09.2006	19.09.2006	HMEYER	○○○
20373	4600	19.09.2006	19.09.2006	HMEYER	○○○
20374	4600	19.09.2006	19.09.2006	HMEYER	○○○
20375	4600	19.09.2006	19.09.2006	HMEYER	○○○
20376	2100	19.09.2006	19.09.2006	HEMEYER	○○○
20378	5100	19.09.2006	19.09.2006	HEMEYER	○○○

11 Läufe zu 3 Produkt-ID's werden angezeigt. BW1 (3) (400) agenss07 INS

Abbildung 4.4: STS-Archivierung

Die Abbildung 4.4 zeigt die Oberfläche der STS–Archivierung. Das GUI besteht aus vier Tab-Reitern, die den ganzen Archivierungsprozess abdecken:

- Archivieren
- Löschen
- Anzeigen & Rückladen
- Archivierungsläufe

Die Oberfläche muss für die neue Archivierungslösung erweitert werden. In der neuen Lösung können mehrere Anwendungen für die Archivierung eingestellt und dafür die Archivierung gestartet werden. Daher muss bei der Archivierung eine Liste mit den eingestellten Anwendungen angezeigt werden.

Was in der STS–Archivierung fehlt, ist das Customizing, zum Einstellen und Bearbeiten von neuen Anwendungen, dazugehörigen Archivierungsobjekten, Tabellen und deren Beziehungen.

4.7 Archivierungsformat

Wie in der STS–Archivierung sollte in der neuen Lösung als Archivierungsformat XML ausgewählt werden. In Kapitel 2.4.1 wurden die Standardformate verglichen und die Vorteile von XML gegenüber den anderen aufgezählt. Hier werden die Punkte aufgezählt, die für die zu entwickelnde Lösung wichtig sind.

4.7.1 Metadaten

Durch die XML ist es möglich die Metadaten der Datenbanktabellen zusammen mit den Anwendungsdaten in der Archivdatei zu speichern. Damit ist es möglich, die Archivdatei auch längere Zeit nach ihrer Erstellung sowie über Releasewechsel hinweg zu lesen. Zu Metadaten gehören z.B.:

- Schema der Datenbanktabelle
- Datentyp einer Spalte
- Länge einer Spalte
- verwendete Codepage (ASCII, UTF-8, EBCDIC)
- Zahlenformat (z.B. Integer)

4.7.2 Anzeigen der Daten

Die eine Anforderung war das Anzeigen der archivierten Daten. Mit der XML ist es möglich ohne großen Aufwand die gewünschten Daten anzuzeigen. Denn XML wird von jedem Browser unterstützt und mit XSLT ist es möglich XML bei Bedarf in andere Formate zu transformieren.

4.7.3 Export der Daten

Die Anforderung war zwar nicht gestellt, aber im Falle der digitalen Steuerprüfung müssen die Daten auf einem Datenträger kopiert und an den Prüfer weitergegeben werden können. Die Daten müssen dabei in einem bestimmten Dateiformat vorliegen, eines davon ist XML. Mit der Anzeige der Daten in der Anwendung ist eigentlich die technische Grundlage geschaffen, da muss nur noch eine Funktion eingebaut werden, dass die Daten in eine Datei exportiert werden.

5 Realisierung

Das grundsätzliche Problem bei der Implementation der Datenarchivierung ist das Lesen von beliebigen Tabellen. Da die Datentypen und Tabellen zur Compilezeit nicht bekannt sind, gibt es zwei unterschiedliche Ansätze für die Umsetzung. Einerseits kann der Entwickler auf die Techniken der dynamischen Programmierung zugreifen und andererseits kann er nachdem die Einstellungen getätigt worden sind (Customizing), ein lauffähiges Programm generieren lassen, in dem die Datentypen und Tabellen bekannt sind und statisch auf die Tabellen zugegriffen werden kann.

Für die Anwendung wurden beide Varianten ausprobiert. Nur bei der Programmgenerierung wurde es bei der Archivierung belassen, weil es für den Vergleich der beiden Techniken ausreicht.

In Kapitel 3.1 wurde gezeigt, welche Möglichkeiten es unter SAP gibt, um eine Archivierungslösung zu entwickeln. Die Auswahl fiel auf ADK. Denn das wichtigste war, dass die alte Anwendung von Paricon auf ADK basierte. Man konnte sich auf bestehende Funktionen, Klassen oder auf Programme beziehen. Ausserdem musste für STS keine neuen Archivierungsobjekte angelegt werden.

5.1 Archivieren mit dynamischer Programmierung

Die Abbildung 5.1 zeigt den vereinfachten Ablauf der Archivierung. Mit der Archivierung meinen wir hier nur das Schreiben der Daten in das Archiv. Denn der gesamter Archivierungsprozess besteht aus einem Schreibprozess und darauf folgendem Lösprozess. Dann ist die Archivierung vollständig.

Das Schreibprogramm wird mit dem Programm *ARCHIVE_DATA* gestartet. Das Programm fordert vom Benutzer zuerst die Auswahl der Anwendung und ruft danach die Methode *archive* der Klasse *CL_ARCHIVER*, die als eine Kontrollerklasse fungiert. Die leitet die Aufrufe an die konkreten Klassen, wo die eigentliche Logik implementiert ist.

Die Methode *archive* der Klasse *CL_ARCHIVER* erzeugt anhand des Anwendungsnamens ein Selektionsbildschirm mit den Feldern, die in der Tabelle */ARC/APP_SELEC* für die jeweilige Anwendung vorher eingegeben worden sind. Mit den Kriterien, die der Benutzer

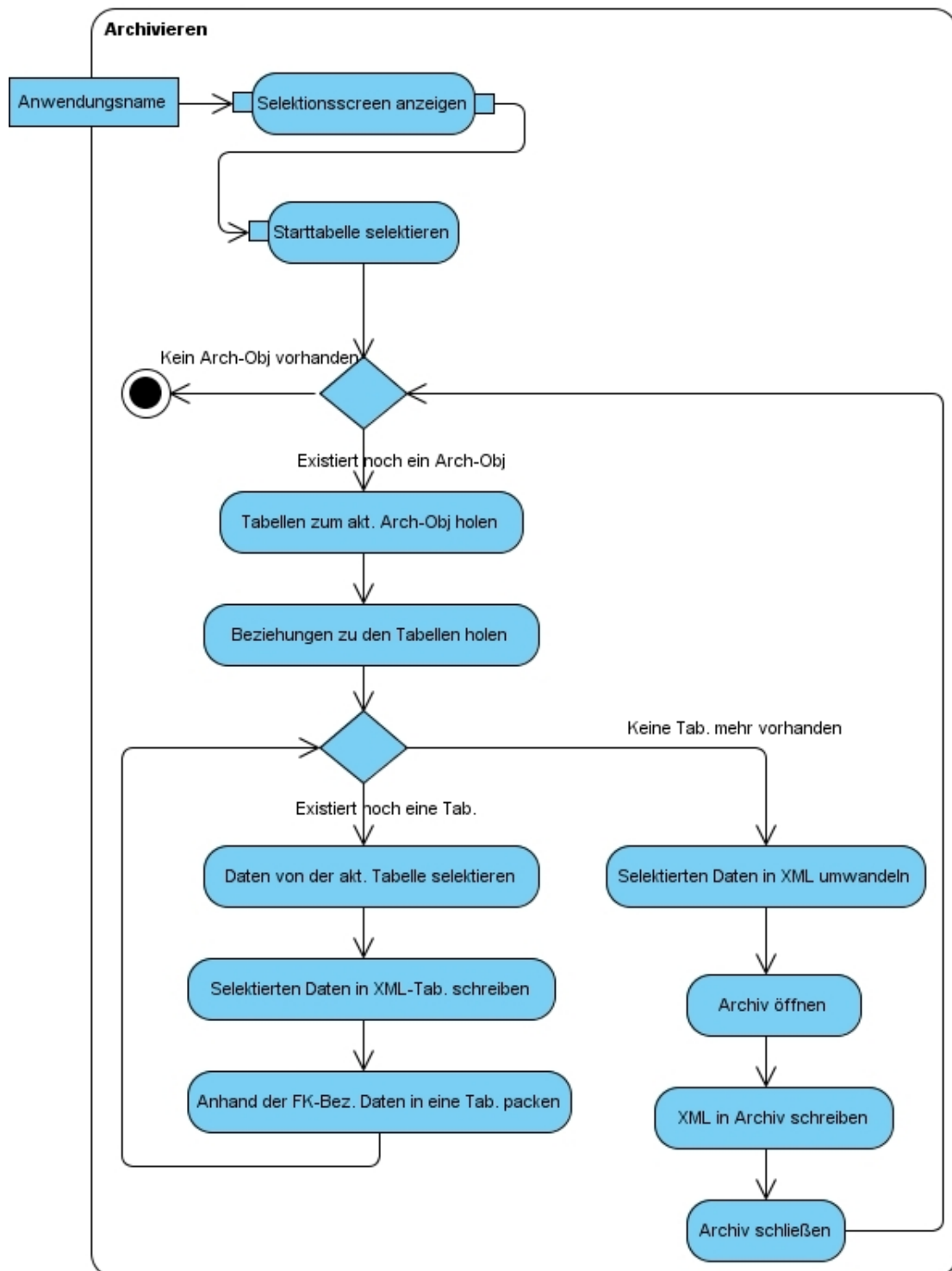


Abbildung 5.1: Ablauf der Archivierung

für jedes Feld eingibt, wird eine Where-Bedingung erstellt. Zusammen mit der Where-Bedingung und mit dem Anwendungsnamen wird dann die Methode *archive* der Klasse *CL_ARCHIVE_WRITER* aufgerufen.

In der Klasse *CL_ARCHIVE_WRITER* ist die eigentliche Logik des Schreibens implementiert. Die Selektion der Daten der zu archivierenden Tabellen, Transformieren der Daten in XML und Übergeben der XML-String an ADK geschieht in dieser Klasse. Dabei erfolgen alle Zugriffe auf die Tabellen dynamisch. Das Schlüsselwort dazu in ABAP ist das Feldsymbol. Die Feldsymbole kann man mit den in C bekannten Zeigern vergleichen, die aber in ABAP einfacher handhabbar im Gegensatz zu C sind.

5.1.1 Speichern der Daten für Tabellenbeziehungen

Abbildung 5.2 zeigt den Aufbau der Tabelle, wo Daten für Fremdschlüsselbeziehungen gespeichert werden. Wenn das Schreibprozess für ein Archivierungsobjekt zu Ende ist wird die Tabelle neu initialisiert. Dabei werden bis auf die Zeilen alles gelöscht, wo unter dem Feld *flg_global* ein 'X' steht. Das Feld dient zum Kennzeichnen, ob die Daten bei anderen Archivierungsobjekten gebraucht werden.

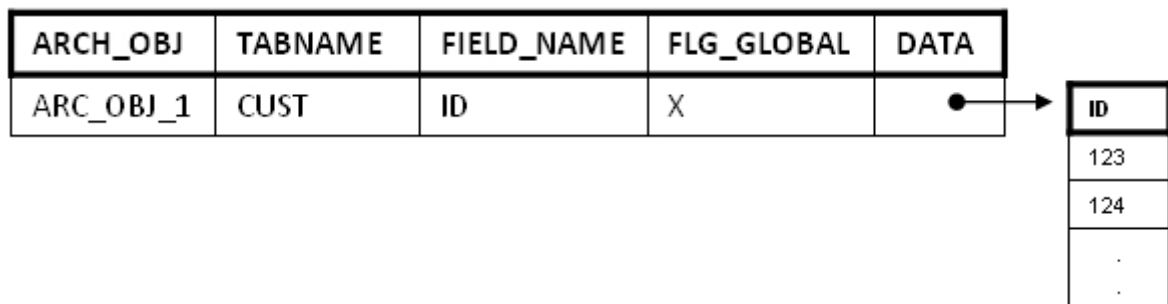


Abbildung 5.2: Fremdschlüsseltabelle

Das letzte Feld mit dem Namen *DATA* ist eine generische Datenreferenz. Das heißt, da werden keine Werte gespeichert, sondern die Adresse der Daten, in diesem Fall die Adresse einer internen Tabelle. Diese Tabelle wird auch zur Laufzeit erzeugt und die Werte dann gespeichert.

Dabei wird es mit vielen Variablen gearbeitet, die vom generischen Typ sind. Da ist aber jetzt die Gefahr, dass der Überblick verloren geht und nicht mehr klar ist, wofür welche Variable benutzt wird. In dem Fall muss jeder Schritt im Sourcecode kommentiert werden, damit der Entwickler selbst oder auch andere das Programm später weiterentwickeln und warten können.

Im Kapitel 4.1.1 wurden weitere Sprachelemente vorgestellt, mit den die dynamische Programmierung ermöglicht wird. Deswegen wird an dieser Stelle auf die entsprechenden Kapitel verwiesen.

5.1.2 Anordnen der Archivierungsobjekte

In den Kapiteln 3.1 und 4.3 wurde die Notwendigkeit der Archivierungsobjekte ausführlich diskutiert und erläutert, warum für eine Anwendung, die archiviert werden soll, mehrere Archivierungsobjekte definiert werden. Bei der Implementierung wurde festgestellt, wenn mehrere Archivierungsobjekte existieren, dass das Objekt bei der Archivierung nicht zufällig ausgewählt werden sollte. Denn zwischen den Archivierungsobjekten können logische Beziehungen existieren.

Die Beziehung zwischen den wird definiert, wenn eine Tabelle von einem Archivierungsobjekt zu einer Tabelle von einem anderen Archivierungsobjekt besteht. Aus diesem Grund sollten die Archivierungsobjekte in einer Reihenfolge gebracht werden.

Als erstes wird überprüft, ob die Starttabelle der Anwendung archiviert wird, wenn ja, wird das dazugehörige Archivierungsobjekt genommen und in die Liste gepackt.

Zweitens wird es in den Beziehungen die Tabellen selektiert, die eine Beziehung zu der Starttabelle haben und es werden die dazugehörigen Archivierungsobjekte, sofern die noch nicht vorhanden sind, in die Liste hinzugefügt.

Drittens es werden von den Tabellen die Archivierungsobjekte selektiert, die bei den Beziehungen der Tabellen unterschiedliche Archivierungsobjekte definiert sind. Das heisst es existiert eine Beziehung zwischen zwei Archivierungsobjekten. Dabei wird natürlich das Archivierungsobjekt von der Zieltabelle geholt.

Und zuletzt werden die restlichen Archivierungsobjekte der Anwendung in die Liste gepackt.

Nachdem die Archivierungsobjekte in der richtigen Reihenfolge stehen, wird für jedem einen Archivierungslauf im ADK gestartet. Am Ende stehen die Daten von Tabellen für jedes Archivierungsobjekt in getrennten Archiven.

Dabei werden zuerst dazugehörigen Tabellennamen aus der Tabelle */ARC/TABLES* und die Tabellenbeziehungen aus der Tabelle */ARC/TAB_SELEC* geholt.

5.1.3 Dynamische Where-Bedingung

Da erst zur Laufzeit bekannt ist, welche Tabelle mit welcher Tabelle in Beziehung stehen, muss auch die Where-Bedingung der SQL-Abfrage dynamisch aufgebaut werden. Hierzu bietet ABAP einige Methoden das ganze zu vereinfachen.

Einmal gibt es für Where-Bedingung der IN-Operator. So sieht der Syntax davon aus:

```
...  
WHERE column1 IN selectiontab1 AND  
column2 IN selectiontab2  
...
```

Dabei ist `column` eine Spalte der Datenbanktabelle und `selectiontab` eine sogenannte Selektionstabelle. Eine Selektionstabelle besteht aus den Spalten `sign`, `option`, `low`, `high`. Deswegen müssen wir, bevor wir eine Tabelle aus unsere oben beschriebenen Fremdschlüsseltabelle benutzen wollen, eine neue Tabelle mit diesen fünf Spalten generieren und als `sign I` (include), für `option EQ` (equal) unter dem Feld `low` unsere Werte speichern. Das Problem dabei ist, die Tabelle darf nicht unendlich groß sein. Etwa bei 2000 Datensätzen gibt SAP eine Fehlermeldung aus. Was man dagegen machen kann, ist nur, wenn die zu vergleichende Werte numerisch sind, kann man die in Gruppen zusammenfassen und mit dem `BT`-Operator bei `low` die untere Grenze und bei `high` die obere Grenze angeben.

Neben IN-Operator bietet ABAP noch die Methode „FOR ALL ENTRIES WHERE“. So ist der Syntax:

```
...  
FOR ALL ENTRIES IN itab WHERE column1 = itab-col1 AND column2  
= itab-col2  
...
```

Bei dieser Methode muss unsere Datentabelle aus der Fremdschlüsseltabelle nicht verändert werden. An Stelle von `itab` kann eine Referenz auf die Tabelle angegeben werden. Der logische Ausdruck der Where-Bedingung wird für jede einzelne Zeile der internen Tabelle `itab` ausgewertet. Das Problem besteht nur, wenn die Tabellenbeziehung aus mehreren Feldern besteht. Dann muss hier ein Kreuzprodukt der beiden Felder erstellt und statt `itab` diese neue Tabelle angegeben werden. Der Nachteil dabei ist, wenn zu viele Daten in den Spalten existieren. Bei zwei Spalten à 1000 Zeilen hätte man bei der neuen Tabelle 1 Mio. Datensätze, so das Vergleich erheblich lange dauern würde.

Als Fazit kann ich sagen, dass beide Methoden ihre Vor- und Nachteile haben. Der Entwickler muss anhand seiner Problemstellung entscheiden, welche Methode für seine Lösung am besten geeignet ist. Bei der Datenarchivierung, wo der Zweck die Entlastung des Systems ist und deswegen eigentlich meistens sehr große Tabellen mit über 100 000 Datensätzen hat,

ist empfehlenswert beide Methoden zu kombinieren. Denn meistens sind die Primärschlüssel vom numerischen Typ und da kann man mit Unter- und Obergrenzen arbeiten. Bei anderen Typen könnte man die andere Methode benutzen. Denn es gibt wenige Beziehungen, die mit mehreren Feldern definiert sind.

5.2 Archivieren mit Programmgenerierung

Bei dieser Variante besteht die Archivierung aus zwei Schritten. Der einer ist die Generierung der Programmmodule zur Ausführung und der andere ist die eigentliche Ausführung der Archivierung durch den Benutzer.

Wie es schon erwähnt wurde, habe ich es einfachheitshalber bei der Generierung des Schreibprozesses belassen. Wenn sich die Entscheidung, wie die Anwendung implementiert werden sollte, auf die Generierung fällt, sollten auch Lösch-, Anzeige- und Rückladeprozess mitberücksichtigt werden.

Die Programmgenerierung sollte, sobald der Benutzer im Customizing etwas geändert oder etwas neues hinzugefügt hat, erfolgen. Abbildung 5.3 zeigt die Klasse, die die Programme und Funktionsbausteine in SAP generiert.

CL_GENERATE_APP
-app_name
+generate() +gen_export_parameter(i_arch_obj, it_arch_objs, rt_export_parameter) +gen_import_parameter(i_arch_obj, rt_import_parameter) +gen_report(it_code) +gen_report_begin(it_arch_objs, ct_code) +gen_report_end(i_arch_obj, it_import_parameter, it_export_parameter, ct_code) +gen_fuba(i_func_name, it_import_parameter, it_export_parameter, r_include_name) +gen_fuba_decl(i_include_name) +gen_fuba_source(i_arch_obj, i_include_name) +sort_tables(it_tables, it_tab_selec, rt_tables) +order_arch_objs(rt_arch_objs_sorted)

Abbildung 5.3: Klasse zur Programmgenerierung

Die Klasse generiert für eine vorgegebene Anwendung ein Programm und die Funktionsbausteine mit der Anzahl der Archivierungsobjekte für die jeweilige Anwendung. Die Abbildung 5.4 zeigt den Ablauf der Generierung.

In einem Funktionsbaustein zu einem Archivierungsobjekt werden nur die Tabellen abgearbeitet, die zu dem Archivierungsobjekt zugeordnet sind. Dabei wird für jede Datenbanktabelle eine interne Tabelle und zum Zugriff auf diese interne Tabelle ein Feldsymbol deklariert.

Das Lesen der Datenbanktabellen erfolgt nicht dynamisch, sondern, es existiert für jede Datenbanktabelle eine Select-Anweisung. Dadurch fallen das aufwendige Bestimmen des Typs der Datenbanktabelle und das Lesen der Datenbanktabelle in die interne Tabelle vom generischen Typ aus. Dadurch wird die Anzahl der Codezeilen natürlich entsprechend der Anzahl der Datenbanktabellen viel. Man spart aber an Zeit beim Programmablauf.

Die Beziehung zwischen den Archivierungsobjekten ist so gelöst, dass für alle Felder, die die Beziehungen definieren, eigene statische interne Tabellen im generierten Programm deklariert werden. Diese Tabellen werden in den Funktionsbausteinen zu den entsprechenden Archivierungsobjekten, den die Datenbanktabelle des Feldes zugeordnet ist, gefüllt und vom Funktionsbaustein zurückgegeben. Danach wird diese interne Tabelle an die Funktionsbausteine, wo es gebraucht wird übergeben.

5.3 Gruppierung der Daten

Auf die Gruppierung wurde schon im Kapitel 4.4 bei der Erklärung von Tabellen `/ARC/TABLES` und `/ARC/TAB_SELEC` schon eingegangen. In diesem Kapitel wird die Vorgehensweise bei Implementierung erklärt.

Unabhängig von der Art der Programmierung müssen die Daten von Tabellen, bevor die ins Archiv gepackt werden, gruppiert, sozusagen in Paketen zusammengefasst werden. Denn der Arbeitsspeicher in einem System ist nicht unbegrenzt und viele Anwendungen teilen die Kapazität.

In einem Datenbanksystem können dagegen Tabellen existieren, die mehrere Gigabyte umfassen. Deswegen ist es bei solchen Tabellen unmöglich, alle Daten auf einmal zu lesen. Auch wenn die Tabellen nicht so groß sind, existiert das Problem aber immer noch. Im Falle der Archivierung werden die Daten aus den Tabellen erst mal im Arbeitsspeicher zwischengespeichert und erst wenn die Selektion zu Ende ist, werden die ins Archiv geschrieben. In diesem Fall kann der Speicher auch knapp werden, wenn zu viele Tabellen selektiert werden.

Die Lösung dafür ist das paketweise selektieren und schreiben der Daten. Dazu wird eine Tabelle ausgesucht, mit der auch die Selektion gestartet wird. Diese Tabelle dient sozusagen

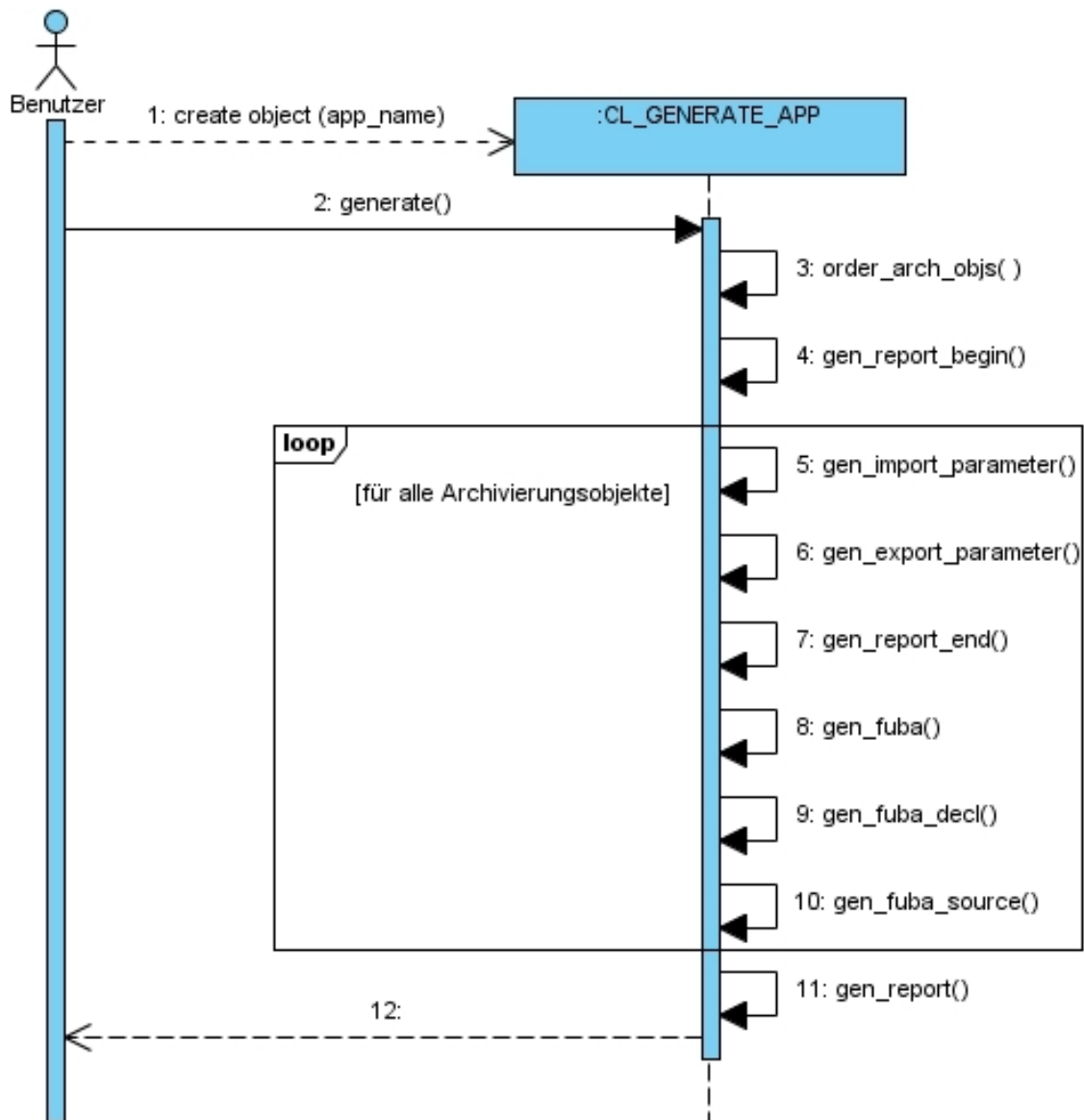


Abbildung 5.4: Ablauf der Generierung

als Starttabelle für Selektion. Je nachdem wie klein die zu archivierenden Pakete sein sollen, werden Felder von dieser Starttabelle ausgewählt. Es gilt je mehr Felder, desto kleiner bzw. weniger werden die selektierten Daten. Diese Definitionen geschehen in Customizing, in dem die Tabellen von oben mit entsprechenden Werten gefüllt werden.

Bei der Programmausführung, also Archivierung, wird erst mal eine Liste mit Werten aus diesen Feldern erzeugt und danach über diese Liste iteriert. Bei der Select-Anweisung werden diese einzelnen Werte in die Where-Bedingung eingebunden.

Mit diesem Verfahren ist es jetzt möglich beliebig große Tabellen zu lesen. Dabei müssen natürlich untersucht werden, wie viele Tabellen zu archivieren sind und wie viel Speicherplatz die einzelnen Tabellen auf der Datenbank in Anspruch nehmen. Bestimmung der richtigen Anzahl von Feldern ist hier entscheidend für die Performance und Zuverlässigkeit des Systems. Wenn zu wenige Felder ausgewählt werden, kann es sein, dass das System die Archivierung abbricht, weil der Speicher voll ist. Und auf der anderen Seite, wenn zu viele Felder existieren, dass unnötig sehr viele Select-Anweisungen über die zu archivierenden Tabellen ausgeführt wird. Das führt dazu, dass die Archivierung sehr lange dauert.

5.4 Archivieren von generierten Tabellen

Einer der Anforderungen war das Archivieren von generierten Datenbanktabellen. Das Problem dabei ist, dass die zu archivierende Anwendung zur Laufzeit neue Datenbanktabellen erzeugen. Die Archivierungsanwendung kann aber nicht feststellen welche Tabellen die sind, wie die Struktur der Tabellen aussehen. Es ist auch nicht möglich über Customizing in der Archivierungsanwendung solche Tabellen bekannt zu machen, denn der Anwender kann auch nicht so einfach rausbekommen, wie die Tabellen heißen.

Um dieses Problem zu lösen bietet SAP sogenannte Business Add-Ins, kurz BAdI. BAdIs sind Erweiterungen des SAP-Standards.

Um für ein Programm eine Erweiterung vorzusehen, wird ein Business Add-In definiert. Im Rahmen des Add-Ins legt der Anwendungsentwickler ein Interface an. Die Erweiterungsverwaltung generiert daraus eine Adapterklasse, die das Interface implementiert und damit die Schnittstelle zur Implementierung des Kunden, Partners etc. herstellt. Der Entwickler erzeugt im Anwendungsprogramm eine Instanz der Adapterklasse und ruft zum gegebenen Zeitpunkt die entsprechende Methode auf. [SAP]

Zu einem Business Add-In gehört ein Interface und eine dieses implementierende Adapterklasse. Das Interface wird vom Verwender implementiert. Die Abbildung 5.5 veranschaulicht die Architektur von BAdI.

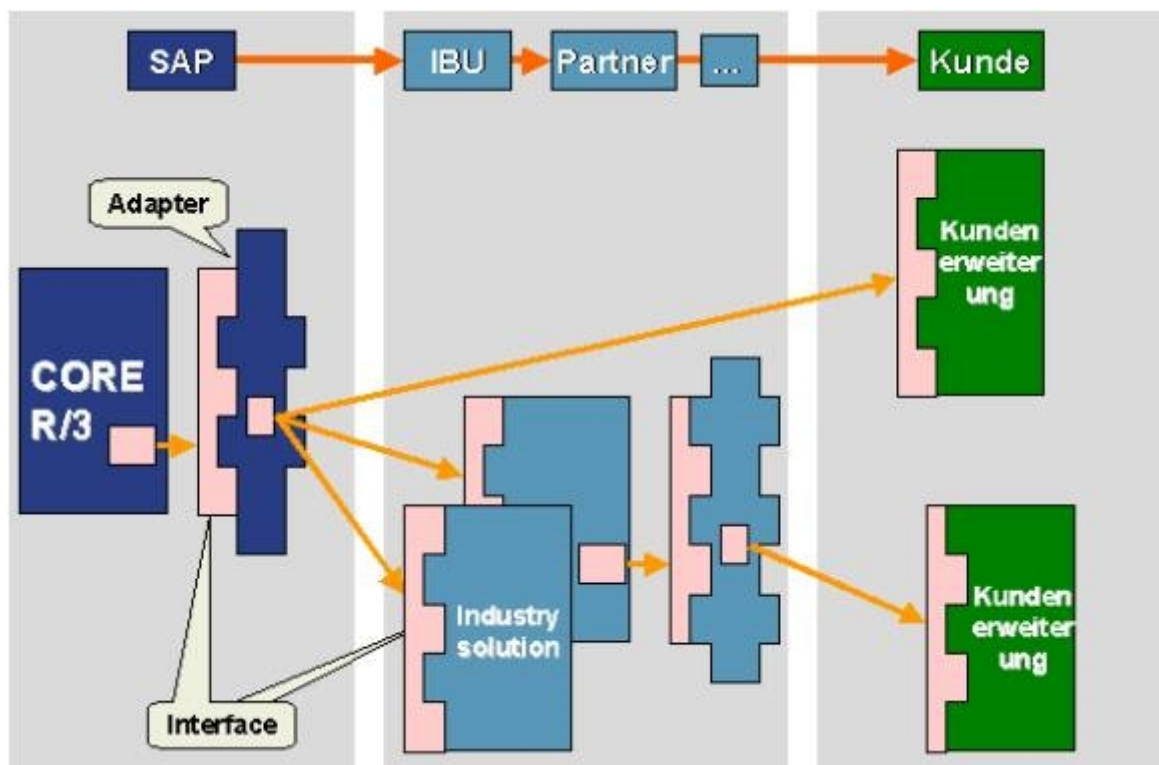


Abbildung 5.5: Architektur von BAdI (Quelle:SAP)

In dem konkreten Fall von Archivierungslösung implementiert jede Anwendung, die Datenbanktabellen generieren, das Interface. Im Customizing der Archivierungslösung wird für die generierten Tabellen eine symbolische Tabelle definiert, die die gleiche Struktur wie generierten Tabellen hat und als Typ wird generiert angegeben. Wenn bei der Archivierung diese Tabelle an der Reihe ist, wird die Methode vom Interface der anderen Anwendung aufgerufen. Die Methode liefert die Daten als interne Tabelle zurück. Damit können die Daten weiterverarbeitet und in das Archiv gepackt werden.

5.5 Fazit

In diesem Kapitel wurden die Themen angesprochen, die bei der Realisierung einer generischen Archivierungslösung wichtig sind.

Die Entscheidung, die der Entwickler oder das Unternehmen treffen muss, ist, mit welchem Verfahren eine Anwendung implementiert wird. Jedes Verfahren hat seine Vor- und Nachteile.

Der Vorteil bei den dynamischen Methoden sind, dass bei kleinen Aufgaben sehr schnell eine Lösung entwickelt werden kann. Bei der Benutzung von vielen von solchen generischen Variablen geht aber die Übersichtlichkeit verloren und muss, damit der Entwickler später weiß, welche Variable wofür benutzt wurde, sehr viel kommentiert werden.

Nachteil von Programmgenerierung hängt ein wenig vom Programmierer ab. Man neigt dabei sehr schnell die Variablen einfach durchzunummerieren. Das macht den Code natürlich unleserlich. Es wird nicht oder sehr wenig auf die Formatierung des Quellcodes beachtet.

6 Schluss

In diesem Kapitel werden die Ergebnisse zusammengefasst und die Erkenntnisse hervorgebracht. Dabei wird die entstandene Lösung nochmal kritisch betrachtet und weitere Schritte werden aufgezählt, was noch gemacht werden muss, um die Anwendung produktiv einsetzen zu können.

6.1 Zusammenfassung

Ziel dieser Bachelorarbeit war ausgehend von einer im produktiven Betrieb eingesetzten Archivierungslösung ein Konzept für eine allgemein gültige Lösung für die Datenarchivierung zu erstellen. Der in dem Konzept vorgeschlagene Lösungsvorschlag sollte später in einem Prototyp umgesetzt werden. Das Basissystem, auf dem die Anwendung laufen sollte, war SAP.

Um die Aufgabe zu lösen, wurde zuerst die Archivierungslösung, die noch im Betrieb ist, nach ihren Fähigkeiten untersucht. Die Anwendung war speziell für ein anderes Produkt implementiert und daher viel zu spezifisch. Für die Tabellen aus den anderen Anwendungen konnte die Lösung, ohne den Code zu ändern nicht eingesetzt werden.

Nach der Analyse dieser Lösung wurden neue Anforderungen erstellt, die die neue Anwendung erfüllen sollte. Bei der Erstellung der Anforderungen war das Ziel nicht irgendein Prototyp, sondern ein Produkt, das auch vermarktet werden konnte.

Um die Anforderungen zur Datenarchivierung zu erfüllen, bot das SAP-System zwei Technologien an, das ADK(Archive Development Kit) und die XML-Archivierung. Nach der Untersuchung der beiden Technologien wurde letztendlich für das ADK entschieden.

Aufgrund der Einschränkungen bei der alten Lösung wurde auch das Datenmodell neu entworfen. Dabei wurde auf Mandantenfähigkeit und Mehrsprachigkeit Wert gelegt und entsprechend in den Tabellen übernommen. Durch das neue Datenmodell ist es nun möglich beliebige Tabellen von SAP-Anwendungen zu archivieren.

Bei der Implementierung wurden zwei Verfahren, die die Programmiersprache ABAP anbietet, angewendet. Das eine ist die dynamische Programmierung und das andere Code-Generierung. Mit der dynamischen Programmierung wurden die Funktionen Archivieren, Löschen, Rückladen und Anzeigen implementiert. Somit konnte erstmal überprüft werden, dass der erster Entwurf vom Datenmodell bis auf einige kleine Fehler anwendbar ist. Nach der Korrektur wurde diesmal nur das Archivieren mit der Code-Generierung implementiert.

6.2 Fazit

Durch das Prototyp und auch mit dem darunter liegenden Datenmodell konnte gezeigt werden, dass es möglich ist, mit einer einzigen Anwendung beliebige Tabellen unter SAP zu archivieren. Denn sowohl beim ADK und auch bei der XML-Archivierung musste für jede Anwendung extra Programme zum Archivieren, Löschen, Rückladen und Anzeigen entwickelt werden. Diese Tätigkeit fällt jetzt vollständig weg.

Durch das Customizing ist die Möglichkeit gegeben, beliebige Anwendungen einzustellen und deren Tabellen und Beziehungen zwischen den Tabellen zu definieren.

Der Prototyp wurde auf einen Stand gebracht, der die meisten Anforderungen grundsätzlich erfüllt. Es bedarf allerdings noch großen Aufwand, damit der Prototyp den Produktstatus erreicht.

6.3 Ausblick

In der Realisierung wurde bewusst, auf GUI verzichtet, weil die GUI-Entwicklung sehr zeitaufwändig ist und nicht der Sinn der Aufgabe war. Damit aber der Produkt vermarktet werden kann, muss eine Benutzeroberfläche implementiert werden. Die STS-Archivierung bietet dafür eine gute Grundlage.

Es ist auch noch nicht möglich, einzelne Daten anzeigen oder zurückzuladen, weil noch keine Indizes über die archivierten Daten gebildet werden. Sonst muss immer der gesamte Archiv von einem Archivierungslauf zurückgeladen oder angezeigt werden. Weiterhin sollte eine Möglichkeit geschaffen werden, um die archivierten Daten in eine Datenformat zu exportieren. Denn im Falle einer Betriebsprüfung kann der Steuerprüfer die Daten auf einem Datenträger verlangen.

In den SAP-Standardreports gibt es die Möglichkeit als Datenquelle Datenbank oder Archiv auszuwählen. Beim Archiv wird statt eines Programms der Name des Archivierungsobjekts, das für die aktuelle Anwendung definiert ist, angegeben. Daraus führt das ADK das für die

Anzeige definierte Programm. So muss natürlich für jede Anwendung oder Report ein anderes Programm geschrieben werden. Jetzt müsste nur noch ein Archivierungsobjekt definiert und ein allgemeingültiges Programm zur Anzeige implementiert werden, das auf die neue Archivierungslösung aufbaut.

Literaturverzeichnis

- [BIT06] BITKOM. Leitfaden zum elektronischen datenzugriff der finanzverwaltung. www.bitkom.de, 2006.
- [Boh05] Johannes Bohn. Entwicklung eines archivierungsmoduls in einer sap-umgebung. Diplomarbeit, Hochschule für Angewandte Wissenschaften Hamburg, 2005.
- [Gmb] AvenDATA GmbH. Gdpdu portal. <http://www.gdpdu-portal.com/>.
- [Hoe00] Frank Hoeger. Anforderungsanalyse und spezifikation zur einföhrung eines optischen archivsystems der lufthansa technik logistik gmbh für die archivierung on warenbegleitpapieren. Diplomarbeit, Hochschule für Angewandte Wissenschaften Hamburg, 2000.
- [Kam97] Dr. Ulrich Kampffmeyer. *Grundsätze der elektronischen Archivierung in Deutschland*. VOI, 1997.
- [Kam02] Dr. Ulrich Kampffmeyer. Revisionssichere archivierung. www.contentmanager.de, 2002.
- [Kam05] Dr. Ulrich Kampffmeyer. Gdpdu & elektronische archivierung. Vortrag auf der GDPdU-Jahreskonferenz 2005, 2005.
- [Nat] Deutsche Nationalbibliothek. Kompetenznetzwerk langzeitarchivierung. <http://www.nestor.de/>.
- [Nim00] Brigitta Nimz. Elektronische archivierung. <http://www.lwl.org/waa-download/archivpflege/heft53/nimz.html>, 2000.
- [SAP] SAP. Online hilfe. <http://help.sap.com>.
- [Sch01] Ralf Schaarschmidt. *Archivierung in Datenbanksystemen*. Vieweg+Teubner Verlag, 2001.
- [Ste02] Helmut Stefani. *Datenarchivierung mit SAP*. Galileo Press, SAP Press, 2002.
- [SYM] SYMWEB. Internetlexikon. <http://www.symweb.com>.
- [VOI] VOI. Verband organisations- und informationssysteme e.v. <http://www.voi.de/>.

Glossar

ABAP ist eine proprietäre Programmiersprache der Softwarefirma SAP, die für die Programmierung kommerzieller Anwendungen im SAP-Umfeld entwickelt wurde.

ADK Archive Development Kit.

Das ADK stellt die Laufzeitumgebung für die Datenarchivierung bereit und bildet die Grundlage für die Erstellung von Datenarchivierungslösungen. Es bietet Schnittstellen zwischen den Anwendungen, der Datenbank und dem Archiv, in dem die Daten abgelegt sind.

Archivierungsobjekt Das Archivierungsobjekt ist das Herzstück des Archivierungskonzeptes im R/3. Im Archivierungsobjekt ist festgelegt, was und wie archiviert wird. Das Archivierungsobjekt beschreibt also, welche Datenbankobjekte gebündelt werden müssen, um ein betriebswirtschaftlich abgeschlossenes Objekt zu erhalten, das unabhängig von den technischen Gegebenheiten zum Zeitpunkt der Archivierung interpretierbar ist.[SAP]

Customizing Übersetzt heißt es kundenspezifische Anpassungen. In Bezug auf eine Software versteht man die Anpassung von Standardprogrammen an anwenderspezifische Belange.[SYM] Laut SAP bedeutet Customizing: „Mit Customizing passen Sie die unternehmensneutral und branchenspezifisch ausgelieferte Funktionalität den spezifischen betriebswirtschaftlichen Anforderungen Ihres Unternehmens an...Das Customizing steuert das Verhalten der SAP-Anwendungen. Sie können mit vielen SAP-Anwendungen erst arbeiten, nachdem Sie das entsprechende Customizing vorgenommen haben. Z.B. stellen Sie im Customizing die Währungen und Einkäufergruppen ein, die in Ihrem Unternehmen genutzt werden.“[SAP]

R/3 Der Name SAP R/3 entstand aus der Konzeption als Client–Server–System, das „R“ steht dabei für realtime („Echtzeit“) und die „3“ steht für die drei Ebenen (Schichten), aus denen ein R/3–System besteht

Records Management In der Norm ISO 15489 wird Records Management definiert als 'als Führungsaufgabe wahrzunehmende, effiziente und systematische Kontrolle und

Durchführung der Erstellung, Entgegennahme, Aufbewahrung, Nutzung und Aussonderung von Schriftgut einschließlich der Vorgänge zur Erfassung und Aufbewahrung von Nachweisen und Informationen über Geschäftsabläufe und Transaktionen in Form von Akten’.

PricewaterhouseCoopers definiert Records Management als Methode, ‘die eine Organisation dabei unterstützt, physische und elektronische Geschäftsunterlagen während des ganzen Lebenszyklus optimal für die Geschäftsprozesse nutzbar zu machen und eine professionelle Archivierung zu gewährleisten. Records Management umfasst damit nicht nur die Einführung neuer technischer Lösungen, sondern insbesondere auch die Definition von entsprechenden Strategien und Standards sowie die Anpassungen der Aufbau- und Ablauforganisation.’

VOI Verband Organisations- und Informationssysteme e.V. ist der Verband der Anbieter für Enterprise Content Management (ECM) und Dokumenten- Management- Systeme (DMS) in Deutschland.

Index

ADK - Archive Development Kit, [23](#)

Archivierung, [11](#)

 Datenarchivierung, [16](#)

 Datenbankarchivierung, [16](#)

 Dokumentenarchivierung, [16](#)

 XML-Archivierung, [24](#)

Archivierungsobjekt, [23](#), [37](#)

ASCII, [18](#)

Comma Separated Values, [19](#)

Content, [29](#)

CSV, [19](#)

Customizing, [28](#)

Customizingtabellen, [35](#)

Emulation, [21](#)

Laufinformationen, [28](#)

Laufinfotabellen, [35](#)

Migration, [20](#)

Records Management, [11](#)

SGML, [19](#)

Technikmuseum, [21](#)

Texttabellen, [37](#), [40](#)

Unicode, [18](#)

UTF-8, [18](#)

XML, [20](#)

XML-Archivierungsobjekt, [24](#)

Grundsätze zum Datenzugriff und zur Prüfbarkeit digitaler Unterlagen (GDPdU)
(BMF-Schreiben vom 16. Juli 2001 - IV D 2 - S 0316 - 136/01 -)

Unter Bezugnahme auf das Ergebnis der Erörterungen mit den obersten Finanzbehörden der Länder gilt für die Anwendung der Regelungen zum Datenzugriff und zur Prüfbarkeit digitaler Unterlagen (§ 146 Abs. 5, §147 Abs. 2, 5, 6, § 200 Abs. 1 AO und § 14 Abs. 4 UStG) Folgendes:

I. Datenzugriff

Nach § 147 Abs. 6 AO ist der Finanzbehörde das Recht eingeräumt, die mit Hilfe eines Datenverarbeitungssystems erstellte Buchführung des Steuerpflichtigen durch Datenzugriff zu prüfen. Diese neue Prüfungsmethode tritt neben die Möglichkeit der herkömmlichen Prüfung. Das Recht auf Datenzugriff steht der Finanzbehörde nur im Rahmen steuerlicher Außenprüfungen zu. Durch die Regelungen zum Datenzugriff wird der sachliche Umfang der Außenprüfung (§ 194 AO) nicht erweitert; er wird durch die Prüfungsanordnung (§ 196 AO, § 5 BpO) bestimmt. Gegenstand der Prüfung sind wie bisher nur die nach § 147 Abs. 1 AO aufbewahrungspflichtigen Unterlagen. Es ist jedoch erforderlich, die Prüfungsmethoden den modernen Buchführungstechniken anzupassen. Dies gilt um so mehr, als in zunehmendem Maße der Geschäftsverkehr papierlos abgewickelt wird und ab dem 1. Januar 2002 der Vorsteuerabzug aus elektronischen Abrechnungen mit qualifizierter elektronischer Signatur und Anbieter-Akkreditierung nach dem Signaturgesetz möglich ist.

Die Einführung dieser neuen Prüfungsmethode ermöglicht zugleich rationellere und zeitnähere Außenprüfungen.

1. Umfang und Ausübung des Rechts auf Datenzugriff nach § 147 Abs. 6 AO

Das Recht auf Datenzugriff beschränkt sich ausschließlich auf Daten, die für die Besteuerung von Bedeutung sind (steuerlich relevante Daten).

Die Daten der Finanzbuchhaltung, der Anlagenbuchhaltung und der Lohnbuchhaltung sind danach für den Datenzugriff zur Verfügung zu halten.

Soweit sich auch in anderen Bereichen des Datenverarbeitungssystems steuerlich relevante Daten befinden, sind sie durch den Steuerpflichtigen nach Maßgabe seiner steuerlichen Auf-

zeichnungs- und Aufbewahrungspflichten zu qualifizieren und für den Datenzugriff in geeigneter Weise vorzuhalten.

Bei unzutreffender Qualifizierung von Daten kann die Finanzbehörde im Rahmen ihres pflichtgemäßen Ermessens verlangen, dass der Steuerpflichtige den Datenzugriff auf diese steuerlich relevanten Daten nachträglich ermöglicht. Das allgemeine Auskunftsrecht des Prüfers (§§ 88, 199 Abs.1 AO) und die Mitwirkungspflichten des Steuerpflichtigen (§§ 90, 200 AO) bleiben unberührt.

Bei der Ausübung des Rechts auf Datenzugriff stehen der Finanzbehörde nach dem Gesetz drei Möglichkeiten zur Verfügung. Die Entscheidung, von welcher Möglichkeit des Datenzugriffs die Finanzbehörde Gebrauch macht, steht in ihrem pflichtgemäßen Ermessen; falls erforderlich, kann sie auch mehrere Möglichkeiten in Anspruch nehmen:

- a) Sie hat das Recht, selbst unmittelbar auf das Datenverarbeitungssystem dergestalt zuzugreifen, dass sie in Form des Nur-Lesezugriffs Einsicht in die gespeicherten Daten nimmt und die vom Steuerpflichtigen oder von einem beauftragten Dritten eingesetzte Hard- und Software zur Prüfung der gespeicherten Daten einschließlich der Stammdaten und Verknüpfungen (Daten) nutzt (unmittelbarer Datenzugriff). Dabei darf sie nur mit Hilfe dieser Hard- und Software auf die elektronisch gespeicherten Daten zugreifen. Dies schließt eine Fernabfrage (Online-Zugriff) auf das Datenverarbeitungssystem des Steuerpflichtigen durch die Finanzbehörde aus.

Der Nur-Lesezugriff umfasst das Lesen, Filtern und Sortieren der Daten gegebenenfalls unter Nutzung der im Datenverarbeitungssystem vorhandenen Auswertungsmöglichkeiten.

- b) Sie kann vom Steuerpflichtigen auch verlangen, dass er an ihrer Stelle die Daten nach ihren Vorgaben maschinell auswertet oder von einem beauftragten Dritten maschinell auswerten lässt, um den Nur-Lesezugriff durchführen zu können (mittelbarer Datenzugriff). Es kann nur eine maschinelle Auswertung unter Verwendung der im Datenverarbeitungssystem des Steuerpflichtigen oder des beauftragten Dritten vorhandenen Auswertungsmöglichkeiten verlangt werden.
- c) Sie kann ferner verlangen, dass ihr die gespeicherten Unterlagen auf einem maschinell verwertbaren Datenträger zur Auswertung überlassen werden (Datenträgerüberlassung). Der zur Auswertung überlassene Datenträger ist spätestens nach Be-

standskraft der aufgrund der Außenprüfung ergangenen Bescheide an den Steuerpflichtigen zurückzugeben oder zu löschen.

2. Umfang der Mitwirkungspflicht nach §§ 147 Abs. 6 und 200 Abs. 1 Satz 2 AO

Der Steuerpflichtige hat die Finanzbehörde bei Ausübung ihres Rechts auf Datenzugriff zu unterstützen (§ 200 Abs. 1 AO). Im Einzelnen gilt folgendes:

- a) Beim unmittelbaren Datenzugriff (Abschnitt I Nr. 1 Buchstabe a) hat der Steuerpflichtige dem Prüfer die für den Datenzugriff erforderlichen Hilfsmittel zur Verfügung zu stellen und ihn für den Nur-Lesezugriff in das DV-System einzuweisen. Die Zugangsberechtigung muss so ausgestaltet sein, dass dem Prüfer dieser Zugriff auf alle steuerlich relevanten Daten eingeräumt wird. Sie umfasst u.a. auch die Nutzung der im DV-System vorhandenen Auswertungsprogramme. Enthalten elektronisch gespeicherte Datenbestände andere, z.B. steuerlich nicht relevante personenbezogene oder dem Berufsgeheimnis (§ 102 AO) unterliegende Daten, so obliegt es dem Steuerpflichtigen oder dem von ihm beauftragten Dritten, durch geeignete Zugriffsbeschränkungen sicherzustellen, dass der Prüfer nur auf steuerlich relevante Daten des Steuerpflichtigen zugreifen kann. Die Zugangsberechtigung hat auch die Nutzung der im DV-System vorhandenen Auswertungsprogramme zu umfassen.

Das Datenverarbeitungssystem muss die Unveränderbarkeit des Datenbestandes gewährleisten (§ 146 Abs. 4 AO; Abschnitt V des BMF-Schreibens zu den Grundsätzen ordnungsmäßiger DV-gestützter Buchführungssysteme (GoBS) vom 7. November 1995, BStBl I S. 738). Eine Veränderung des Datenbestandes und des Datenverarbeitungssystems durch die Finanzbehörde ist somit ausgeschlossen.

- b) Beim mittelbaren Datenzugriff (Abschnitt I Nr. 1 Buchstabe b) gehört zur Mithilfe des Steuerpflichtigen beim Nur-Lesezugriff (Abschnitt I Nr. 1 Buchstabe a Abs. 2) neben der Zurverfügungstellung von Hard- und Software die Unterstützung durch mit dem Datenverarbeitungssystem vertraute Personen. Der Umfang der zumutbaren Mithilfe richtet sich nach den betrieblichen Begebenheiten des Unternehmens. Hierfür können z.B. seine Größe oder Mitarbeiterzahl Anhaltspunkte sein.
- c) Bei der Datenträgerüberlassung (Abschnitt I Nr. 1 Buchstabe c) sind der Finanzbehörde mit den gespeicherten Unterlagen und Aufzeichnungen alle zur Auswertung der Daten notwendigen Informationen (z.B. über die Dateistruktur, die Datenfelder

sowie interne und externe Verknüpfungen) in maschinell auswertbarer Form zur Verfügung zu stellen. Dies gilt auch in den Fällen, in denen sich die Daten bei Dritten befinden.

3. Grundsatz der Verhältnismäßigkeit

Die Finanzbehörde hat bei Anwendung der Regelungen zum Datenzugriff den Grundsatz der Verhältnismäßigkeit zu beachten. Dies bedeutet u.a.:

- a) Bei vor dem 1. Januar 2002 archivierten Daten kann sie beim unmittelbaren Datenzugriff (Abschnitt I Nr. 1 Buchstabe a Abs. 1) und beim mittelbaren Datenzugriff (Abschnitt I Nr. 1 Buchstabe b) nicht verlangen, dass diese Daten für Zwecke ihrer maschinellen Auswertung (§ 147 Abs. 2 Nr. 2 AO i.V. mit § 147 Abs. 6 AO) nochmals in das Datenverarbeitungssystem eingespeist (reaktiviert) werden, wenn dies mit unverhältnismäßigem Aufwand für den Steuerpflichtigen verbunden wäre. Dies kommt z.B. in Betracht bei fehlender Speicherkapazität, nochmaliger Erfassung der Daten, Archivierung der Daten außerhalb des aktuellen Datenverarbeitungssystems, Wechsel des Hard- oder Software-Systems. Müssen hiernach die Daten nicht reaktiviert werden, braucht der Steuerpflichtige auch nicht die für eine maschinelle Auswertung der betreffenden Daten erforderliche Hard- und Software zur Verfügung zu halten, wenn sie nicht mehr im Einsatz ist. Dies gilt auch, wenn die Aufbewahrungsfrist (§ 147 Abs. 3 AO) noch nicht abgelaufen ist.

Diese für die maschinelle Auswertbarkeit der Daten erforderliche technische, organisatorische und zeitliche Einschränkung bezieht sich nicht auf die Pflicht des Steuerpflichtigen zur Lesbarmachung der Daten (§ 147 Abs. 2 Nr. 2 AO, § 147 Abs. 5 AO). Die Lesbarmachung muss während der ganzen Aufbewahrungsfrist sichergestellt sein.

- b) Bei nach dem 31. Dezember 2001 archivierten Daten ist beim unmittelbaren Datenzugriff (Abschnitt I Nr. 1 Buchstabe a Abs. 1) und beim mittelbaren Datenzugriff (Abschnitt I Nr. 1 Buchstabe b) die maschinelle Auswertbarkeit (§ 147 Abs. 2 Nr. 2 und Abs. 6 AO) in Form des Nur-Lesezugriffs (Abschnitt I Nr. 1 Buchstabe a Abs. 2) sicherzustellen. Im Falle eines Systemwechsels ist es nicht erforderlich, die ursprüngliche Hard- und Software vorzuhalten, wenn die maschinelle Auswertbarkeit auch für die nach dem 31. Dezember 2001, aber vor dem Systemwechsel archivierten Daten durch das neue oder ein anderes System gewährleistet ist.

- c) Für die Datenträgerüberlassung (Abschnitt I Nr. 1 Buchstabe c) kann die Finanzbehörde nicht verlangen, vor dem 1. Januar 2002 auf nicht maschinell auswertbaren Datenträgern (z.B. Mikrofilm) archivierte Daten auf maschinell auswertbare Datenträger aufzuzeichnen.

II. Prüfbarkeit digitaler Unterlagen

1. Elektronische Abrechnungen im Sinne des § 14 Abs. 4 Satz 2 UStG

Die qualifizierte elektronische Signatur mit Anbieter-Akkreditierung nach § 15 Abs. 1 des Signaturgesetzes ist Bestandteil der elektronischen Abrechnung^{*)}. Der Originalzustand des übermittelten ggf. noch verschlüsselten Dokuments muss jederzeit überprüfbar sein. Dies setzt neben den Anforderungen nach Abschnitt VIII Buchstabe b) Nr. 2 der GoBS (a.a.O.) insbesondere voraus, dass

- vor einer weiteren Verarbeitung der elektronischen Abrechnung die qualifizierte elektronische Signatur im Hinblick auf die Integrität der Daten und die Signaturberechtigung geprüft werden und das Ergebnis dokumentiert wird;
- die Speicherung der elektronischen Abrechnung auf einem Datenträger erfolgt, der Änderungen nicht mehr zulässt. Bei einer temporären Speicherung auf einem änderbaren Datenträger muss das DV-System sicherstellen, dass Änderungen nicht möglich sind;
- bei Umwandlung (Konvertierung) der elektronischen Abrechnung in ein unternehmenseigenes Format (sog. Inhouse-Format) beide Versionen archiviert und nach den GoBS mit demselben Index verwaltet werden sowie die konvertierte Version als solche gekennzeichnet wird;
- der Signaturprüf Schlüssel aufbewahrt wird;
- bei Einsatz von Kryptographietechniken die verschlüsselte und die entschlüsselte Abrechnung sowie der Schlüssel zur Entschlüsselung der elektronischen Abrechnung aufbewahrt wird;
- der Eingang der elektronischen Abrechnung, ihre Archivierung und ggf. Konvertierung sowie die weitere Verarbeitung protokolliert werden;
- die Übertragungs-, Archivierungs- und Konvertierungssysteme den Anforderungen der GoBS, insbesondere an die Dokumentation, an das interne Kontrollsystem, an das Sicherungskonzept sowie an die Aufbewahrung entsprechen;

^{*)} Anpassung des § 14 Abs. 4 Satz 2 UStG 1999 an das Signaturgesetz vom 16. Mai 2001 (BGBl. I S. 876) ist im Entwurf eines Steueränderungsgesetzes 2001 vorgesehen.

- das qualifizierte Zertifikat des Empfängers aufbewahrt wird.

2. Sonstige aufbewahrungspflichtige Unterlagen

- Bei sonstigen aufbewahrungspflichtigen Unterlagen i.S.d. § 147 Abs. 1 AO, die digitalisiert sind und nicht in Papierform übermittelt werden, muss das dabei angewendete Verfahren den GoBS entsprechen.
- Der Originalzustand der übermittelten ggf. noch verschlüsselten Daten muss erkennbar sein (§ 146 Abs. 4 AO). Die Speicherung hat auf einem Datenträger zu erfolgen, der Änderungen nicht mehr zulässt. Bei einer temporären Speicherung auf einem änderbaren Datenträger muss das Datenverarbeitungssystem sicherstellen, dass Änderungen nicht möglich sind.
- Bei Einsatz von Kryptographietechniken sind die verschlüsselte und die entschlüsselte Unterlage aufzubewahren.
- Bei Umwandlung (Konvertierung) der sonstigen aufbewahrungspflichtigen Unterlagen in ein unternehmenseigenes Format (sog. Inhouse-Format) sind beide Versionen zu archivieren und nach den GoBS mit demselben Index zu verwalten sowie die konvertierte Version als solche zu kennzeichnen.
- Wenn Signaturprüfchlüssel oder kryptographische Verfahren verwendet werden, sind die verwendeten Schlüssel aufzubewahren.
- Bei sonstigen aufbewahrungspflichtigen Unterlagen sind der Eingang, ihre Archivierung und ggf. Konvertierung sowie die weitere Verarbeitung zu protokollieren.

III. Archivierung digitaler Unterlagen

1. Originär digitale Unterlagen nach § 146 Abs. 5 AO sind auf maschinell verwertbaren Datenträgern zu archivieren. Originär digitale Unterlagen sind die in das Datenverarbeitungssystem in elektronischer Form eingehenden und die im Datenverarbeitungssystem erzeugten Daten; ein maschinell verwertbarer Datenträger ist ein maschinell lesbarer und auswertbarer Datenträger. Die originär digitalen Unterlagen dürfen nicht ausschließlich in ausgedruckter Form oder auf Mikrofilm aufbewahrt werden. Somit reicht die Aufzeichnung im COM-Verfahren (Computer-Output-Microfilm) nicht mehr aus. Diese Einschränkung gilt nicht, wenn die vor der Übertragung auf Mikrofilm vorhandenen Daten vorgehalten werden, die eine maschinelle Auswertbarkeit durch das Datenverarbeitungssystem gewährleisten. Nicht ausreichend ist auch die ausschließliche Archivierung in maschinell nicht auswertbaren Formaten (z.B. pdf-Datei).

Eine Pflicht zur Archivierung einer Unterlage i.S. des § 147 Abs. 1 AO in maschinell auswertbarer Form (§ 147 Abs. 2 Nr. 2 AO) besteht nicht, wenn diese Unterlage zwar DV-gestützt erstellt wurde, sie aber nicht zur Weiterverarbeitung in einem DV-gestützten Buchführungssystem geeignet ist (z.B. Textdokumente).

2. Originär in Papierform angefallene Unterlagen, z.B. Eingangsrechnungen, können weiterhin mikroverfilmt werden.
3. Kann im Falle eines abweichenden Wirtschaftsjahrs die Archivierung ab 1. Januar 2002 nachweisbar aus technischen Gründen nicht auf einem maschinell auswertbaren Datenträger (§ 147 Abs. 2 Nr. 2 AO) erfolgen, wird dies nicht beanstandet, wenn der Steuerpflichtige bis spätestens zu Beginn des anschließenden abweichenden Wirtschaftsjahrs den Archivierungspflichten gemäß § 147 Abs. 2 Nr. 2 AO nachkommt.

IV. Anwendung

1. Die Regelungen zum Datenzugriff (Abschnitt I) sind bei steuerlichen Außenprüfungen anzuwenden, die nach dem 31. Dezember 2001 beginnen.
2. Die Regelungen zur Prüfbarkeit digitaler Unterlagen (Abschnitt II) gelten
 - a) für elektronische Abrechnungen mit Inkrafttreten des § 14 Abs. 4 Satz 2 UStG (1. Januar 2002) und
 - b) für sonstige aufbewahrungspflichtige Unterlagen, die nach dem 31. Dezember 2001 erstellt werden.

Im Übrigen bleiben die Regelungen des BMF-Schreibens zu den Grundsätzen ordnungsmäßiger DV-gestützter Buchführungssysteme (GoBS) vom 7. November 1995 (BStBl I S. 738) unberührt.

Dieses Schreiben wird im Bundessteuerblatt Teil I veröffentlicht.

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 25. März 2009

Ort, Datum

Unterschrift