

Sabeur Bouchnak

Servergestütztes Qualitätsmanagement bei GSM-  
basierter Zählerauslesung

Diplomarbeit eingereicht im Rahmen der Diplomprüfung  
im Studiengang Informations- und Elektrotechnik  
Studienrichtung Informationstechnik  
am Department Informations- und Elektrotechnik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Ing Franz Schubert  
Zweitgutachter : Prof. Dr. Gustav Vaupel

Abgegeben am .12 Mai 2008

**Sabeur Bouchnak**

**Thema der Diplomarbeit**

Servergestütztes Qualitätsmanagement bei GSM-basierter Zälerauslesung

**Stichworte**

Qualitätsmanagement, Netzwerkmanagement, JAVA, EXCEL, ACCESS, VBA, Pivot-Table, Datenbank, GSM, LAN

**Kurzzusammenfassung**

Diese Arbeit befasst sich mit der Qualitätssicherung bei der Auslesung von Lastdaten in einem Forschungsprojekt im Bereich des Lastmanagements. Für die Qualitätssicherung wurden anhand der Standard des Netzwerkmanagement nach ISO 7498 - 4 Qualitätsmanagementvorschläge für den LAN-Bereich des Forschungsprojekts und Qualitätsmanagementlösungen für den GSM-Bereich des Projekts entwickelt.

**Sabeur Bouchnak**

**Title of the paper**

Development and Construction of a Microprocessor controlled allocation processor

**Keywords**

Quality Management, Network Management, JAVA, EXCEL, ACCESS, VBA, Pivot-Table, Database, GSM, LAN

**Abstract**

This thesis describes quality assurance procedures used for meter reading processes in the R&D project "e-island". Based on the ISO 7498 – 4 standard suggestion for quality management procedures of the LAN part and solutions quality management procedures for the GSM part of the network have been developed.

End of text

## Inhaltsverzeichnis

1	Einleitung.....	5
1.1	Das Forschungsprojekt INSEL .....	6
1.2	Ziel dieser Diplomarbeit.....	7
2	Qualitätsmanagement in einem Netzwerk (Netzwerkmanagement) .....	9
2.1	Funktionsbereiche des Netzwerkmanagements.....	9
2.1.1	Konfigurationsmanagement .....	9
2.1.2	Fehlermanagement .....	10
2.1.3	Leistungsmanagement.....	10
2.1.4	Sicherheitsmanagement.....	10
2.1.5	Abrechnungsmanagement .....	11
2.2	Relationalität der fünf Managementfunktionsbereiche und deren Umsetzung	
	11	
3	Netzwerkmanagementtools und Einsatzgebiete.....	13
3.1	Zu überwachende kritische Netzwerkelemente.....	13
3.2	Netzwerkmanagementprodukte für kleine Unternehmen .....	14
4	Qualitätsmanagement bei der GSM-basierten Zählerauslesung.....	18
4.1	Dokumentation der Liegenschaften.....	20
4.1.1	Begründung der Notwendigkeit einer guten Dokumentation .....	20
4.1.2	Struktur und Inhalt der Dokumentation.....	20
4.2	Qualitätssicherung.....	27
4.2.1	Bestimmung der Einsatzgebiete.....	27
4.2.2	Analyse der Ereignisreports .....	29
4.2.3	Feststellung der Verbindungs- und Gesamtzeit .....	29
4.2.4	Fehleranalyse.....	31
4.2.4.1	PPP-Fehler: .....	31
4.2.4.2	WGET-Fehler: .....	33
4.2.4.3	Database-Fehler:.....	34
5	Qualitätsmanagement-Tool (Insel) .....	37
5.1	Vorbereitungs-Tool (JAVA).....	37
5.1.1	Auswahl der Programmiersprache.....	37
5.1.2	Prinzip des JAVA-Vorbereitungs-Tools .....	38
5.1.3	Aufbau und Ablauf des JAVA-Programms .....	38
5.1.4	Fehlerbehandlung (Exceptions) .....	45
5.2	Auswertungstool.....	46
5.2.1	Was ist eine Datenbank? .....	47
5.2.2	Anforderung und Regeln für eine Datenbank.....	47
5.2.2.1	Mindestanforderungen an relationale Datenbanken .....	47
5.2.2.2	Regeln der Relationalität .....	48
5.2.3	Anpassungen für die Datenbank Insel .....	50
5.2.4	Aufbau der Datenbank Insel.....	51
5.2.4.1	Verknüpfungen .....	54
5.2.4.2	Tabellen.....	55

5.2.4.3	Abfragen .....	57
5.2.4.4	Formulare .....	59
5.2.4.4.1	Pivot-Tabellen .....	59
5.2.4.4.2	Formulare der Datenbank Insel .....	61
5.2.4.5	Module (VBA) .....	69
5.3	Analyse der Ergebnisdaten des Auswertungstools .....	72
6	Zusammenfassung und Ausblick .....	73
6.1	Zusammenfassung .....	73
6.2	Ausblick .....	74
7	Anhang .....	75
7.1	Literaturverzeichnis .....	75
7.2	Abbildungsverzeichnis .....	79
7.3	Tabellenverzeichnis .....	80
7.4	Source Code des JAVA-Programms .....	81
7.5	Source Code des VBA-ACCESS-Programms .....	90
7.6	Excel-Liegenschaftsdokumentation .....	93
7.7	JAVA-Projektdateien (Vorbereitungstool) .....	93
7.8	ACCESS-Datenbanktool (Auswertungstool) .....	93
7.9	Erklärung der Selbständigkeit .....	93

# 1 Einleitung

In Zeiten der globalen Erderwärmung, des Klimaschutzes und der wachsenden Anforderungen an den Wohlstand, der meistens mit steigendem Energieverbrauch verknüpft ist, steigen auch die Bemühungen bei der Suche nach alternativen Energiequellen, die einerseits klima- und umweltverträglich und andererseits kostengünstig sind. Weltweit erfreut sich die Nutzung regenerativer Energien in ihren verschiedenen Formen einer großen Beliebtheit, so lag der Beitrag der weltweiten Stromerzeugung aus regenerativen Energien (Geothermie, Wind, Wasserkraft, Biomasse u. a.) im Jahr 2005 insgesamt 3,3 TWh. Allein in Deutschland stieg der Anteil des aus erneuerbarer Energie erzeugten Stroms von 1990 bis Ende 2006 von 18,463 auf 70,433 TWh [1].

Abbildung 1 zeigt die Entwicklung der Stromerzeugung aus erneuerbaren Energien 1990 bis 2006.

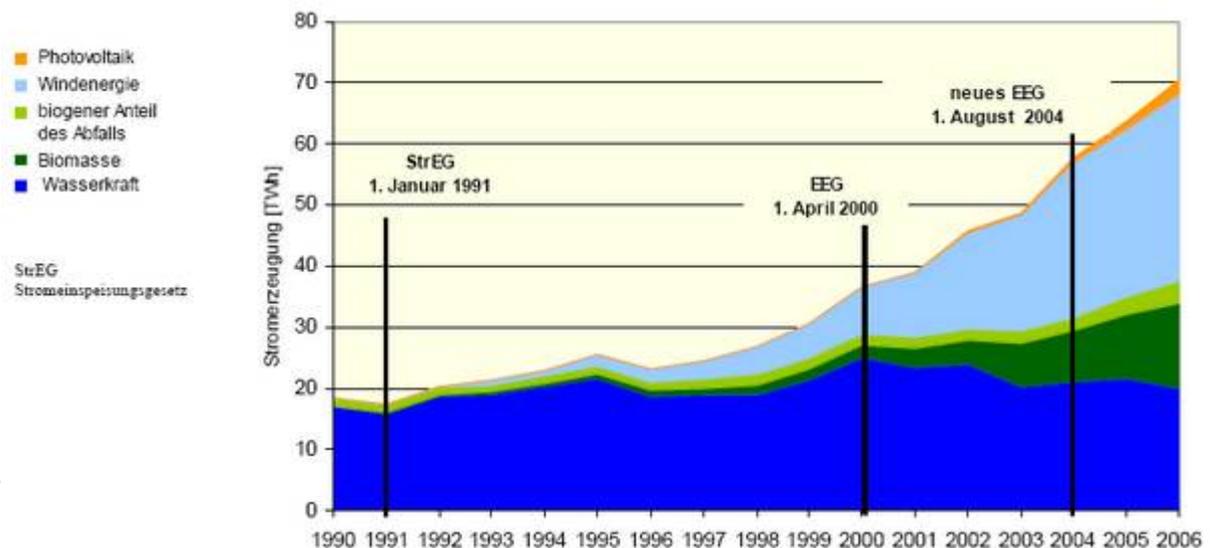


Abbildung 1: Die Entwicklung der Stromerzeugung aus erneuerbaren Energien 1990 bis 2006 [1].

Dieser steigende Anteil an regenerativen Energien hat eine Senkung des Grundlastanteils der durch die konventionellen Kraftwerke zu deckenden Last zur Folge und es kann auch zur Überschusserzeugung kommen. Durch Lastmanagement lässt sich jedoch die Integration regenerativer Energien in das Netz erheblich verbessern. Ein weiteres Problem des steigenden Anteils erneuerbarer Energien an der Elektrizitätsversorgung ist auch die Zunahme an Schwankungen der durch konventionelle Kraftwerke zu deckenden Last. Aus diesem Gründen gewinnt der Bereich des optimierten Lastmanagement in Zukunft verstärkt an Bedeutung. [2]

Das Forschungsprojekt **Internetbasiertes System** eines erweiterbaren Lastmanagements zur Integration in virtuelle Kraftwerke, kurz **INSEL**, zu dem auch diese Diplomarbeit zum Teil gehört, ist eines der zahlreichen Forschungsprojekte zum Thema Lastmanagement.

## **1.1 Das Forschungsprojekt INSEL**

Das Insel-Projekt ist ein mit 220.000 € vom Bundesministerium für Bildung und Forschung (BMBF) gefördertes Forschungsprojekt, das seit dem 1. September 2006 bis voraussichtlich Ende August 2009 andauert.

Die Forschungspartner in diesem Projekt sind:

- Die Behörde für Wirtschaft und Arbeit der Freien und Hansestadt Hamburg,
- die Evonik AG (früher Steag Saar Energie AG),
- das Ingenieurbüro SUmBi und
- das Ingenieurbüro ENVIDATEC GmbH.

Das Projekt verfolgt als Hauptziel die Schaffung eines virtuellen Verbunds aus 40 der lastintensivsten Liegenschaften der Freien und Hansestadt Hamburg, der dann als ein Großverbraucher betrachtet werden kann und mit der Leitwarte des virtuellen Kraftwerks der Evonik verbunden werden soll. Dieser Verbund soll ein Netzwerk von Lastmanagement Anlagen bilden (als Simulation vorgesehen) aus dem dann die gemeinsam erreichte Abschaltungsleistung als Minutenreserve an das virtuelle Kraftwerk der Evonik verkauft werden soll.

Ein zweites Ziel dieses Projektes ist es, die Möglichkeit für die Liegenschaften bzw. für die Stadt Hamburg (vertreten durch die Behörde für Stadtentwicklung und Umwelt (BSU)) zu schaffen, künftig günstigere Konditionen bei der Verhandlung von neuen Stromlieferverträgen zu erwirtschaften, wenn das Summenlastverhalten der öffentlichen Liegenschaften bekannt wird.

Für die Realisierung dieses Vorhabens werden in einer ersten Phase die lastintensivsten Liegenschaften mit Kommunikationsmodulen „VIDA 84“ ausgestattet, um die aktuellen hochaufgelösten Lastgangsdaten der einzelnen Liegenschaften zeitnah (einmal pro Tag) einzusammeln und daraus das Potenzial der abschaltbaren Leistung abzuschätzen.

Die VIDAs 84 sollen die Lastprofile mit einer Auflösung von Minutenschritten aufzeichnen und einmal am Tag an einen Datenbank-Server übertragen.

Abbildung 02 zeigt die vorgesehene Systeminfrastruktur des Projekts Insel.

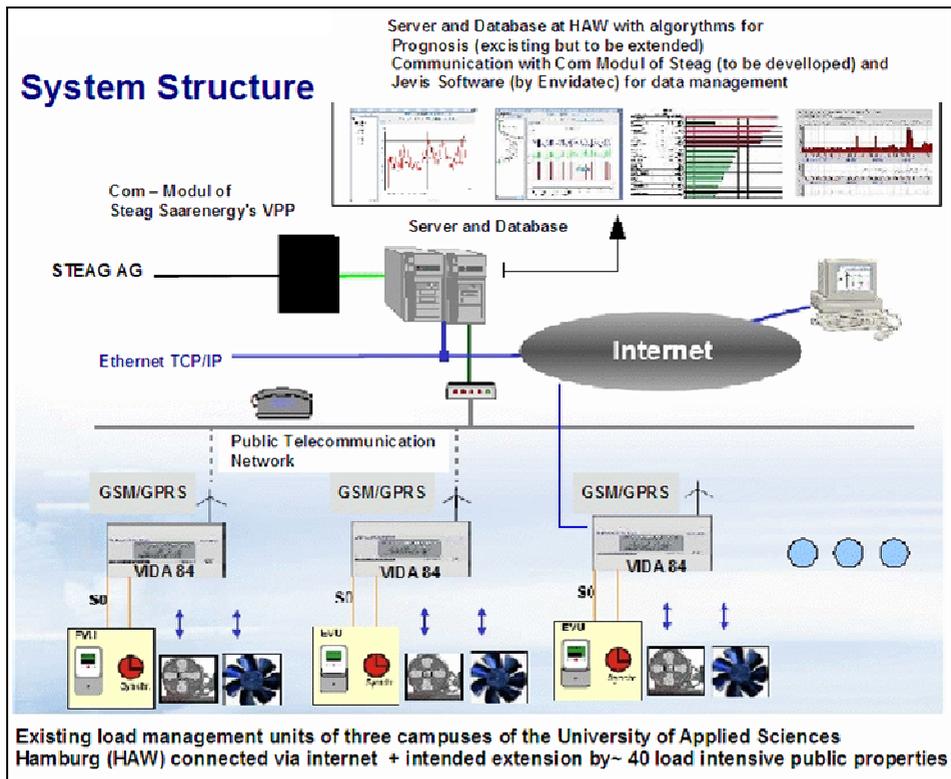


Abbildung 2: Systeminfrastruktur des Projekts Insel[3].

In den weiteren Phasen des Projekts wird durch Prognosetools und Simulationsprogramme ermittelt, wie viel freie Leistung zu welchem Zeitpunkt als Reserveleistung an das Virtuelle Kraftwerk der Evonik angeboten werden kann [4].

## 1.2 Ziel dieser Diplomarbeit

Diese Diplomarbeit ist ein Teil der ersten Phase des Projekts, die wiederum sehr wichtig für den weiteren Verlauf des Projekts ist. In dieser Phase werden die Daten (Lastverlauf, Schaltvorgänge, Umweltdaten) durch die VIDA 84 gesammelt, die die Grundlage sind für die späteren Simulationen des Lastverhaltens der einzelnen Liegenschaften und die Prognosen der möglichen Lastreduktion seitens der Liegenschaften bzw. der möglichen Leistung, die als Reserveleistung an das Virtuelle Kraftwerk abgegeben werden kann. Dieser Prozess beinhaltet die Installation der Datenerfassungsgeräte in den einzelnen Liegenschaften und die Bereitstellung des Datenservers, in dem die gesammelten Daten gespeichert werden sollen.

Ziel dieser Diplomarbeit ist die Sicherung dieses Prozesses und die kontinuierliche Überwachung bzw. die Auswertung des Kommunikationsablaufs, angefangen mit der Dokumentation aller Daten, die mit den Kommunikationsmodulen und dem Prozess der Installation zu tun haben (Kapitel 4.1) bis zur Überwachung und Sicherung des reibungslosen Verlaufs des Prozesses der Übertragung der erfassten Daten in den Datenserver (Kapitel 3 und 4). Darüber hinaus stellt diese Diplomarbeit ein Konzept zur Qualitätssicherung für das Management und die Evaluierung der Effektivität des Kommunikationsnetzwerks zur Verfügung (Kapitel 5). Sie bietet mit Hilfe des

Auswertungstools, Möglichkeiten zur Kontrolle der Verbindungsmodalitäten und zur Analyse möglicher Probleme beim Prozess der Datenübertragung, die die Fehlerbehebung erleichtern.

Zusätzlich kann das für die Auswertung eingesetzte Tool bei der kontinuierlichen Überprüfung der aufgezeichneten Daten benutzt werden.

In den folgenden Kapiteln soll zunächst anhand der Netzwerkmanagement-Standards nach [ISO 7498-4] der theoretische Aspekt des Qualitätsmanagements dargestellt und die daraus für das Netzwerk des Insel-Projekts zutreffenden Bereiche umgesetzt werden. Dabei liegt der Fokus auf dem GSM-Teilnetzwerk, für das ein Qualitätssicherungstool entwickelt wird. Für den übrigen Teil des Netzwerks, den LAN-Bereich, werden geeignete Produkte präsentiert und auf deren Vorteile und Einsatzgebiete hingewiesen.

## **2 Qualitätsmanagement in einem Netzwerk (Netzwerkmanagement)**

PC-Netzwerke sind mittlerweile eines der wichtigsten Hilfsmittel zur Erreichung der Geschäftsziele von Unternehmen. Es ist daher notwendig, dass die durch IT bereitgestellten Informationen den Ansprüchen der Geschäftsprozesse genügen, damit die Geschäftsziele erreicht werden können. Das heißt aber auch, wenn das Netz nicht betriebsbereit ist, steht das Unternehmen zu meist still [5].

Um die Stabilität und die geforderte Dienstgüte (Quality of Service) einhalten zu können, müssen die Geräte im Netzwerk abgefragt werden. Die Kontrolle des Status und die Beobachtung der Auslastung unterschiedlicher Komponenten müssen die Möglichkeit ergeben, das gesamte LAN-System im laufenden Betrieb zu warten. Die Problematik liegt hier bei der Heterogenität des Netzwerks in Zeiten, in denen LANs und WANs immer komplexer werden.

In einem heterogenen Netzwerk kommen oftmals verschiedene Komponenten unterschiedlicher Hersteller zum Einsatz, was eine herstellerübergreifende Lösung erfordert. Diese Problematik war früher nicht stark zu spüren, da fast alle installierten Netzwerkkomponenten vom selben Hersteller stammten [6, 8].

Damit ein herstellerübergreifendes Management im Netzwerk gewährleistet wird, müssen Standards beachtet werden. Die Standards für das Netzwerkmanagement sind im OSI (Open Systems Interconnect) Management Framework festgelegt [ISO 7498-4] [7]. Daraus gehen fünf Managementbereiche hervor.

### **2.1 Funktionsbereiche des Netzwerkmanagements**

#### **2.1.1 Konfigurationsmanagement**

Das Konfigurationsmanagement dient im weitesten Sinne dazu, die im Netz vorhandenen Komponenten zu überwachen, deren Bestandteile und Einstellungen zu kontrollieren und zu verändern. Es umfasst eigentlich alle Funktionen die im Zusammenhang mit Konfigurationsdaten stehen: sammeln und darstellen, kontrollieren sowie aktualisieren von bestimmten Konfigurationsparametern. Dazu gehören auch folgende Aspekte [8, 9]:

- Existenz und Namen von Netzkomponenten
- Technische Daten von Netzkomponenten
- Beziehungen zwischen Netzkomponenten
- Status (aktiv/inaktiv) von Netzkomponenten
- Adressierungen
- Routing-Kontrolle

### 2.1.2 Fehlermanagement

Beim Fehlermanagement geht es darum, die Verfügbarkeit des Netzwerks zu erhöhen. Durch die ständige Beobachtung des Netzes und der angeschlossenen Systeme werden Änderungen in den Netzparametern erkannt und daraus Rückschlüsse auf die zu erwartenden Fehler gezogen. Die wichtigsten Aufgabenbereiche des Fehlermanagements sind:

- **Fehlererkennung:**

Hierzu werden in Intervallen Diagnosetests durchgeführt, auch Pollingverfahren genannt, oder Fehlermeldungen von einem Managed Objekt an den Manager gesendet, auch Trappingverfahren genannt.

- **Fehlerdiagnose:**

Um die Art und die Ursache des Fehlers zu erforschen, müssen die Aktivitäten der Objekte aus den Ereignis- und Fehler-Reports (Error Log) analysiert werden.

- **Fehlerbehebung:**

Hierzu bedient sich das Fehlermanagement der Dienste des Konfigurations-Management, startet passende Fehler-Behebungsprogramme oder fordert einen aktiven Eingriff des Administrators [8, 10].

### 2.1.3 Leistungsmanagement

Beim Leistungsmanagement wird die Optimierung der Leistungsfähigkeit des Netzes angestrebt, dazu wird die Systembelastung ermittelt und Leistungsengpässe werden angezeigt. Es werden die ständig über Polling und Trapping gesammelten Informationen genutzt, um eine benutzerdefinierte Anzeige und Auswertung zu ermöglichen. Hierzu muss der Administrator in der Lage sein, selbstständig Grenzwerte und Pollingfrequenzen zu bestimmen und Leistungsreports über alle wichtigen Variablen zu generieren.

Für das Leistungsmanagement werden typischerweise Echtzeitstatistiken, historische Statistiken und Pegelanzeiger mit Schwellwerten für die automatische Alarmierung benötigt [8, 11].

### 2.1.4 Sicherheitsmanagement

Das Sicherheitsmanagement hängt unmittelbar mit der Zielspezifikation der Benutzerverwaltung zusammen und soll vor unberechtigtem Zugriff auf Datennetze und deren Ressourcen schützen. Dazu gehören Strategien gegen nicht autorisierten Datenempfang, Datenverfälschung bei der Übertragung und Zugriffsberechtigungen auf Netze und LAN-Segmente. Dabei müssen viele sicherheitspolitische, ethische, rechtliche sowie wirtschaftliche Aspekte berücksichtigt werden.

Es gibt zahlreiche Aspekte, die nicht von einer Managementnorm abgedeckt werden können. Die Ausführungen dieses Standards weisen für viele Experten noch große grundsätzliche Lücken auf [8, 12].

## 2.1.5 Abrechnungsmanagement

Mit dem Abrechnungsmanagement ist es möglich, die mit dem Netzwerk erbrachten Leistungen benutzerbezogen identifizieren und abrechnen zu können, zum Beispiel wer wann und für welchen Zeitraum welche Netzwerkressourcen benutzt hat. Es ermöglicht auch die Setzung von Limits, nach deren Überschreitung der Zugang zum Netz eingeschränkt wird.

Anwendungsbeispiele für das Abrechnungsmanagement sind Billing-Funktion und Tariffing-Funktion [8, 13].

## 2.2 Relationalität der fünf Managementfunktionsbereiche und deren Umsetzung

In der Praxis ist es üblich, dass sich die Aufgaben der fünf Bereiche überschneiden, so benötigt beispielsweise das Fehlermanagement Dienste des Konfigurationsmanagements. Deshalb ist es meistens schwierig, die Funktionen der Management-Bereiche strikt von einander zu trennen.

Abbildung 3 zeigt die fünf Managementfunktionsbereiche in Abhängigkeit von einander.

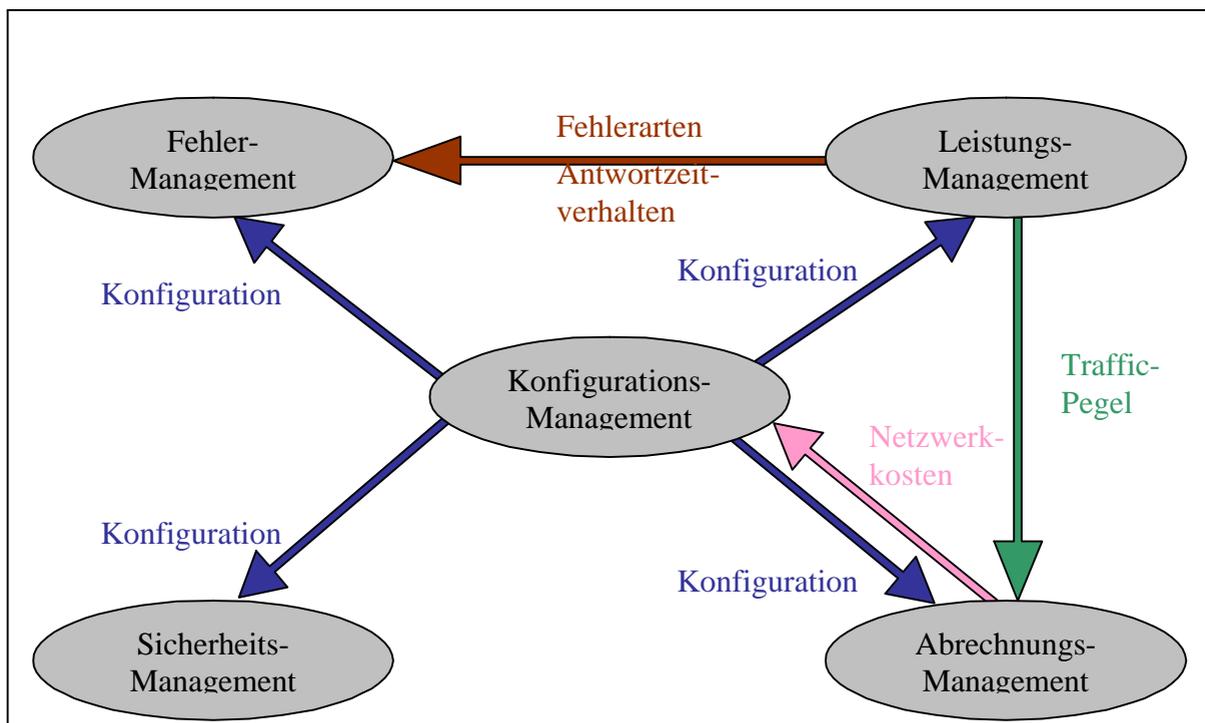


Abbildung 3: Die fünf Managementfunktionsbereiche in Abhängigkeit von einander[14].

Es existieren außer den OSI-Managementfunktionsbereichen weitere, weniger relevante Bereiche wie zum Beispiel das Bestandsführungsmanagement, das Problemmanagement und die System-Administration, auf die hier nicht weiter eingegangen wird (weiterführend dazu sei verwiesen auf: [15, 16]).

Für die Umsetzung der Netzwerkmanagement-Standards nach dem OSI-Management Framework Modell werden im Insel-Projekt nur die drei ersten Funktionsbereiche des Netzwerkmanagements betrachtet und umgesetzt.

Bei den drei Funktionsbereichen des Netzwerkmanagements handelt es sich um „Konfigurationsmanagement“, „Fehlermanagement“ und „Leistungsmanagement“. Hierbei wird zunächst auf den Stand der Technik und die existierenden Produkte eingegangen, die für das Netzwerk des Insel-Projekts eingesetzt werden können. Des Weiteren werden anhand der drei wichtigsten Funktionsbereiche des Netzwerkmanagements Qualitätssicherungskonzepte für die Bereiche erarbeitet, die von den auf dem Markt existierenden Produkten nicht abgedeckt werden können.

„Sicherheitsmanagement“ und „Abrechnungsmanagement“ sind die zwei letzten Bereiche, die nicht Thema der Diplomarbeit sind und somit nicht näher betrachtet werden.

## 3 Netzwerkmanagementtools und Einsatzgebiete

Waren die meisten Netzwerkmanagementanbieter früher der Meinung, dass ihre Managementsysteme für mittelständische und kleine Unternehmen nicht geeignet sind, so hat sich die Lage in den letzten Jahren geändert. Eine wachsende Anzahl von Netzwerkmanagement-Anbietern stellt ihre Software nun auch mittelständischen und kleinen Kunden zur Verfügung [17]. Mittlerweile existiert eine große Anzahl von Open-Source und kommerziellen Netzwerkmanagementtools, die die grundlegenden kritischen Elemente der Netzwerk-Infrastruktur überwachen und managen. Diese Netzwerkelemente werden im folgenden Abschnitt kurz erläutert.

### 3.1 Zu überwachende kritische Netzwerkelemente

- **E-Mail-Server:** Jede Organisation verfügt über einen E-Mail-Server, der E-Mails an alle LAN-Benutzer verteilt. Wenn der E-Mail-Server ausfällt, werden die Benutzer von der Außenwelt getrennt und die wichtigen Funktionen wie z. B. der Kunden-Support fallen aus. Für diesen Bereich sind Netzwerkmanagementtools also wichtig, um die Verfügbarkeit des E-Mail-Servers, die E-Mails in der Warteschlange, die Größe der empfangenen E-Mails usw. zu überwachen [18].
- **WAN-Verbindungen:** Kleine Unternehmen können durch die optimale Ausnutzung von WAN-Verbindungen Kosten sparen, denn eine zu hohe Verkehrsleistung steigert die Kosten, eine zu niedrige Verkehrsleistung dagegen würgt das Netz ab. Mit Hilfe des Netzwerkmanagementtools kann der Administrator eine höhere Netzwerkleistung erzielen und feststellen, welcher Nutzer die meiste Bandbreite verbraucht, um die erforderlichen Maßnahmen zu treffen. Außerdem ermöglichen die Managementtools den Router in regelmäßigen Abständen auf Verfügbarkeit und Performance zu überwachen, weil beim Ausfall des Routers das gesamte Netzwerk ebenfalls ausfällt [18].
- **Servers & Services:** Im Server laufen kritische Anwendungen, die unbedingt auf Auslastung von CPU, Speicher (RAM), Speicherplatz, Diensten (FTP, DNS, ECHO, IMAP, LDAP, TELENET, http, POP, etc.) und ihre Reaktionszeiten überwacht werden müssen [18].
- **Server-Logs:** In Unternehmen sollten auch die Server-Logs von laufenden Rechnern auf fehlgeschlagene Anmeldeversuche, Kontosperrungen, falsche Passwörter, gescheiterte Versuche zur Sicherung von Dateien, Sicherheitsprotokolle, Manipulationen usw. untersucht und überwacht werden. Die Überwachung dieser Logs ergibt ein klares Bild über die Sicherheitslücken im Unternehmen [18].
- **Anwendungen & Webseiten:** In einem Unternehmen laufen mehrere aufgabenkritische Anwendungen und Webseiten, die unbedingt in regelmäßigen Abständen überwacht werden müssen. Anwendungen und URLs müssen verfügbar und reaktionsschnell sein [18].

### **3.2 Netzwerkmanagementprodukte für kleine Unternehmen**

Kleine Unternehmen haben unterschiedliche Netzwerkmanagementbedürfnisse und -erwartungen, weil sie nur begrenzt über technisches Know-how und geschultes Personal verfügen. Deshalb brauchen sie Produkte die kostengünstig, leicht zu installieren und zu verwenden sind und mehrere Optionen bieten.

Open Source bietet viele Werkzeuge für verschiedene IT-Bedarfe, einschließlich Netzwerküberwachung, Überwachung der Bandbreite, Netzwerkerkennung etc. Zu den bekanntesten Open Source Managementtools zählen Nagios, Open NMS, JFNMS, BigBrother, BigSister, Genesysn, NetDisco, etc.

Nagios (Network + Hagios) - eines der beliebtesten Tools- kann den Status verschiedener Dienste (zum Beispiel SSH, FTP, HTTP) sowie den Festplattenplatz, Speicher- und CPU-Auslastung, Uptime usw. über diverse Module (Plug-ins) abfragen und auswerten. Da einige Testmethoden auf Protokollebene arbeiten (TCP, UDP, SNMP, ...), ist es möglich, verschiedene Betriebssysteme zu überwachen. Für speziellere Aufgaben kommen weitere Programme zum Einsatz, die ebenfalls frei verfügbar sind (NC\_Net, NSClient). Mit entsprechender Zusatzhardware ist es sogar möglich, Umweltbedingungen (zum Beispiel Temperaturwerte, Luftfeuchtigkeit, Füllstände von Flüssigkeitstanks, ...) zu überwachen. Sobald ein Dienst oder ein Host einen kritischen Wert erreicht oder gar nicht mehr verfügbar bzw. erreichbar ist, alarmiert Nagios die Kontaktpersonen über beliebige Kanäle (zum Beispiel E-Mail, SMS, Pager, IM-Messages, Telefon, ...). Dabei besteht auch die Möglichkeit, festzulegen, in welcher Reihenfolge Meldungen an weitere Kontaktpersonen erfolgen sollen, wenn eine Störung nach den ersten Meldungen nicht behoben wurde (Eskalationsmanagement). Ebenfalls können bei der Überwachung der Dienste untereinander bestehende Abhängigkeiten berücksichtigt werden. Wird etwa die Erreichbarkeit eines Rechners und ein auf ihm laufendes Programm überwacht, so wird bei einem Ausfall des gesamten Rechners die Meldung über das nicht mehr laufende Programm unterdrückt [19, 20].

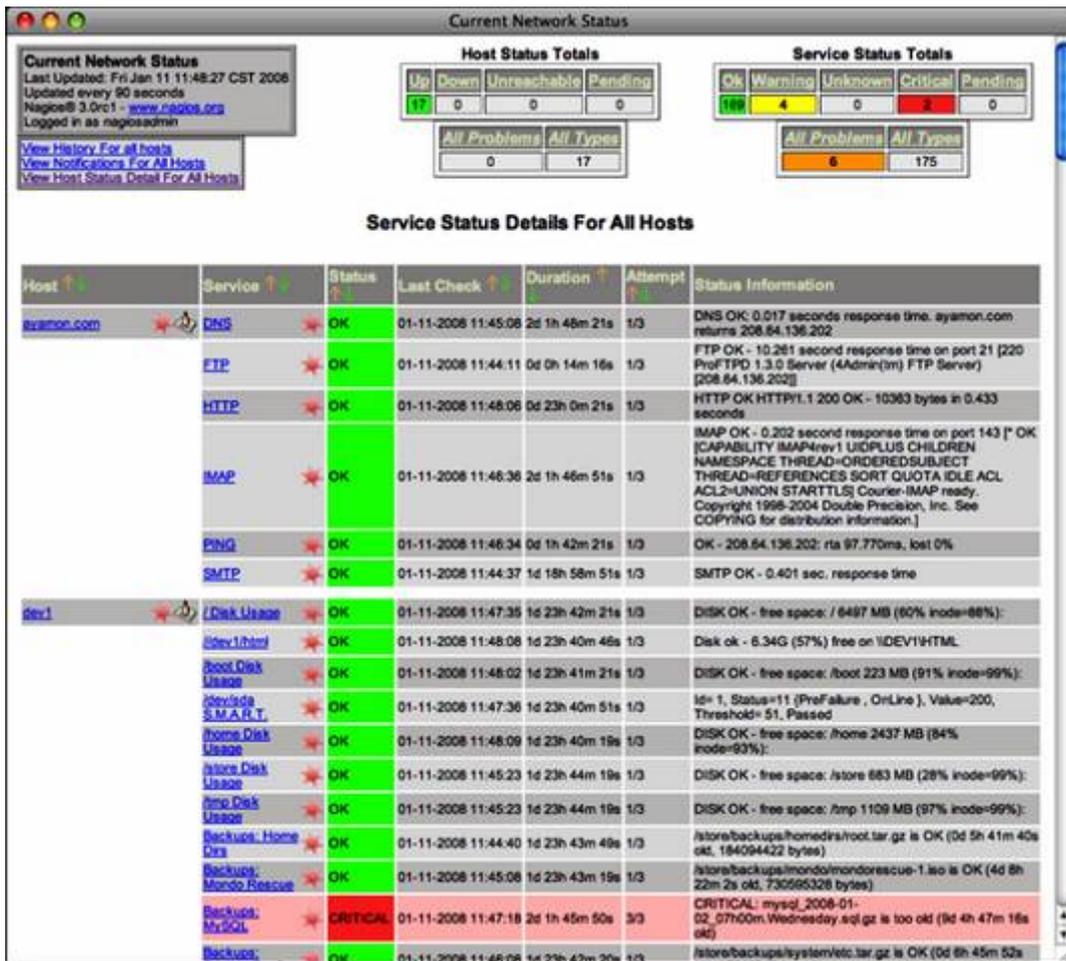


Abbildung 4: Screenshot\_1 aus dem Netzwerkmanagementprodukt Nagios [19].

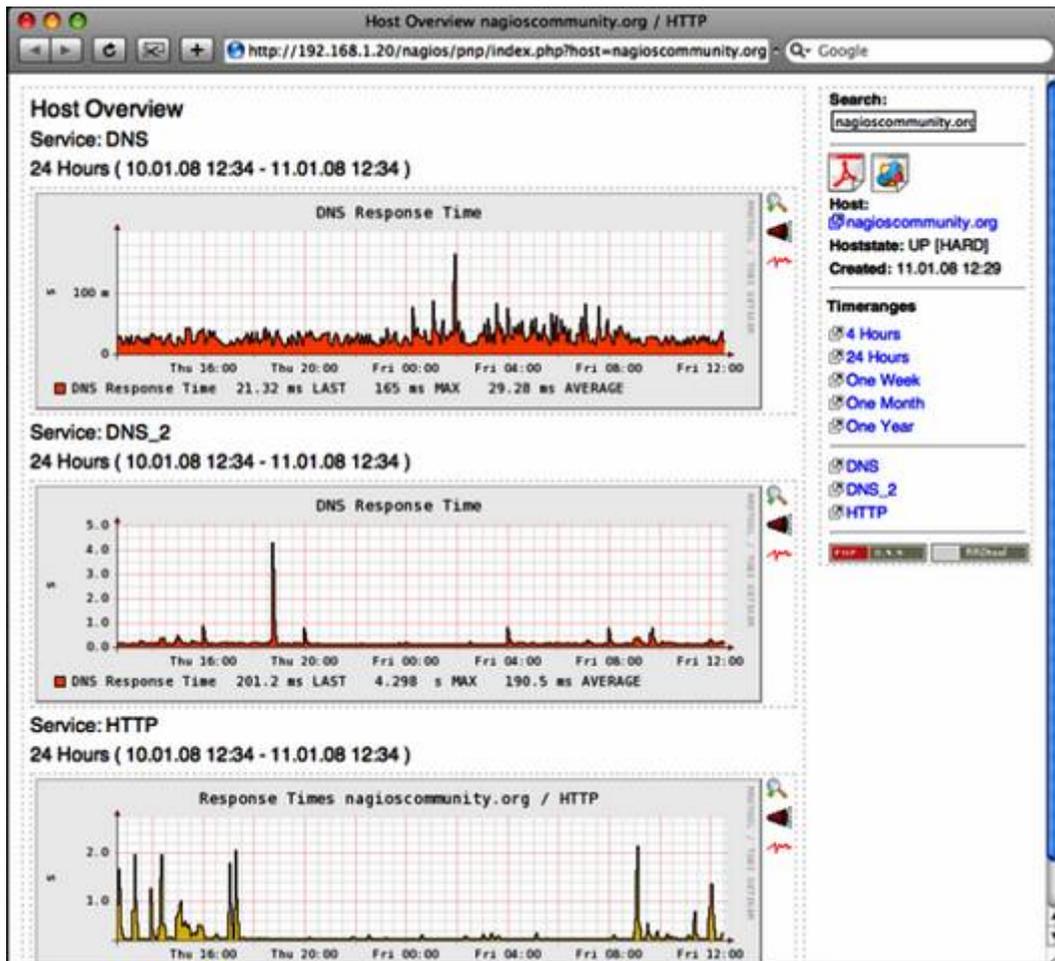
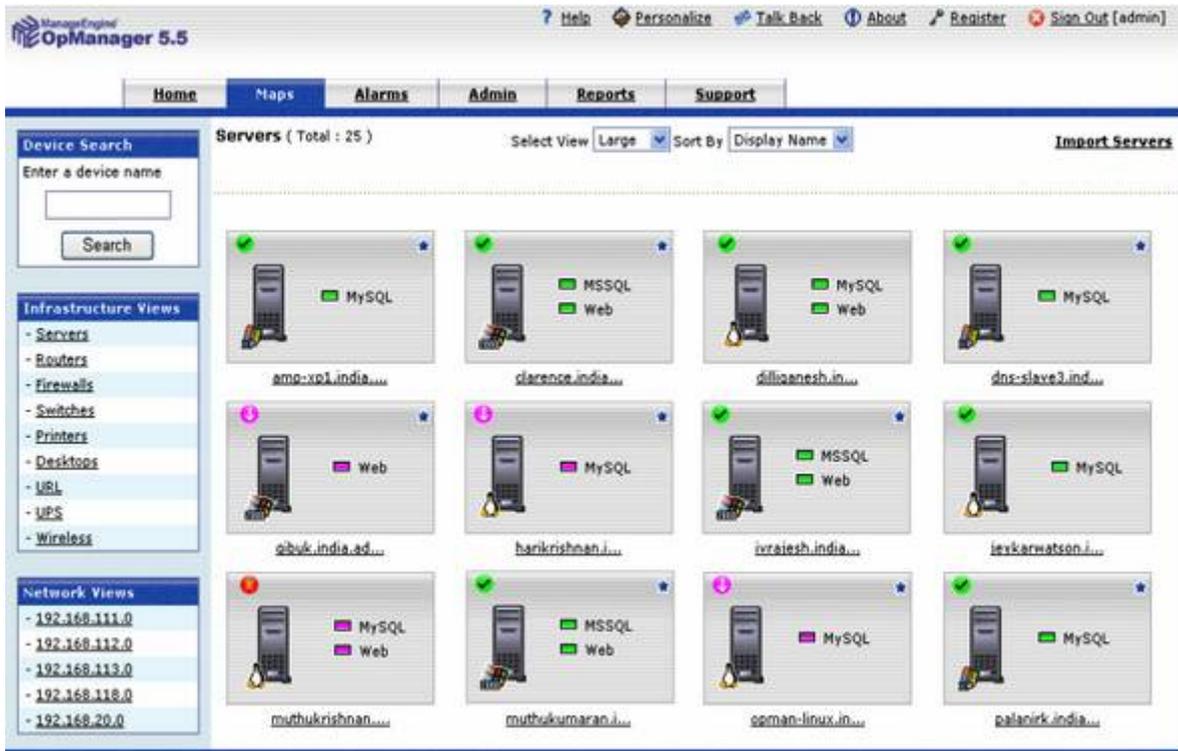


Abbildung 5: Screenshot\_2 aus dem Netzwerkmanagementprodukt Nagios [19].

Eine weitere Alternative zu den Open-Source-Tools ist die Nutzung der Freeware-Versionen von kommerzieller Netzwerkmanagementsoftware wie ManageEngine, AdventNet, KIWI Cat-Tools, SysUpTime Network Monitor etc. Die wesentlichen Vorteile dieses Ansatzes liegen auf der Hand: Kleine und mittlere Unternehmen erhalten eine Software mit vergleichbarem Qualitätsstandard, wie sie von großen Unternehmen verwendet wird. Diese Freeware-Software ist auch einfach zu installieren und zu verwenden im Gegensatz zu den Open-Source Lösungen. Der einzige Nachteil ist, dass es einige Einschränkungen für die Anzahl der zu verwaltenden Geräte gibt. Dies hat jedoch den Vorteil, dass man mit den Freeware-Versionen anfangen und später, wenn das Unternehmen oder das Netzwerk wächst, bequem mit einem Update auf die Vollversion umsteigen kann.

OpManager von ManageEngine ist eine der beliebtesten Netzwerk-Überwachungssoftware und wird von zahlreichen IT-Administratoren weltweit benutzt. Mit der Gratisversion von OpManager kann man 20 der wichtigsten Geräte wie Router, Server, Switches, Mail-Server, Firewalls, Drucker, etc. auf Verfügbarkeit, Auslastung und „Gesundheit“ überwachen lassen und bei Bedarf sofort per E-Mail oder SMS die Kontaktpersonen über Probleme alarmieren [21].



(C) Copyright 2003-2005, AdventNet Inc.

Abbildung 6: Screenshot\_1 aus dem Netzwerkmanagementprodukt OpManager [21].



Abbildung 7: Screenshot\_2 aus dem Netzwerkmanagementprodukt OpManager [21].

## **4 Qualitätsmanagement bei der GSM-basierten Zählerauslesung**

Leider erstrecken sich die bereit erwähnten Lösungen und Produkte nur auf geschlossene LAN-Netzwerke oder höchstens WLAN-Netzwerke. Dehnt sich das Netzwerk außerhalb dieser geschlossenen LAN- oder WLAN-Verbindungen aus, führt kein Weg daran vorbei, nach individuellen Lösungen zu suchen, die mindestens die Komponenten des Netzwerkes überwachen, die von den übrigen Managementtools nicht verwaltet werden können.

Das Netzwerk des Insel-Projekts ist so ein Netzwerk, das aus einem LAN- und einem GSM-Netzwerk besteht. Das LAN-Netzwerk besteht aus Rechnern, Routern, Modems, Servern etc. der Firma ENVIDATEC und einem Datenserver zum Datenmanagement der Daten des Insel-Projekts. Das GSM-Netzwerk besteht aus den Kommunikationsmodulen (VIDA84) für die Erfassung der Lastgänge der Liegenschaften des Insel-Projekts. Diese beiden Teilnetzwerke kommunizieren untereinander über GSM durch das Modem seitens des LAN-Teilnetzwerkes und die GSM-Telefonkarten, die sich in jedem Datenlogger (VIDA84) befinden auf Seiten des GSM-Teilnetzwerkes.

Abb. 8 zeigt die Struktur des Insel-Projekt-Netzwerkes.

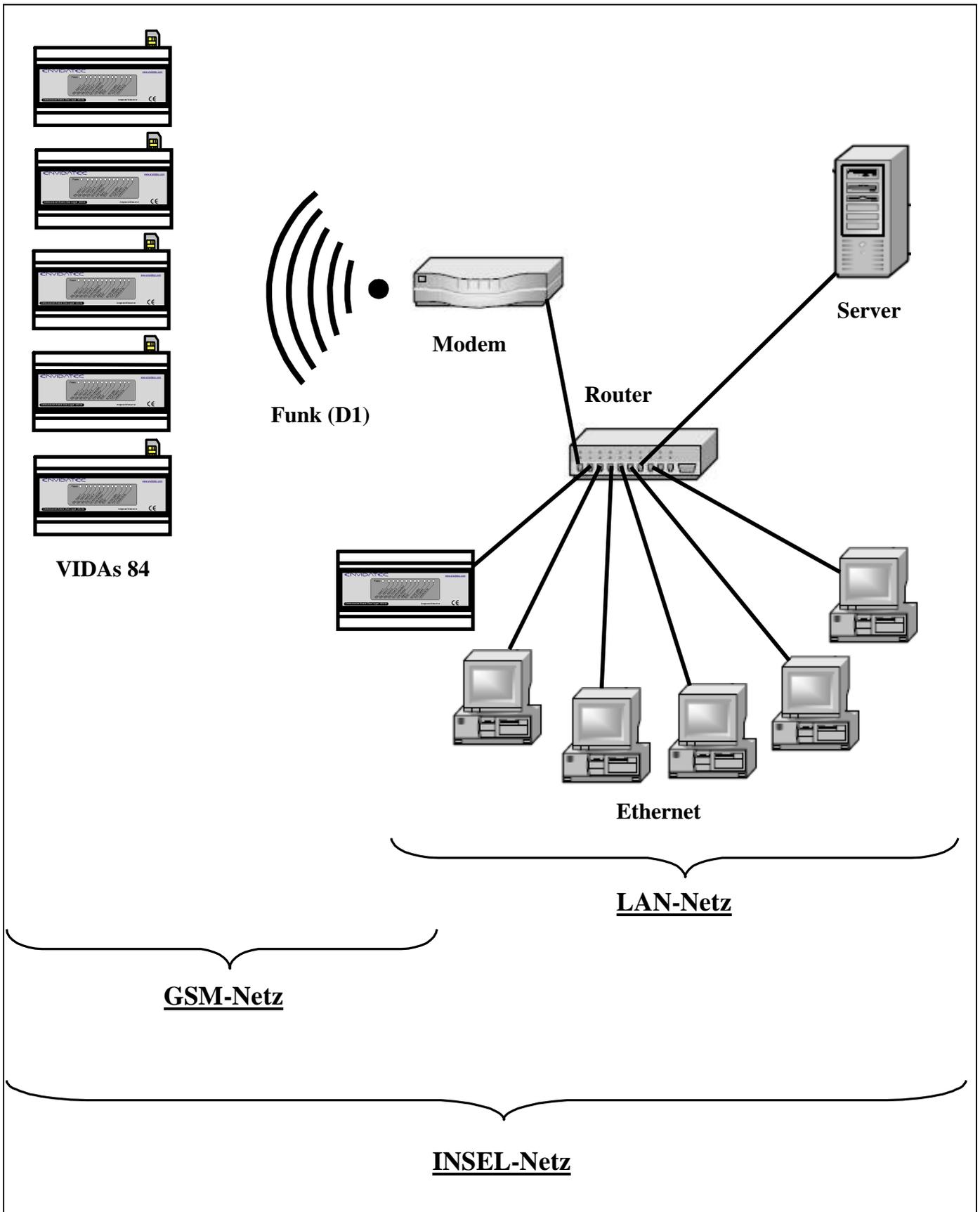


Abbildung 8: Struktur des Insel-Projekt-Netzwerkes.

## **4.1 Dokumentation der Liegenschaften**

Durch die Dokumentation der Liegenschaften werden die wichtigsten Aspekte des ersten Funktionsbereichs im Netzwerkmanagement, des „Konfigurationsmanagements“, behandelt. In diesem werden alle Netzkomponenten, die in Verbindung mit dem GSM-Netz stehen, deren Adressierungen, deren technische Daten und ihre Beziehungen zu einander protokolliert.

Die übrigen Aspekte des Konfigurationsmanagements gehören entweder zum LAN-Netzbereich, wie z. B. die Routing-Kontrolle, und werden somit durch die bereits erwähnten Tools abgedeckt, oder werden durch die zu entwickelnden Tools behandelt, wie die Ermittlung von Aktivität oder Inaktivität der Netzkomponenten.

In den folgenden beiden Unterkapiteln wird auf die Notwendigkeit einer guten Dokumentation hingewiesen. Des Weiteren wird die Struktur der Dokumentation demonstriert mit den für das Konfigurationsmanagement wichtigen Daten sowie weiteren Daten für die Fehlerbehebung.

### **4.1.1 Begründung der Notwendigkeit einer guten Dokumentation**

Prozesse und Projekte zu dokumentieren ist in der Regel keine bei Arbeitnehmern beliebte Arbeit und wird daher nicht selten vernachlässigt, obwohl eine gute Dokumentation eines der Qualitätsmerkmale für den Erfolg eines Projekts oder eine Aufgabe überhaupt stellt. Außerdem spielt eine gute Dokumentation eine große Rolle bei der Projektkontrolle und der Projektplanung. Sie hilft erheblich bei der Fehlererkennung und bei der Problemlösung. Es lohnt sich immer zu dokumentieren.

Die Dokumentation der Liegenschaften im Insel-Projekt soll der Qualitätssicherung dienen, in der alle wichtigen Informationen, die mit den Liegenschaften zu tun haben dokumentiert werden. Dies sind insbesondere zum einen Angaben zu Adresse der Liegenschaft, Ansprechpartner, Einbauort der Datenlogger, Art der Stromzähler und deren Spezifikationen, etc. als auch alle Informationen und Spezifikationen der Datenlogger wie die Seriennummer des Geräts, die Hardware- und Softwarenummer Verbindungsart des Geräts, Daten der benutzten GSM-Karte, Verbindungsart, etc.. Solche Ansammlung von Informationen stellt bei übersichtlicher Strukturierung und Transparenz ein sehr hilfreiches Mittel dar, um auftretende Probleme im Kommunikationsnetzwerk des Insel-Projekts so schnell wie möglich beheben zu können.

### **4.1.2 Struktur und Inhalt der Dokumentation**

Die Dokumentation der Liegenschaften wurde in Excel verfasst und stellt für jede Liegenschaft eine Arbeitstabelle dar, die in 7 Abschnitte geteilt ist:

- Bilderabschnitt: der Bilderabschnitt präsentiert 2 Bilder, die einmal die Außensicht der Liegenschaft und einmal den Datenlogger mit dem Zähler zusammen zeigen.

- Liegenschaftsabschnitt: in dem Liegenschaftsabschnitt befinden sich Informationen über:
  - Die ID-Nummer der Liegenschaft,
  - die Adresse der Liegenschaft,
  - die zuständige Behörde für die Liegenschaft und
  - die Vertragsnummer der Liegenschaft beim Projekt.
  
- Stromzählerabschnitt: Im Stromzählerabschnitt befinden sich Informationen über:
  - Die Art der Stromzähler (Messwandler 1/6),
  - den Typ der Stromzähler (digitaler oder analoger Stromzähler),
  - die Nummer des Zählers,
  - den Wandlerfaktor des Zählers,
  - die S0-Impulse des Zählers und
  - die Strom- und Spannungswerte des Zählers.
  
- Datenloggerabschnitt: Im Datenloggerabschnitt befinden sich Informationen über:
  - Den Loggernamen,
  - Die Nummer des Geräts (Gerätenummer von Envidatec),
  - die Seriennummer des Geräts (Herstellernummer),
  - die HAW Inventarnummer des Geräts,
  - die Hardwarenummer des Geräts (MAC-ID),
  - die Firm- und Webware des Geräts,
  - die Verbindungsart des Geräts
  - das Einbaudatum und den Einbauort des Geräts und
  - die Gerätepassnummer des Geräts.
  
- SIM-Kartenabschnitt: Im SIM-Kartenabschnitt befinden sich Informationen über:
  - Die Telefonnummer der eingesetzte GSM-Karte,
  - die Kartenummer der eingesetzte GSM-Karte,
  - die PIN der eingesetzte GSM-Karte und
  - die PUK der eingesetzte GSM-Karte.
  
- Ansprechpartnerabschnitt: Im Ansprechpartnerabschnitt befinden sich Informationen über:
  - Den Name der oder des Ansprechpartner(-s)
  - die Funktion des Ansprechpartner(-s),
  - die Firma des Ansprechpartner(-s),
  - die Adresse des Ansprechpartner(-s),
  - die Telefonnummern des Ansprechpartner(-s) und
  - die E-Mail Adresse des Ansprechpartner(-s).
  
- Bemerkungsabschnitt: Der Bemerkungsabschnitt ist für Bemerkungen bzw. Ausnahmen, die bei der Liegenschaft auftreten können.

Die folgenden drei Tabellen stellen als Beispiel die Dokumentation der ersten Liegenschaft dar, die mit einem Kommunikationsmodul VIDA 84 ausgerüstet wurde.

## Liegenschafts-Protokoll

*Bundesamt für Seeschifffahrt und Hydrographie*



Stromzähler & Datenlogger



Bundesamt für Seeschifffahrt und Hydrographie

Tabelle 1: Der erste Teil der BSH Arbeitstabelle mit Bilder.

<b>Liegenschaft</b>			
ID-Nummer			
Adresse_PLZ	Bernhard-Nocht-Str. 78		20359
zust. Behörde	Bildungsbehörde		
Vertragsnr.	60069074		
<b>Stromzähler</b>			
Art	Meßwandler 1/6		
Typ	LZMJ-IB 2232084100		
Zählernr.	100 16512		
Wandlerfaktor	1000	Strom	50/5 A
SO-Impulse/kWh	12000	Spannung	10/0.1 kV
<b>Datenlogger</b>			
Loggername	VIDA 84	Gerätenr.	55198
		Serienr.	1023900669008
		HAW-Inv.Nr.	
		MAC-ID	00:07:8E:0A:35:50
		Firmware	E1.125-S
		Webware	E1.115-S
Verbindungsart	GSM		
Einbaudatum	06.11.2007		
Einbauort	Raum 08 – Keller		
Gerätepaßnr.			
<b>SIM-Karte</b>			
Telefonnr.	0151/12622560		
Kartennr.	50546494		
PIN	4520		
PUK			
<b>Ansprechpartner</b>			
A.partner1	Wilfried Weber	Firmenname	BSH
Funktion	Verwaltungsleiter	Telefon	040/ 3190-2320
Mobil		Fax	
Adresse	Bernhard-Nocht-Str. 78		
E-Mail			
A.partner2	Helmut Briede	Firmenname	
Funktion	Bundesbauabteilung	Telefon	040/ 42842-342
Mobil		Fax	
Adresse			
E-Mail	<a href="mailto:helmut.briede@bba.hamburg.de">helmut.briede@bba.hamburg.de</a>		
A.partner3		Firmenname	
Funktion		Telefon	
Mobil		Fax	
Adresse			
E-Mail			
A.partner4		Firmenname	
Funktion		Telefon	
Mobil		Fax	
Adresse			
E-Mail			

Tabelle 2: Der zweite Teil der BSH Arbeitstabelle mit den wichtigen Daten.

A.partner5	-----	Firmenname	-----
Funktion	-----	Telefon	-----
Mobil	-----	Fax	-----
Adresse	-----		
E-Mail	-----		

**Bemerkungen**

Tabelle 3: Der dritte Teil der BSH Arbeitstabelle.

Außerdem beinhaltet die Dokumentation eine Übersichtstabelle, in der alle Daten aus allen Liegenschaften nach Abschnitten gruppiert sind. Durch einen Klick auf das „+“-Zeichen kann man den gesamten Inhalt der vorliegenden Gruppen darstellen. Die Inhalte der einzelnen Liegenschaften sind durch Verknüpfungen mit dazugehörigen Arbeitsmappen immer aktuell.

Aus der realisierten Gesamtübersicht lässt sich bei Bedarf ohne größere Mühe eine Datenbank erzeugen.

Tabelle 4 zeigt die Übersichtstabelle in ihrer gruppierten Form.

4 Qualitätsmanagement beider GSM-basierten Zählerauslesung

+
+
+
+
+
+

<i>Liegenschaft</i>	<i>Stomzähler</i>	<i>Datenlogger</i>	<i>Telefonnr.</i>	<i>Ansprechpartner1</i>	<i>Ansprechpartner2</i>	<i>Bemerkungen</i>
<a href="#">Bergedorf</a>	0	VIDA 84	0	Stefan Wittkowski	0	0
<a href="#">Berliner Tor</a>	0	0	0	0	0	0
<a href="#">Botanik</a>	Meßwandler 1/6	VIDA 84	0151/12622685	Hr. Nehls	Fr. Petra Litke	0
<a href="#">BSH</a>	Meßwandler 1/6	VIDA 84	0151/12622560	Wilfried Weber	Helmut Briede	0
<a href="#">BSU</a>	Meßwandler 1/6	VIDA 84	0151/126 22561	Hr. Anschütz	Hr. Lenuweit	0
<a href="#">Eisbahn</a>	Meßwandler 1/6	VIDA 84	0151/12622558	Hr. Hüttenrau	Hr. Medag	0
<a href="#">DESY</a>	Meßwandler 1/6	VIDA 84	0151/12622564	Hr. Schwitzky	Fr. Petra Litke	0
<a href="#">LVA</a>	Meßwandler 1/6	VIDA 84	0151/1262255	Hr. Pape	0	Absolut kein Empfang
<a href="#">Musikhalle</a>	Meßwandler 1/6	VIDA 84	0151/12622562	Hr. Langenhahn	0	0
<a href="#">Physik</a>	Meßwandler 1/6	VIDA 84	0151/12622563	Hr. Heine	Fr. Petra Litke	Zählerschrank war nicht verplombt, daher Kabel für Impulse mit
<a href="#">Schaartorschleuse</a>	Meßwandler 1/6	VIDA 84	0151/12622559	Hr. von der Kammer	Hr. Marold	2 Zähler vorhanden
<a href="#">Sch. Mümmelmannsberg</a>	Meßwandler 1/6	VIDA 84	0151/162622690	Hr. Runowski	Hr. Hein	Es gibt 2 Zähler und ein Summierer mit den Daten:
<a href="#">Sch. Steilshoop</a>	Meßwandler 1/6	VIDA 84	0151/12622557	Hr. Walter	Hr. Eggers	0
<a href="#">Staatsoper</a>	Meßwandler 1/6	VIDA 84	0151/12622556	Hr. Magsig	Hr. Mehr	0

**Tabelle 1: Übersichtstabelle in der gruppierten Form.**

## 4.2 Qualitätssicherung

### 4.2.1 Bestimmung der Einsatzgebiete

Für die Umsetzung der Qualitätssicherung in der Praxis gilt es anhand des Funktionsbereichs „Fehlermanagement“ des Netzwerkmanagements das Netz bzw. die Netzkomponenten und die angeschlossenen Systeme zu beobachten, um Fehler und deren Ursachen zu erkennen.

Um die Verfügbarkeit des Netzes zu sichern und das Überwachungsgebiet abzugrenzen, ist es wichtig, die drei nachfolgenden Fragen zu stellen. Deren Beantwortung veranschaulicht den Prozess des Fehlermanagements.

- **Was?** Was muss im Insel-Projekt-Netzwerk überwacht werden, um die Sicherung der Durchführbarkeit zu gewährleisten.
- **Warum?** Warum muss man die erwähnten Objekte überwachen.
- **WO?** Wo muss man die betreffenden Elemente überwachen. Da das Netzwerk aus zwei Teilnetzwerken besteht, ist es wichtig zu wissen, ob im LAN- oder im GSM-Teil oder sogar in beiden Teilen überwacht werden muss.

Eine Übersichtstabelle gibt die Antworten auf die drei wichtigen Fragen auf einem Blick.

<b>Was?</b>	<b>WO?</b>	<b>Warum?</b>
Modem	LAN	Bei Ausfällen des Modems gibt es keine Verbindung zwischen den beiden Teilnetzwerken.
Server	LAN	Ist der Server nicht erreichbar, gibt es überhaupt keine Daten mehr.
Plattenspeicher	LAN (Server)	Ist die Platte voll gibt es keine Daten mehr.
CPU & RAM	LAN(Server)	Bei großer Auslastung, gibt es negative Auswirkung auf vorstehende Prozesse.
GSM-Netz (D1)	GSM	Bei Ausfall oder starker Auslastung des D1-Netz, gibt es keine Daten mehr / große Verbindungszeiten.
VIDA 84 (Datenlogger)	GSM	Bei Ausfall der VIDA gibt es keine neuen Daten mehr.
Verbindungsdauer	GSM	Gibt Einblick in die Leistung des Netzwerks und die Höhe der Kosten.
Gesamtdauer	GSM + LAN (Server)	Gibt Auskunft über die Leistung.
Fehler (Verluste)	GSM + LAN (Server, Modem)	Gibt Informationen über die Art und die Häufigkeit der Probleme der beiden Teilnetze.

**Tabelle 5: Die Antworten auf die Was? Wo? Warum? Fragen.**

Die Verfügbarkeit des GSM-Netzes und des Datenloggers VIDA84 kann nur indirekt durch Betrachtung der entstehenden Fehler bei der Kommunikation festgestellt werden.

Nachdem feststeht, was, wo und warum überwacht werden soll, kann durch Analyse der Ereignis- und Fehler-Reports nach einer Lösung zur Überwachung der Komponenten und Prozesse (VIDAs, GSM-Netz, Verbindungsdauer, Gesamtdauer, Fehler) gesucht werden, die nicht über die vorgeschlagene Netzwerkmanagement-Software überwacht werden können.

## 4.2.2 Analyse der Ereignisreports

Als Ausgangsposition besteht für jede Kommunikation zwischen Modem und Kommunikationsmodul (VIDA84) ein Logfile, das von der Partner-Firma Envidatec zur Verfügung gestellt wurde, in dem alle Verbindungs- und Datenspeicherungsversuche protokolliert sind.

Da diese Protokolle nur in Textform vorliegen und sich für eine automatische Auswertung nicht eignen, müssen zunächst die wichtigsten Informationen herausgefiltert werden um sie anschließend analysieren und in tabellarischen und grafisch anschaulichen Form darzustellen zu können.

Im Folgenden werden Ausschnitte aus den Ausgangsdaten dargestellt, um die Struktur des Logfiles zu demonstrieren. Dabei werden die für die Überwachung des GSM-Teilnetzwerkes wichtigen Informationen hervorgehoben.

## 4.2.3 Feststellung der Verbindungs- und Gesamtzeit

Der folgende Abschnitt repräsentiert eine erfolgreiche Verbindung zwischen dem Modem im LAN-Teilnetzwerk und einen Datenlogger Gerät im GSM-Teilnetzwerk.

```

01 2007-09-19 10:00:01 Starting ADF session
02 2007-09-19 10:00:02 Last update is 2007-09-19
03 2007-09-19 10:00:02 Opening PPP link to 015112622558
04 2007-09-19 10:00:02 ----- PPP log follows -----
05 Serial connection established.
06 Using interface ppp0
07 Connect: ppp0 <--> /dev/ttyS1
08 local IP address 192.168.10.5
09 remote IP address 192.168.10.6
10 2007-09-19 10:00:33 ----- PPP log finished -----
11 2007-09-19 10:00:33 Retrieving gateway data
12 2007-09-19 10:02:36
    URL:http://192.168.10.6/logdata.csv?DAYv=19&MONv=09&YEARv=2007&DAYb=01&MONb=
    01&YEARb=132008 [77947/77947] -> "files/20070919_100002.csv"
14 2007-09-19 10:02:36 Dropping PPP link
15 2007-09-19 10:02:51 Successfully retrieved files/20070919_100002.csv
16 2007-09-19 10:02:51 New lastupdate is 2007-09-19
17 2007-09-19 10:02:51 Running /home/adf84/bin/vida84-import-adf.sh vida84_fast/INSEL_12
    ora3
18 2007-09-19 10:02:51 Starting VIDA84 ADF file import
19 2007-09-19 10:02:51 Converting files/20070919_100002.csv ... OK, 603 sample(s) in each
    data row
20 2007-09-19 10:02:51 Skipping column 1 (target data row is 0)
21 2007-09-19 10:02:51 Skipping column 10 (target data row is 0)
22 2007-09-19 10:02:52 Skipping column 11 (target data row is 0)
23 2007-09-19 10:02:52 Skipping column 12 (target data row is 0)
24 2007-09-19 10:02:52 Skipping column 13 (target data row is 0)
25 2007-09-19 10:02:52 Skipping column 14 (target data row is 0)
26 2007-09-19 10:02:52 Skipping column 15 (target data row is 0)
27 2007-09-19 10:02:52 Skipping column 16 (target data row is 0)
28 2007-09-19 10:02:52 Skipping column 2 (target data row is 0)
29 2007-09-19 10:02:52 Skipping column 3 (target data row is 0)
30 2007-09-19 10:02:52 Skipping column 4 (target data row is 0)

```

```

31 2007-09-19 10:02:52 Skipping column 5 (target data row is 0)
32 2007-09-19 10:02:52 Skipping column 6 (target data row is 0)
33 2007-09-19 10:02:52 Skipping column 7 (target data row is 0)
34 2007-09-19 10:02:52 Skipping column 8 (target data row is 0)
35 2007-09-19 10:02:52 Importing tmpcsv/20070919100251_9.csv (column 9 -> DR 360252285)
   at ORA3
36 2007-09-19 10:02:52 JEDFI version 1.20 start
37 2007-09-19 10:02:54 File: tmpcsv/20070919100251_9.csv
38 2007-09-19 10:02:54 Reading data from file...
39 2007-09-19 10:02:54 Using format JEDFE
40 2007-09-19 10:02:55 Storing data to Database, please wait...
41 2007-09-19 10:02:55 Connecting to Oracle...
42 2007-09-19 10:02:58 Inserting rows...
43 2007-09-19 10:02:58 INSERT failed: DATE:18-09-2007 22:00:00 VALUE:0.0
44 2007-09-19 10:02:58 INSERT failed: DATE:18-09-2007 22:01:00 VALUE:0.0
45 2007-09-19 10:02:58 INSERT failed: DATE:18-09-2007 22:02:00 VALUE:0.0
46 2007-09-19 10:02:58 INSERT.....
   .
47 2007-09-19 10:03:03 INSERT failed: DATE:19-09-2007 06:01:00 VALUE:0.0
48 2007-09-19 10:03:03 INSERT failed: DATE:19-09-2007 06:02:00 VALUE:0.0
49 2007-09-19 10:03:04 603 records total
50 2007-09-19 10:03:04 120 records imported
51 2007-09-19 10:03:04 483 records failed (duplicate entries)
52 2007-09-19 10:03:04 Operation time = 8559 msec
53 2007-09-19 10:03:04 Completed
54 2007-09-19 10:03:04 tmpcsv/20070919100251_9.csv (column 9) imported successfully to all
   server(s)
55 2007-09-19 10:03:04 Conversion and import of files/20070919_100002.csv was successfull,
   deleting
56 2007-09-19 10:03:04 Conversion finished, 1 source file(s) found, 1 fully processed, 0 left due
   to errors
57 2007-09-19 12:00:02 Starting ADF session

```

Der Start einer Verbindung wird mit Datum, Uhrzeit und „Starting ADF session“, protokolliert wie es Zeile 01 aus dem ersten Logfileabschnitt zeigt. Ab der Zeile 15 wird die Verbindung nach erfolgreichem Herunterladen der Lastdaten „Successfully retrieved files ...“ beendet und mit dem Importieren und Speichern der Lastdaten im Datenbankserver bis Zeile 54 begonnen, wo das Ende der Sitzung mit „Completed“ protokolliert wird. In Zeile 20 bis 34 werden beim Importieren die leeren Zeilen aus den empfangenen Daten aussortiert, in Zeile 43 bis 48 werden aus den importierten Daten vor der Datenspeicherung die fehlerhaften Daten wie „0,0“, „-99999,99“ oder Duplikate aussortiert.

In den Zeilen 49 bis 51 ist die Anzahl der insgesamt gesammelten Daten „records total“, der importierten „records imported“ und der fehlerhaften Daten „records failed“ nacheinander dokumentiert. Aus diesen drei Zeilen lässt sich auch bei einer erfolgreichen Verbindung feststellen, ob die gesammelten Daten sinnvoll sind. Ist zum Beispiel die Anzahl der fehlerhaften Daten größer als die Anzahl der insgesamt gesammelten Daten, so kann man davon ausgehen, dass der Datenlogger die meiste Zeit unvernünftige Daten wie „0,0“ oder „-99999.99“ aufzeichnet.

## 4.2.4 Fehleranalyse

Bei der Analyse der Verbindungs-Logfiles sind 3 Fehlerarten festzustellen, die von verschiedenen Diensten verursacht wurden. Jeder dieser Dienste ist für einen der drei wichtigen Schritte des Kommunikationsaufbaus verantwortlich. Für den ersten Schritt des Verbindungsaufbaus ist das PPP-Protokoll zuständig, aus dem der gleichnamige Fehler „PPP-Fehler“ stammt. Im zweiten Schritt kümmert sich der WGET-Dienst, aus dem der gleichnamige Fehler resultiert, um das Herunterladen der Lastdaten. Im letzten Schritt übernimmt die Datenbank Oracle, die Speicherung der Lastdaten in den Datenbankserver, aus dem auch das Database Fehler stammt.

### 4.2.4.1 PPP-Fehler:

der PPP-Fehler wird vom Netzwerkprotokoll PPP (Point-to-Point-Protokoll) bei nicht Erreichbarkeit oder bei Modemproblemen hervorgerufen. PPP ist ein Protokoll zur Übertragung von Datenpaketen über Modem bzw. ISDN. Es unterstützt das TCP/IP-Protokoll über die normale Telefonleitung, in dem auf beiden Seiten die Datenpakete so eingepackt werden, dass sie über Telefonleitung transportiert und am anderen Ende wieder ausgepackt werden können [22, 23].

Das PPP-Protokoll umfasst, wie aus dem Quellcode des Protokolls zu entnehmen ist, [24] 19 mögliche Fehlern bzw. Gründe um eine Verbindung abzuberechnen:

- **Error 1:** An immediately fatal error of some kind occurred, such as an essential system call failing, or running out of virtual memory.
- **Error 2:** An error was detected in processing the options given, such as two mutually exclusive options being used.
- **Error 3:** Pppd is not setuid-root and the invoking user is not root.
- **Error 4:** The kernel does not support PPP, for example, the PPP kernel driver is not included or cannot be loaded.
- **Error 5:** Pppd terminated because it was sent a SIGINT, SIGTERM or SIGHUP signal.
- **Error 6:** The serial port could not be locked.
- **Error 7:** The serial port could not be opened.
- **Error 8:** The connect script failed (returned a non-zero exit status).
- **Error 9:** The command specified as the argument to the \fR option could not be run.
- **Error 10:** The PPP negotiation failed, that is, it didn't reach the point where at least one network protocol (e.g. IP) was running.

- **Error 11:** The peer system failed (or refused) to authenticate itself.
- **Error 12:** The link was established successfully and terminated because it was idle.
- **Error 13:** The link was established successfully and terminated because the connect time limit was reached.
- **Error 14:** Callback was negotiated and an incoming call should arrive shortly.
- **Error 15:** The link was terminated because the peer is not responding to echo requests.
- **Error 16:** The link was terminated by the modem hanging up.
- **Error 17:** The PPP negotiation failed because serial loopback was detected.
- **Error 18:** The init script failed (returned a non-zero exit status).
- **Error 19:** We failed to authenticate ourselves to the peer.

Die Zeilen 07 und 25 im folgenden Abschnitt des Logfiles zeigen als Beispiel das Auftreten eines PPP-Fehlers mit der Begründung, dass die serielle Schnittstelle nicht gesperrt werden konnte. Außerdem ist in diesem Abschnitt zu sehen, dass bei dieser Fehlerart (PPP) keine Verbindungszeiten bzw. Verbindungskosten entstanden sind sondern nur Zeitverzögerungen.

```
01 2007-08-31 16:52:52 Starting ADF session
02 2007-08-31 16:52:52 Last update is 2007-08-13
03 2007-08-31 16:52:52 Opening PPP link to 015112622558
04 2007-08-31 16:52:52 ----- PPP log follows -----
05 Device ttyS1 is locked by pid 32510
06 2007-08-31 16:52:52 ----- PPP log finished -----
07 2007-08-31 16:52:52 FATAL: could not connect gateway, PPP error 6
08 2007-08-31 17:45:01 Starting ADF session
09 2007-08-31 17:45:01 Last update is 2007-08-13
10 2007-08-31 17:45:01 Opening PPP link to 015112622558
11 2007-08-31 17:45:01 ----- PPP log follows -----
12 Serial connection established.
13 Using interface ppp1
14 Connect: ppp1 <--> /dev/ttyS1
15 local IP address 192.168.10.5
16 remote IP address 192.168.10.6
17 2007-08-31 17:45:34 ----- PPP log finished -----
18 2007-08-31 17:45:34 Retrieving gateway data
19 2007-08-31 18:00:01 Starting ADF session
20 2007-08-31 18:00:02 Last update is 2007-08-13
21 2007-08-31 18:00:02 Opening PPP link to 015112622558
22 2007-08-31 18:00:02 ----- PPP log follows -----
23 Device ttyS1 is locked by pid 2547
24 2007-08-31 18:00:02 ----- PPP log finished -----
25 2007-08-31 18:00:02 FATAL: could not connect gateway, PPP error 6
26 2007-08-31 18:15:37 Dropping PPP link
```

```
27 2007-08-31 18:15:37 ERROR: data could not be downloaded, WGET error 1
28 2007-08-31 19:05:01 Starting ADF session
29 2007-08-31 19:05:01 Last update is 2007-08-13
.
.
.
```

#### 4.2.4.2 WGET-Fehler:

Der WGET-Fehler wird vom Netzwerk Werkzeug WGET verursacht, der für das Herunterladen von Lastmanagement-Dateien zuständig ist. WGET ist ein sehr verbreitetes freies (Open-Source) Programm zum Herunterladen von Dateien, Webseiten etc. übers Netzwerk bzw. Internet (FTP, http oder HTTPs). Außerdem ermöglicht WGET langsamen und instabilen Verbindungen, die abgebrochenen Downloads wieder aufzunehmen [25, 26].

Das WGET unterscheidet insgesamt 7 Fehlerarten, die man aus dem Quellcode entnehmen kann [27]:

- **Error 1:** Unsupported scheme.
- **Error 2:** Invalid host name.
- **Error 3:** Bad port number.
- **Error 4:** Invalid user name.
- **Error 5:** Unterminated IPv6 numeric address.
- **Error 6:** IPv6 addresses not supported.
- **Error 7:** Invalid IPv6 numeric address.

Für dieses Projekt ist nur Error 1 (Unsupported scheme) von Bedeutung, die anderen Fehlerarten sind hier deshalb unwichtig, weil sie entweder vom PPP-Dienst vorher behandelt wurden oder sich an eine hier nicht benutzte IP-Version adressieren.

Error 1 bedeutet, dass die an WGET übermittelte URL nicht unterstützt wird. Deshalb wird meistens die Verbindung nach einer Weile abgebrochen, wie es die Zeilen 14 und 27 aus dem folgenden Logfileabschnitt zeigen. Hier ist auch zu beachten, dass trotz des nicht erfolgreichen Herunterladens von Daten schon Verbindungszeit verbucht wird und somit Kosten entstanden sind.

```
01 2007-08-30 18:00:01 Starting ADF session
02 2007-08-30 18:00:01 Last update is 2007-08-13
03 2007-08-30 18:00:01 Opening PPP link to 015112622558
04 2007-08-30 18:00:01 ----- PPP log follows -----
05 Serial connection established.
07 Using interface ppp0
```

```

08 Connect: ppp0 <--> /dev/ttyS1
09 local IP address 192.168.10.5
10 remote IP address 192.168.10.6
11 2007-08-30 18:00:33 ----- PPP log finished -----
12 2007-08-30 18:00:33 Retrieving gateway data
13 2007-08-30 18:30:36 Dropping PPP link
14 2007-08-30 18:30:36 ERROR: data could not be downloaded, WGET error 1
15 2007-08-30 20:00:01 Starting ADF session
16 2007-08-30 20:00:02 Last update is 2007-08-13
17 2007-08-30 20:00:02 Opening PPP link to 015112622558
18 2007-08-30 20:00:02 ----- PPP log follows -----
19 Serial connection established.
20 Using interface ppp0
21 Connect: ppp0 <--> /dev/ttyS1
22 local IP address 192.168.10.5
23 remote IP address 192.168.10.6
24 2007-08-30 20:00:32 ----- PPP log finished -----
25 2007-08-30 20:00:32 Retrieving gateway data
26 2007-08-30 20:30:34 Dropping PPP link
27 2007-08-30 20:30:34 ERROR: data could not be downloaded, WGET error 1
28 2007-08-30 22:00:01 Starting ADF session
29 2007-08-30 22:00:01 Last update is 2007-08-13
30 2007-08-30 22:00:01 Opening PPP link to 015112622558
.
.

```

#### 4.2.4.3 Database-Fehler:

Database-Fehler wird vom Oracle Datenbank Server verursacht. Hierbei werden die heruntergeladenen Daten trotz erfolgreichen Verbindungsaufbaus und Datentransfers nicht importiert bzw. in den Datenbank-Server gespeichert, dies ist zum Beispiel auf Probleme in der Datenbank oder nicht ausreichenden Speicherplatz zurück zu führen. Auch in diesem Fall sind durch die Verbindungsdauer Kosten entstanden.

Die Zeilen 42 bis 87 aus dem folgenden Logfileabschnitt demonstrieren die Fehlermeldungen der Oracle Datenbank. Außerdem weisen die Zeilen 92 bis 94 daraufhin hin, dass ein persönliches Eingreifen erforderlich ist, um die fehlerhaften Dateien zu löschen.

```

01 2007-08-29 10:00:01 Starting ADF session
02 2007-08-29 10:00:01 Last update is 2007-08-29
03 2007-08-29 10:00:01 Opening PPP link to 015112622554
04 2007-08-29 10:00:01 ----- PPP log follows -----
05 Serial connection established.
06 Using interface ppp3
07 Connect: ppp3 <--> /dev/ttyS1
08 local IP address 192.168.10.5
09 remote IP address 192.168.10.6
10 2007-08-29 10:00:32 ----- PPP log finished -----
11 2007-08-29 10:00:32 Retrieving gateway data
12 2007-08-29 10:00:41
URL:http://192.168.10.6/logdata.csv?DAYv=29&MONv=08&YEARv=2007&DAYb=01&MONb=
01&YEARb=2008 [5320/5320] -> "files/20070829_100001.csv" [1]

```

```

13 2007-08-29 10:00:41 Dropping PPP link
14 2007-08-29 10:00:41 Successfully retrieved files/20070829_100001.csv
15 2007-08-29 10:00:42 New lastupdate is 2007-08-29
16 2007-08-29 10:00:42 Running /home/adf84/bin/vida84-import-adf.sh vida84_fast/INSEL_13
    ora3
17 2007-08-29 10:00:42 Starting VIDA84 ADF file import
18 2007-08-29 10:00:42 Converting files/20070829_100001.csv ... OK, 40 sample(s) in each data
    row
19 2007-08-29 10:00:42 Skipping column 1 (target data row is 0)
20 2007-08-29 10:00:42 Skipping column 10 (target data row is 0)
21 2007-08-29 10:00:42 Skipping column 11 (target data row is 0)
22 2007-08-29 10:00:42 Skipping column 12 (target data row is 0)
23 2007-08-29 10:00:42 Skipping column 13 (target data row is 0)
24 2007-08-29 10:00:42 Skipping column 14 (target data row is 0)
25 2007-08-29 10:00:42 Skipping column 15 (target data row is 0)
26 2007-08-29 10:00:42 Skipping column 16 (target data row is 0)
27 2007-08-29 10:00:42 Skipping column 2 (target data row is 0)
28 2007-08-29 10:00:42 Skipping column 3 (target data row is 0)
29 2007-08-29 10:00:42 Skipping column 4 (target data row is 0)
30 2007-08-29 10:00:43 Skipping column 5 (target data row is 0)
31 2007-08-29 10:00:43 Skipping column 6 (target data row is 0)
32 2007-08-29 10:00:43 Skipping column 7 (target data row is 0)
33 2007-08-29 10:00:43 Skipping column 8 (target data row is 0)
34 2007-08-29 10:00:43 Importing tmpcsv/20070829100042_9.csv (column 9 -> DR 360252295)
    at ORA3
35 2007-08-29 10:00:43 JEDFI version 1.20 start
36 2007-08-29 10:00:46 File: tmpcsv/20070829100042_9.csv
37 2007-08-29 10:00:46 Reading data from file...
38 2007-08-29 10:00:46 Using format JEDFE
39 2007-08-29 10:00:47 Storing data to Database, please wait...
40 2007-08-29 10:00:47 Connecting to Oracle...
41 2007-08-29 10:00:52 Inserting rows...
42 java.sql.SQLException: ORA-01654: unable to extend index
    JEVIS.LOGGER_DATA_PKZ58386526_1_1_1 by 8192 in tablespace JEVIS

43         at oracle.jdbc.dbaccess.DBError.throwSQLException(DBError.java:
44         at oracle.jdbc.ttc7.TTloer.processError(TTloer.java:289)
45         at oracle.jdbc.ttc7.Oall7.receive(Oall7.java:573)
46         at oracle.jdbc.ttc7.TTC7Protocol.doOall7(TTC7Protocol.java:1891
47         at oracle.jdbc.ttc7.TTC7Protocol.parseExecuteFetch(TTC7Protocol
48         at oracle.jdbc.driver.OracleStatement.executeNonQuery(OracleSta
49         at oracle.jdbc.driver.OracleStatement.doExecuteOther(OracleStat
50         at oracle.jdbc.driver.OracleStatement.doExecuteWithTimeout(Orac
51         at oracle.jdbc.driver.OraclePreparedStatement.executeUpdate(Ora
52         at envdatec.jevis.tool.jecsvimport.ImportWorker.beginImport(Im
53         at envdatec.jevis.tool.jecsvimport.JECSVImport.<init>(JECSVImp
54         at envdatec.jevis.tool.jecsvimport.JECSVImport.main(JECSVImpor
56 java.sql.SQLException: ORA-01654: unable to extend index
    JEVIS.LOGGER_DATA_PKZ58386526_1_1_1 by 8192 in tablespace JEVIS

57         at oracle.jdbc.dbaccess.DBError.throwSQLException(DBError.java:
58         at oracle.jdbc.ttc7.TTloer.processError(TTloer.java:289)
59         at oracle.jdbc.ttc7.Oall7.receive(Oall7.java:573)
60         at oracle.jdbc.ttc7.TTC7Protocol.doOall7(TTC7Protocol.java:1891
70         at oracle.jdbc.ttc7.TTC7Protocol.parseExecuteFetch(TTC7Protocol
80         at oracle.jdbc.driver.OracleStatement.executeNonQuery(OracleSta
81         at oracle.jdbc.driver.OracleStatement.doExecuteOther(OracleStat
82         at oracle.jdbc.driver.OracleStatement.doExecuteWithTimeout(Orac
83         at oracle.jdbc.driver.OraclePreparedStatement.executeUpdate(Ora
84         at envdatec.jevis.tool.jecsvimport.ImportWorker.beginImport(Im
85         at envdatec.jevis.tool.jecsvimport.JECSVImport.<init>(JECSVImp

```

```
86         at envivatec.jevis.tool.jecsvimport.JECSVImport.main(JECSVImpor
87 2007-08-29 10:00:53 ERROR: Can't import file -- database error
88 2007-08-29 10:00:53 Operation time = -1188374447259 msec
89 2007-08-29 10:00:53 *** Operation did not complete
90 2007-08-29 10:00:53 Failure(s) occurred during import of tmpcsv/20070829100042_9.csv
    (column 9) to one or more servers
91 2007-08-29 10:00:53 Some column(s) of files/20070829_100001.csv could not be imported
    into at least one server, leaving the file
92 2007-08-29 10:00:53 Conversion finished, 1 source file(s) found, 0 fully processed, 1 left due
    to errors
93 2007-08-29 10:00:53 WARNING: you should examine files, left in
    [vida84_fast/INSEL_13/files/*]
94 2007-08-29 10:00:53         and manually delete those that contain errors!
95 2007-08-29 12:00:01 Starting ADF session
```

## 5 Qualitätsmanagement-Tool (Insel)

Nachdem im vorherigen Kapitel festgestellt wurde, was für das Qualitätsmanagement des GSM-Netzwerks wichtig ist und anschließend durch die Analyse des Logfiles zudem festgestellt wurde, welche Schlüsselworte für die Auswertung wichtig sind, kann nun für den Funktionsbereich „Lastmanagement“ durch die gesammelten Informationen eine automatisierte Qualitätsüberwachung des GSM-Netzteils realisieren werden. Dies geschieht in zwei voneinander abhängigen Schritten, wobei der erste Schritt als Vorbereitungsschritt betrachtet werden kann, der im Hintergrund arbeitet und die Grundlage für den zweiten Schritt legt. Der zweite Schritt repräsentiert den für den Anwender sichtbaren und anschaulichen Teil.

### 5.1 Vorbereitungs-Tool (JAVA)

Im Vorbereitungsteil geht es vor allem darum, aus den vorliegenden Logfiles die wichtigen Informationen (Datum, Verbindungszeit, Gesamtzeit, Fehlerarten, ...) herauszufiltern und in eine auswertbare Datenform zu bringen, die sowohl für eine direkte Betrachtung als auch für die weitere Verwendung zum Beispiel in einer Datenbank geeignet ist. Dazu ist eine tabellarische Aufarbeitung besonders geeignet.

#### 5.1.1 Auswahl der Programmiersprache.

Für diese Art von Aufgaben eignen sich eigentlich viele Programmiersprachen, insbesondere die, die Stringmanipulationsfunktionen anbieten. Zur Lösung dieser Aufgabe ist wegen der optimierten Verwendung von Strings, die Auswahl auf die Programmiersprache JAVA gefallen. Strings genießen in Java wegen ihres häufigen Vorkommens in Programmen eine besondere Behandlung. Der Compiler behandelt Strings wie primitive Datentypen, indem er zum Beispiel automatisch ein String-Objekt generiert, wenn er auf einen Text stößt. Außerdem beobachtet der Compiler die Literale (Wörter) im Programm und versucht nach Möglichkeit dasselbe Objekt wieder zu verwenden.

Eine weitere Eigenschaft von String-Objekten ist, dass in der Klasse *java.lang.String* keine Methoden definiert sind, um den Inhalt eines Strings zu manipulieren. Ist es doch einmal nötig, einen String zu verändern, so muss der Umweg über ein *StringBuffer*-Objekt gemacht werden. Diesen Umweg führt der Compiler auch automatisch aus, wenn Strings in einem JAVA Sourceprogramm mit dem „+“-Operator verbunden werden [28, 29].

### **5.1.2 Prinzip des JAVA Vorbereitungs-Tools**

Das Prinzip des JAVA-Programms ist der eines Filters, der die Logfiles Zeile für Zeile einliest und nach Schlüsselbegriffen filtert, dabei werden Zeilen, die den gesuchten Begriff nicht enthalten, ignoriert und nur die betroffenen Zeilen weiterbearbeitet. Die betroffenen Zeilen werden in ein String-Feld zerlegt und je nach Suchbegriff bzw. Schlüsselwort wird das gesuchte Feldelement festgehalten und mit anderen festgelegten Parametern in eine CSV-Zieldatei hineinkopiert, die das Semikolon („ ; “) als Trennzeichen verwendet.

### **5.1.3 Aufbau und Ablauf des JAVA-Programms**

Die folgenden Flussdiagramme zeigen den gesamten Ablauf des JAVA Programms, einschließlich des Unterprogramms „*lastDate*“.

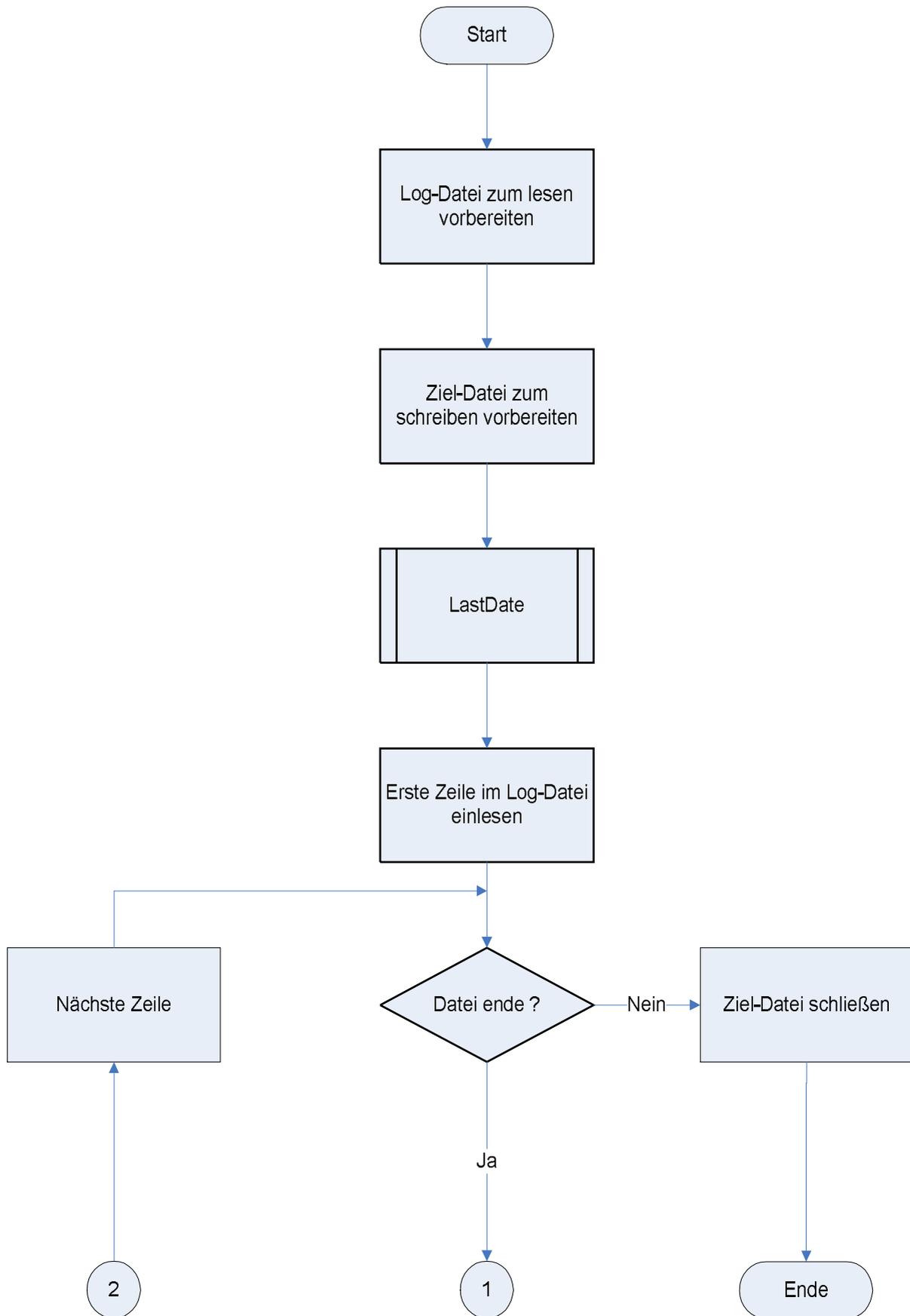


Abbildung 9: Ablaufdiagramm\_1 des JAVA Programms.

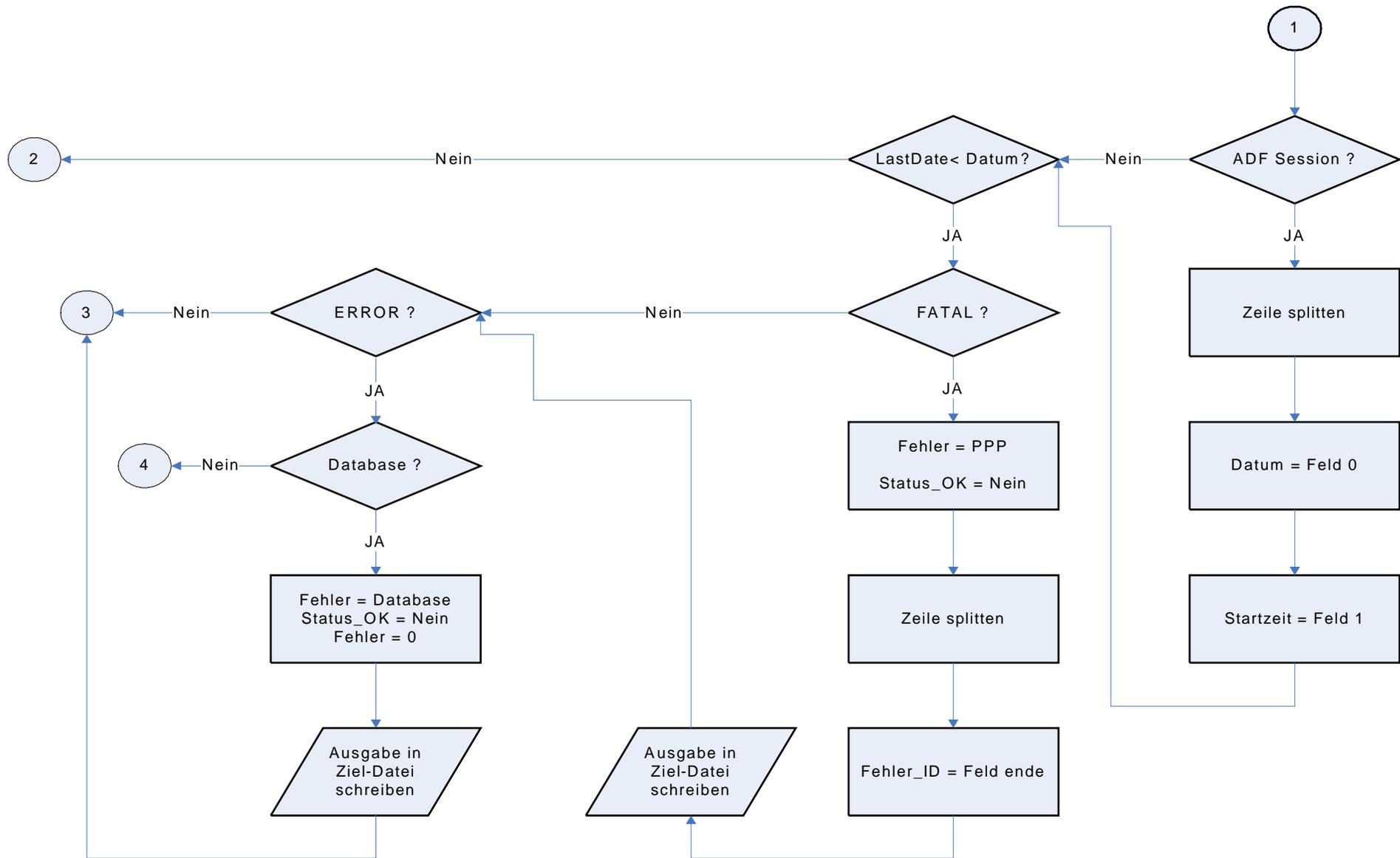


Abbildung 10: Ablaufdiagramm\_2 des JAVA Programms.

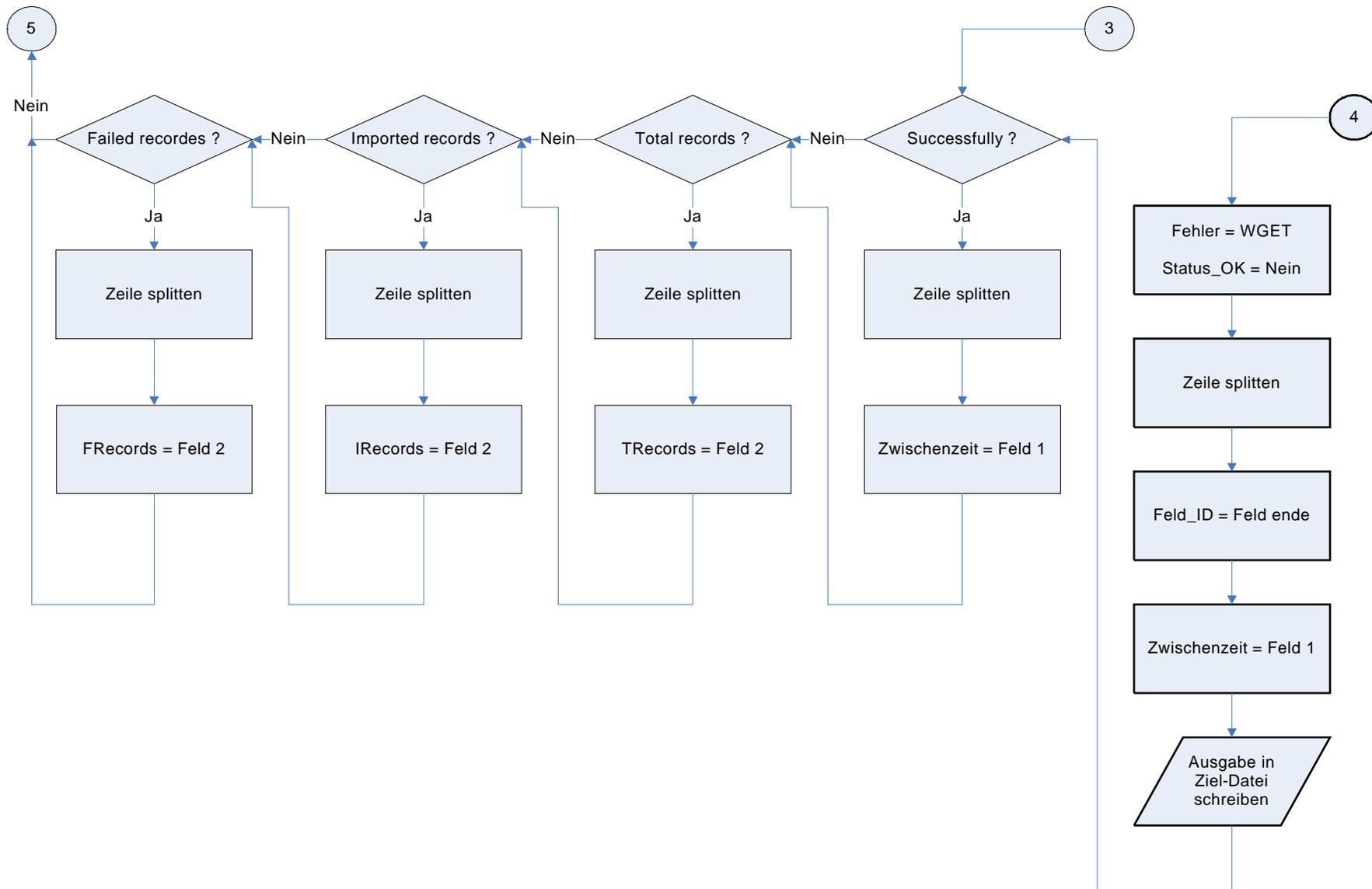


Abbildung 1: Ablaufdiagramm\_3 des JAVA Programms.

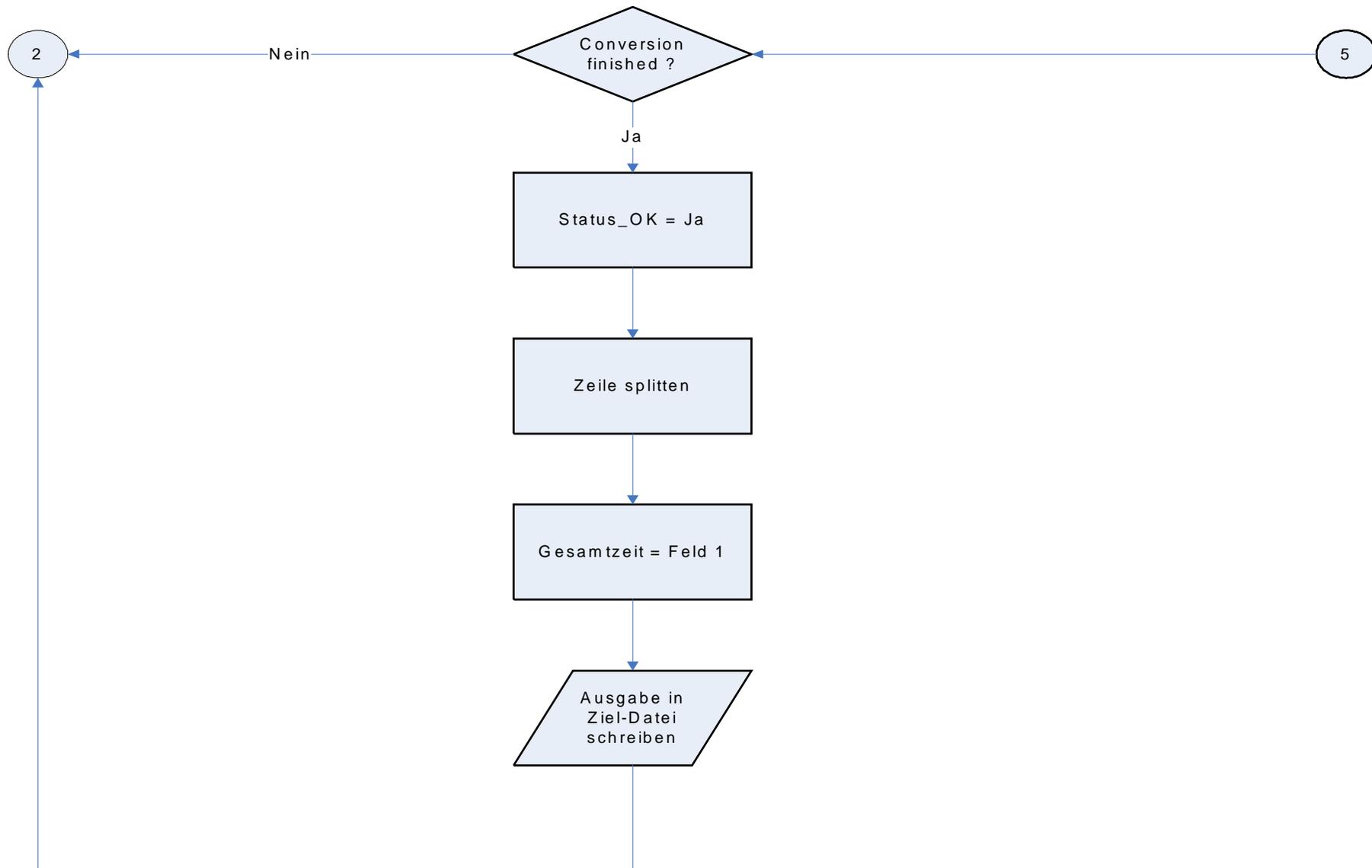


Abbildung 2: Ablaufdiagramm\_4 des JAVA Programms.

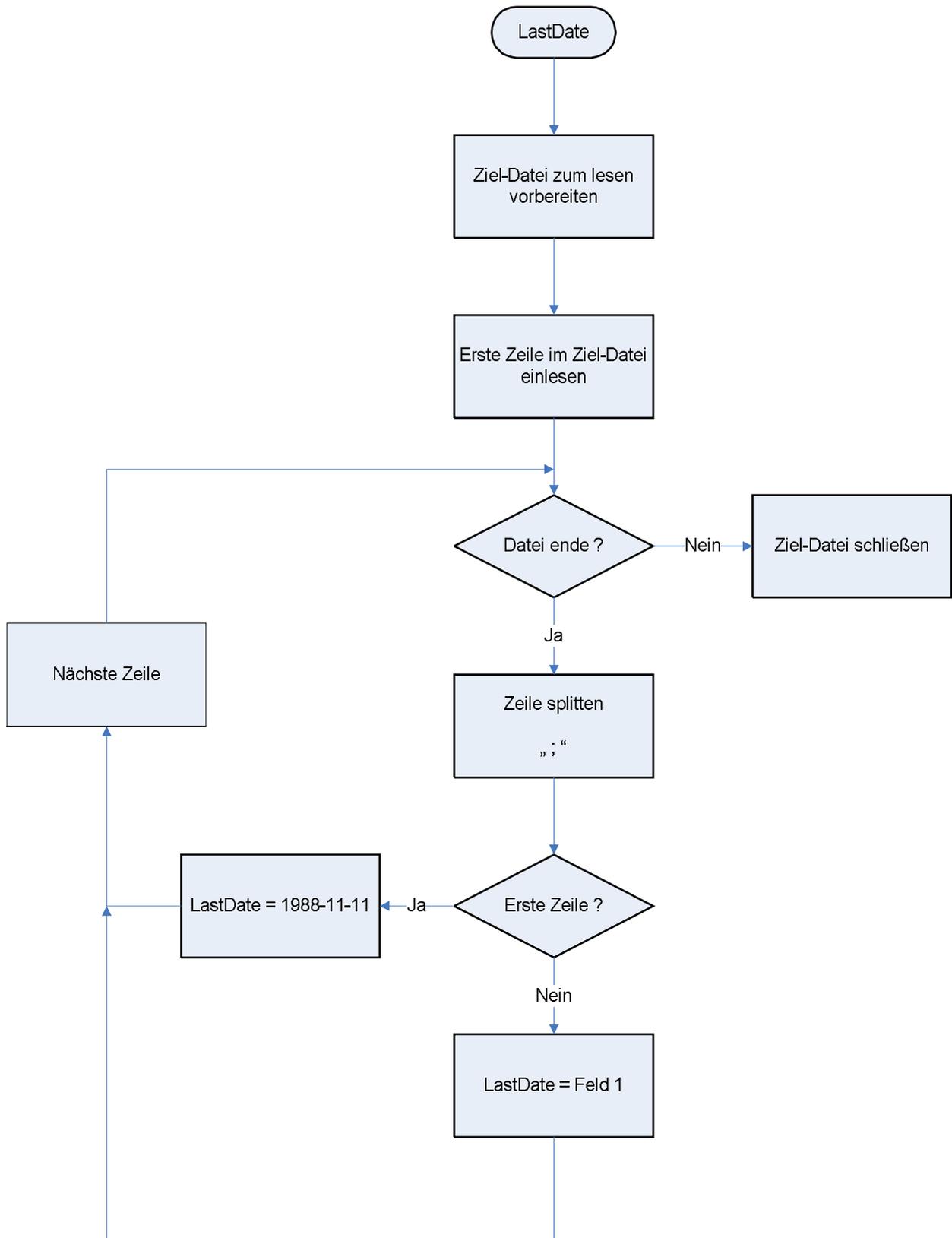


Abbildung 13: Ablaufdiagramm des Unterprogramms LastDate.

Im folgenden Listenabschnitt sind die Objekte zu sehen, die für das Lesen und Schreiben aus- bzw. in eine Datei zuständig sind.

```

FileReader fr = new FileReader(filename);           //Leser-Datei Objekt erzeugen
BufferedReader br = new BufferedReader(fr);        //Zwischenspeicher Objekt
                                                    erzeugen
FileWriter fw = new FileWriter(zielDatei, true);   //schreib-Datei Objekt erzeugen
PrintWriter pw = new PrintWriter(fw);             //Objekt zur Zeilenorientierte Ausgabe
                                                    erzeugen

```

Durch das „*FileReader*“ Objekt wird es möglich, anhand des Logfile-Namen das Logfile zu öffnen und dessen Inhalt einzulesen. Mit dem anschließend angehängten „*BufferedReader*“ Objekt werden die eingelesenen Daten im Hauptspeicher zwischengepuffert, außerdem ermöglicht der „*BufferedReader*“ mit Hilfe von *readline()* ganze Zeichen einzulesen, ohne dass die Zeichen einzeln eingelesen und überprüft werden muss, ob man an ein Zeilenende gekommen ist.

Für das Schreiben in eine Datei erzeugt das „*FileWriter*“ Objekt eine Datei zum Schreiben, falls die Ziel-Datei noch nicht vorhanden ist. Der Parameter „*true*“ bewirkt, dass die neuen Daten angehängt werden, sonst wird die Ziel-Datei einfach überschrieben. Mit dem „*PrintWriter*“ Objekt können alle primitiven Datentypen in textueller Form zeilenorientiert ausgegeben werden, bevor sie den „*FileWriter*“ übergeben werden [30, 31].

Der folgende Listenabschnitt stellt ein Beispiel zur Verdeutlichung der Vorgehensweise beim Ausschuchen der wichtigen Zeilen und die darin gesuchten Informationen im Logfile dar.

```

if( record.contains("ADF session" )){           //testet die eingelesene Zeile auf das
                                                    gesuchte begriff

    record2 = record.split(" ");                //splittet die ausgewählten Zeilen nach
                                                    jeder leer Zeichen

    datum = record2[0];                        //speichert das Datum aus der
                                                    ausgewählten Zeile

```

Mit der „*contains*“ Methode kann bestimmt werden, ob in der bereits im String-Objekt „*record*“ gespeicherten Zeile der Suchbegriff enthalten ist. Hier ist es der Begriff "*ADF session*", der den Start des Verbindungsprozesses signalisiert. Ist dies der Fall, wird die ausgewählte Zeile durch die „*split*“ Methode nach jedem Leerzeichen zerlegt und in den String-Feld-Objekt „*record2*“ gespeichert. Nach der Zerlegung der Zeile kann die gesuchte Information hier das Datum aus dem ersten Feldelement (siehe den ersten Logfileabschnitt im Kapitel 4.2.3) „*record2[0]*“ direkt in der dazugehörigen Variable „*datum*“ gespeichert werden.

Der folgende Listenabschnitt stellt eine wichtige Verzweigungsstelle im Programm dar. Hier wird verhindert, dass die bereits bearbeiteten Zeilen im Logfile – die Daten der vorherigen Tage - noch mal bearbeitet werden.

```
....
if (dt.before(sdf2.parse(datum))){           //testet ob das eingelesene Datum neu
                                           oder alt ist
....
```

Durch die „before“ Methode, die zur *Date* Klasse gehört, wird verglichen, ob das gerade eingelesene Datum nach seiner Konvertierung durch die „parse“ Formatierungsmethode vom String-Format zum Date-Format größer ist als das letzte Datum in derselben Ziel-Datei. Die Ziel-Datei wird durch das Unterprogramm „lastDate“ geöffnet und nach dem letzten eingetragenen Datum durchsucht, danach wieder geschlossen. Dies geschieht jedes Mal bevor sie zum Schreiben der neuen Daten geöffnet wird.

#### 5.1.4 Fehlerbehandlung (Exceptions)

Exceptions bilden in Java eine der wichtigsten Technologien zur Fehlererkennung und -behandlung. Mit der Ausnahmebehandlung lassen sich in JAVA durch die Try- und Catch- Blöcke mögliche auftretende Abstürze im Programmverlauf sicher behandeln.

Im JAVA-Vorbereitungstool ist die Eventualität des Absturzes des Programmablaufs nicht auszuschließen, da die Eingangsdaten bzw. die Logfiles sich aus Protokolldaten verschiedener Protokolle bzw. Dienste (PPP, WGET, Oracle) zusammensetzen. Bei solchen Zusammenstellungen kann das eine oder andere Protokoll sich einen kleinen Programmfehler einfangen: Anstatt von Datum und Uhrzeit befinden sich dann im Logfile andere abweichende Daten die mit dem Datumsformat bzw. Zeitformat nicht kompatibel sind. Es ist auch möglich, dass Datum und Uhrzeit sich nicht an der richtigen Stelle befinden. Solche Anomalien im Logfile haben den Absturz des Programms zur Folge. Für solche Fälle sind im JAVA-Programm Try- und Catch-Blöcke in den kritischen Stellen eingebaut, die als Aufgabe haben, solche Fehleinformationen aufzufangen, durch Default Werte zu ersetzen um einen möglichen Programmabsturz in solchen Fällen zu unterbinden.

Der folgende Listenabschnitt zeigt ein Beispiel der Datumsfehlerbehandlung.

```
try{
    Date test = new Date();

    //testet ob das Datum diese Format "yyyy-mm-dd" hat.

    test = sdf2.parse(datum);

}catch (ParseException e) {

    datum = "1888-11-11";

}
```

In diesem „try“ und „catch“ –Block wird als Erstes getestet, ob das aufgezeichnete Datum das richtige oder ein abweichendes Format aufweist, so wird dieser Verstoß vom „catch“ –Block aufgefangen und das gewünschte Datum durch ein Default Datum „1888-11-11“ ersetzt, das zur Folge hat, dass diese Zeile im weiteren Programm nicht betrachtet wird.

Bei einer falsch eingelesenen Uhrzeit wird diese durch die Default-Uhrzeit „00:00:00“ ersetzt, um einen Programmabsturz zu verhindern. So lässt das Auftreten dieses Datums in den weiteren Auswertungen darauf schließen, dass beim Logfile ein Fehler aufgetreten ist.

## 5.2 Auswertungstool

Die Ausgangsidee für die Auswertung war, die mit der JAVA-Plattform erstellten Ziel-Dateien mit Hilfe des Microsoft Excel Programms auszuwerten und grafisch anschaulich darzustellen. Dabei werden die gesamten Excel Prozeduren mit Visual Basic for Applications (VBA) automatisiert, eine von Visual Basic abgeleitete Skriptsprache die zur Steuerung von Programmabläufen in den Microsoft-Office-Programmen entwickelt wurde. Allerdings hätte sich die Lösung des Problems etwas umständlicher gestaltet, da zum Beispiel die Datenlogger nicht alle am selben Tag in Betrieb genommen wurden. Somit hat jeder Logger ein unterschiedliches Anfangsdatum, was für die die Automatisierung der Auswertungsabläufe, insbesondere bei der Betrachtung des gesamten Netzwerks, sehr problematisch wäre. Bei Access hingegen lässt sich dieses Problem einfacher lösen, da Access ein Datenbankmanagementsystem ist und sich bei Datenbanken Gemeinsamkeiten zwischen Tabellen viel leichter durch SQL-Abfragen ermitteln und gruppieren lassen. Dies und einige weitere Argumente, die im Laufe dieses Kapitel erwähnt werden, waren letztendlich maßgebend für die Entscheidung zugunsten der Auswertung in Access anstelle von Excel, obwohl es sich hier nicht um eine Datenbank Aufgabe im klassischen Sinne handelt.

## 5.2.1 Was ist eine Datenbank?

Im Allgemeinen versteht man unter einer Datenbank eine Ansammlung von Daten oder Informationen in Form von Tabellen, die miteinander in Beziehung stehen und sich auf ein bestimmtes Thema beziehen. Mit Datenbanken werden beispielsweise Aufträge, Kundendaten, Archivinformationen oder, wie in unserem Fall, die Verbindungsdaten der einzelnen Zählergeräte verwaltet.

In der Datenverarbeitung spricht man von einem Datenbanksystem, wenn dieses aus einer Datensammlung und einem Verwaltungssystem besteht. Die Datensammlung ist nach einer bestimmten Struktur aufgebaut und das Verwaltungssystem organisiert die Zugriffe auf die Datenbank und deren Bestand, was nichts anderes heißt, als dass dieses Verwaltungssystem dem Nutzer ermöglicht, bestimmte Informationen zu erfassen, zu speichern, wieder aufzufinden, zu aktualisieren oder weitere Operationen mit den Daten durchzuführen, wie z. B. Daten nach bestimmten Kriterien zu sortieren [32].

Access zum Beispiel ist eine Datenbank aus dem Hause Microsoft, die im Gegensatz zu zahlreichen anderen Datenbankprogrammen außer den typischen Datenbankaufgaben und -operationen, die Darstellung von Charts und Diagrammen unterstützt, was für unsere Aufgabe sehr hilfreich ist, da für den Anwender Grafiken anschaulicher sind und daher mehr Aussagekraft haben als Tabellen.

## 5.2.2 Anforderung und Regeln für eine Datenbank.

Es ist wichtig zu erwähnen, dass Access und somit auch die in Access angelegte Datenbank für das Projekt „Insel“ relationale Datenbanken sind, die bestimmte Mindestanforderungen und Regeln erfüllen müssen [33].

### 5.2.2.1 Mindestanforderungen an relationale Datenbanken

- Alle Informationen müssen in Tabellen dargestellt werden um eine gewisse Flexibilität zu erreichen.

Beispiel:

Datum	Status_OK	Startzeit	Verbindungszeit	Fehler
02.06.2007	Ja	00:06:31	00:01:26	OK
03.06.2007	Nein	00:06:27	00:00:00	WGET

**Tabelle 6: Tabellarisierte Form der Informationen.**

- Die Daten müssen mindestens mit den folgenden Operationen bearbeitet werden:
  - Selektion  
Alle Zeilen der Ausgangstabelle, die eine bestimmte Bedingung erfüllen, werden in eine Ergebnistabelle kopiert

Beispiel:

Alle Zeilen bei denen der Status\_Ok „Ja“ ist.

- Verbund von Tabellen (Join)  
Zwei Tabellen mit mindestens einem gemeinsamen Attribut (Feldnamen) werden verbunden.

Beispiel:

Mit einer zusätzlichen Tabelle der Form

Datum	Fehler_ID
02.06.2007	
03.06.2007	6

**Tabelle 7: Tabelle vor dem Verbund.**

kann folgende Tabelle erzeugt werden:

Datum	Status_OK	Startzeit	Verbindungszeit	Fehler	Fehler_ID
02.06.2007	Ja	00:06:31	00:01:26	OK	
03.06.2007	Nein	00:06:27	00:00:00	WGET	6

**Tabelle 8: Tabelle nach dem Verbund.**

- Projektion  
Die Ergebnistabelle enthält alle Spalten der Ausgangstabelle, die eine bestimmte Bedingung erfüllen.

Beispiel:

Status\_OK ist „Nein“ und Fehler nicht „WGET“.

Die benutzten Operationsbefehle sind ein Teil der zum Abfragen und Manipulieren von Daten meist eingesetzten Datenbanksprache SQL (*Structured Query Language*)

### 5.2.2.2 Regeln der Relationalität

1985 wurden von E. F. Codd zwölf Regeln veröffentlicht [34], die eine relationale Datenbank im strengen Sinne definieren, was aber nicht ausschließt, dass auch Datenbanken, die nicht all diese Regeln erfüllen als relationale Datenbanken betrachtet werden. Es werden hier die wichtigsten Regeln für die Datenbank des Projekts „Insel“ genannt:

- **Darstellung von Information**

Alle Informationen müssen logisch in Tabellen dargestellt sein. Insbesondere Daten und Definitionen von Tabellen und Attributen.

- **Zugriff auf Daten**

Jeder Wert einer relationalen Datenbank muss logisch durch eine Kombination von Tabellename, Primärschlüssel und Attributname (Spaltenname) auffindbar sein. Dies bedeutet, dass in einer Tabelle an jedem Schnittpunkt einer Zeile mit einer Spalte nur ein Wert stehen darf.

Beispiel:

Tabellename: Liegenschaft\_01

Primärschlüssel: Datum = 02.06.2007

Attributname: Verbindungszeit = 00:01:26

- **Systematische Behandlung von Nullwerten**

Nullwerte stellen in Attributen, die nicht Teil eines Primärschlüssels sind, fehlende Information dar und werden durchgängig gleich, insbesondere unabhängig vom Datentyp des Attributes, behandelt.

Beispiel:

Man kann also nicht in numerischen Feldern bei fehlenden Daten das Feld leerlassen, und bei Textfeldern das Zeichen -- einfügen. Fehlende Informationen werden meist mit NULL bezeichnet.

- **Struktur einer Datenbank**

Die Datenbankstruktur wird in derselben logischen Struktur wie die Daten gespeichert, also in Tabellen. Dazu muss die Struktur aller Tabellen, die zu einer Datenbank gehören, in einer Tabelle (dem Katalog) zugänglich sein. Diese Forderung bedingt, dass sich eine Änderung im Katalog automatisch in einer geänderten Datenbankstruktur auswirkt!

- **Abfragen und Editieren ganzer Tabellen**

Abfrage- und Editieroperationen müssen als Operanden ganze Tabellen und nicht nur einzelne Sätze erlauben

- **Logische Unabhängigkeit der Daten**

Anwendungen und Zugriffe dürfen sich logisch nicht ändern, wenn Tabellen so geändert werden, dass alle Information erhalten bleibt (zum Beispiel beim Aufspalten einer Tabelle in zwei Tabellen)

- **Unabhängigkeit der Integrität**

Alle Integritätsbedingungen müssen in der Abfragesprache definierbar sein und in Tabellen dargestellt werden. Das System muss mindestens die folgenden Integritätsbedingungen prüfen:

- Vollständigkeitsintegrität (*Entity Integrity, Existential Integrity*)  
Ein Primärschlüssel muss eindeutig sein und darf insbesondere keinen Nullwert enthalten.

- Beziehungsintegrität (*Referentielle Integrität, Referential Integrity*)  
Zu jedem Fremdschlüsselwert existiert ein Primärschlüsselwert

### 5.2.3 Anpassungen für die Datenbank Insel

Die meisten der erwähnten Anforderung und Regeln sind schon ohne große Veränderungen in der Vorarbeit in JAVA erfüllt. Zum Beispiel stehen alle Daten in Tabellen und sind mit den SQL-Operationsbefehlen bearbeitbar, Trotzdem sind noch Anpassungen nötig, um die übrigen Anforderungen zu erfüllen. So musste zum Beispiel im JAVA Programm dafür gesorgt werden, dass alle Nullwerte, insbesondere die numerischen Felder, nicht leer bleiben, sondern mit Nullen (00:00:00) oder einen Defaultwert ausgefüllt werden. Dieser Schritt wäre zum Beispiel bei der Auswertung mit EXCEL nicht nötig gewesen.

Nach dieser Anpassung im Javaprogramm sind alle Anforderungen und Regeln bis auf die Unabhängigkeit der Integritätsregel erfüllt. Diese Regel ist besonders wichtig, denn der Primärschlüssel ist der bedeutendste Bestandteil einer Datenbanktabelle. Es handelt sich dabei um ein Feld, das jeden Datensatz einer Tabelle eindeutig identifiziert, der in jedem Datensatz einen anderen Wert aufweist. Es darf keine Zeilen geben, die miteinander identisch sind! So ist es allein aufgrund der Existenz dieses Feldes ausgeschlossen, dass in einer Tabelle jemals zwei absolut identische Datensätze vorkommen können, da sie sich zumindest im Inhalt dieses Feldes unterscheiden.

In folgender Beispieltabelle, die das Java Programm produziert, ist eindeutig zu sehen, dass die Regel im Sinne relationaler Datenbanken nicht erfüllt wird.

tab_01_Liegenschaft_1					
Datum	Status_OK	StartZeit	VerbindungsZeit	GesamtZeit	Fehler
11.07.2007	Nein	00:05:11	00:00:00	00:00:00	PPP
12.07.2007	Nein	00:05:02	00:00:00	00:00:00	PPP
13.07.2007	Nein	00:06:11	00:00:00	00:00:00	PPP
13.07.2007	Ja	12:01:37	00:02:22	00:02:38	OK
13.07.2007	Ja	12:28:02	00:00:59	00:01:11	OK
13.07.2007	Nein	16:36:11	00:00:00	00:00:00	PPP
13.07.2007	Nein	21:53:09	00:00:00	00:00:00	PPP
13.07.2007	Nein	22:37:31	00:00:00	00:00:00	PPP
13.07.2007	Nein	22:57:36	00:00:00	00:00:00	PPP
13.07.2007	Nein	23:48:55	00:00:00	00:00:00	PPP
14.07.2007	Nein	00:11:03	00:00:00	00:00:00	PPP
14.07.2007	Nein	11:10:45	00:00:00	00:00:00	PPP
14.07.2007	Nein	19:05:30	00:00:00	00:00:00	PPP
14.07.2007	Nein	20:38:53	00:00:00	00:00:00	PPP
15.07.2007	Nein	00:06:33	00:00:00	00:00:00	PPP

Tabelle 9: Liegenschaftstabelle mit mehreren Verbindungsversuchen an einem Tag.

Im Normalfall werden die Daten durch den Server vom Datenlogger nur einmal pro Tag abgefragt. Es kommt aber öfter vor, wie die Tabelle 9 ab Zeile 3 zeigt, dass während der Testphase oder durch ein Problem bedingt die Daten mehr als nur einmal pro Tag abgefragt werden. Somit ist es ausgeschlossen, dass das Feld Datum als Primärschlüssel eingesetzt werden kann.

Zur Lösung dieses Problems ist es nötig, einen Mehr-Felder-Primärschlüssel zu definieren. Die Kombination vom Feld *Datum* und *Startzeit* ist eindeutig. Es kann keinen zwei Abfragen geben, die beide am gleichen Tag in der gleichen Stunde, Minute und Sekunde durchgeführt werden.

Abbildung 14 zeigt, wie der Mehr-Felder-Primärschlüssel bei einer ACCESS Tabelle aussieht. Die für den Primärschlüssel verwendeten Felder haben in der ersten, grau unterlegten Zeile ein kleines Schlüsselsymbol. Die Realisierung wird im Unterkapitel Module genauer beschrieben.

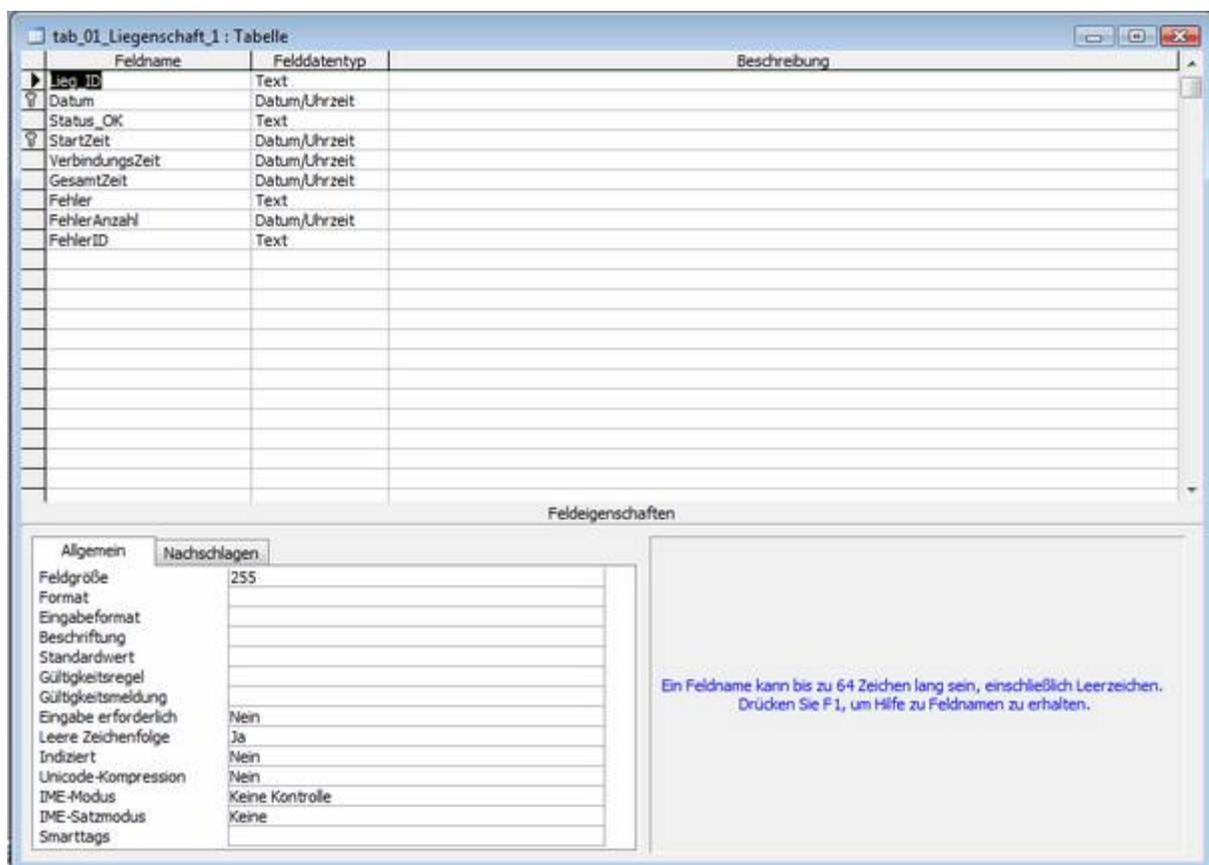


Abbildung 14: Tabelle mit Mehr-Felder-Primärschlüssel in Access.

## 5.2.4 Aufbau der Datenbank Insel

In diesem Abschnitt wird auf die einzelnen Bestandteile der Datenbank Insel eingegangen, um den Zusammenhang zwischen den einzelnen Blöcken zu erläutern. Es wird aber hier nicht explizit Schritt für Schritt erklärt, wie man eine Datenbank aufbaut, sondern vielmehr der allgemeine Aufbau und die Aufgaben der einzelnen

Blöcke erklärt. Außerdem wird in den einzelnen Abschnitten auf die wichtigsten zu beachtenden Besonderheiten dieser Datenbank hingewiesen.

Abbildung 15 zeigt den Aufbau und Zusammenhang zwischen den verschiedenen Objekten der Datenbank Insel.

# 5 Qualitätsmanagement-Tool (Insel)

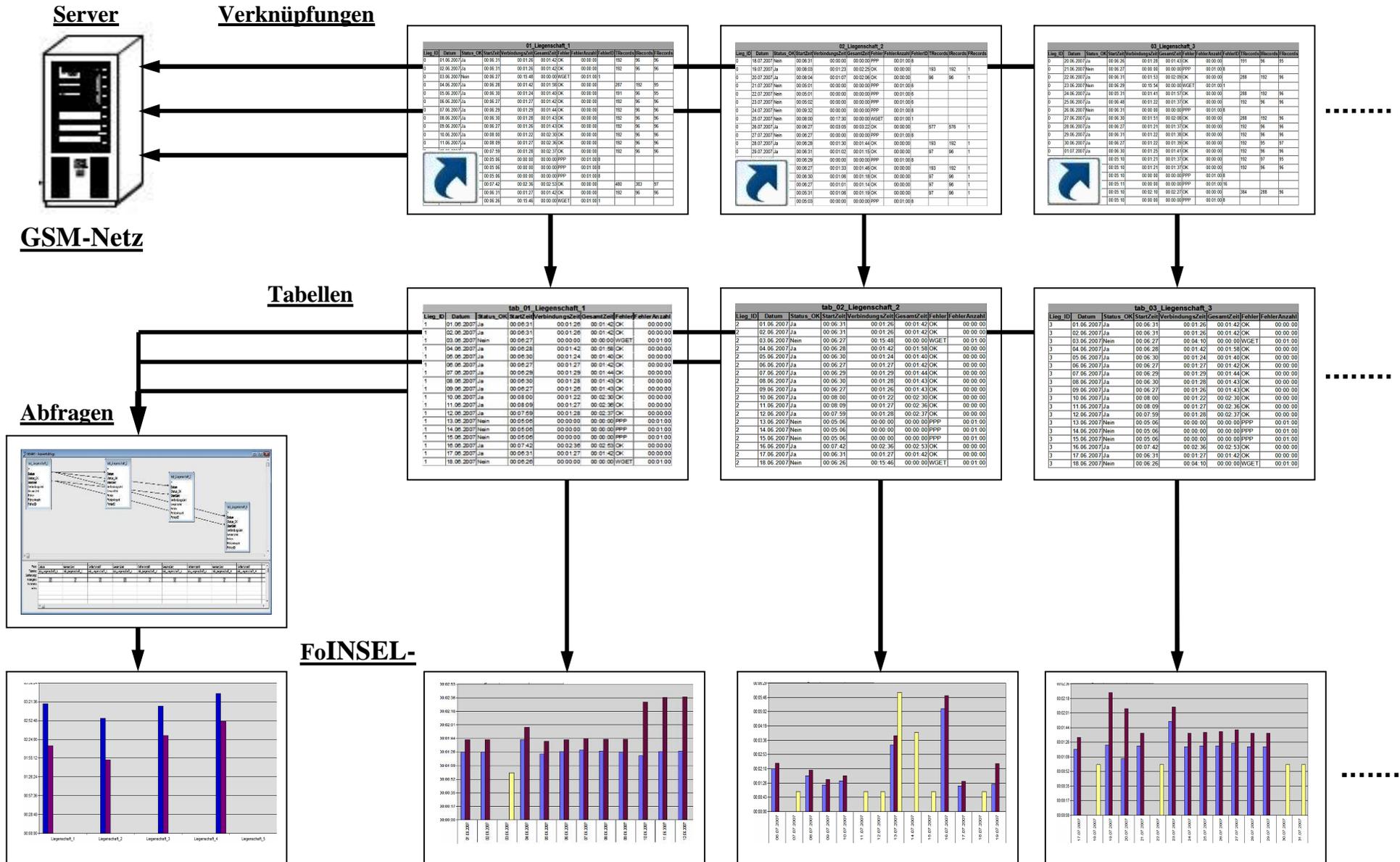


Abbildung 1: Aufbau der Datenbank Insel.

Die im Folgendem vorgeführten Werte und Ergebnisse basieren auf Auswertungen von Logfiles der Firma ENVIDATEC: Dies sind die protokollierten Auslesungsdaten zwischen Auslesegeräten (VIDA84), die im Projekt benutzt wurden, und Stromzählern, die nicht zum Projekt Insel gehören, aber die gleichen Eigenschaften aufweisen. Zurzeit der Diplomarbeit waren die Kommunikationsgeräte noch nicht installiert oder noch nicht mit dem S0-Impuls von Vattenfall synchronisiert (mittlerweile sind ein paar Liegenschaften online, aber die gesammelten Daten würden für eine sinnvolle Auswertung noch nicht ausreichen, weil die zeitliche Periode der gesammelten Daten noch zu klein ist).

### 5.2.4.1 Verknüpfungen

Um die Tabellen mit Daten aufzufüllen gibt es in Access zwei Möglichkeiten: eine besteht darin, die Daten zu importieren, die andere ist, eine Verknüpfung mit dem Datenbestand herzustellen.

Beim Import werden die Daten kopiert und in Access eingebettet, ohne dass es eine weitere Verbindung zur Originaldatei gibt. Diese Methode eignet sich besser bei Ausgangsdaten, die sich nicht oder selten ändern. Besteht aber die Möglichkeit, dass die Originaldaten sich ständig oder täglich ändern, wie es im Projekt „Insel“ der Fall ist, empfiehlt es sich eine Verknüpfung mit dem Datenbestand zu erstellen, um die Beziehung zur der Dynamik der Originaldaten zu bewahren.

Bei der Erstellung von Verknüpfungen wie auch beim Importieren von Daten in Access ist es wichtig, dass die Struktur und der Datentyp der einzelnen Reihen sich nicht ändern. In der Datenbank Insel ist besonders darauf zu achten, dass das richtige Trennzeichen (;) gewählt wird, was hier per Default richtig ist, und dass das richtige Datumsformat gewählt wird. Dieses muss geändert werden, weil das Default Datumsformat bei Access (TMJ) sich vom JAVA Datumsformat (JMT) unterscheidet.

Bei Verknüpfungen ist es auch wichtig, dass der Verknüpfungspfad sich nicht ändert. Verschiebt man die verknüpfte Datei auf ein anders Laufwerk oder in ein anders Verzeichnis, so muss man die Originaldatei neu verknüpfen oder, was viel einfacher ist, mit Hilfe des Tabellenverknüpfungs-Managers die Datenbank darüber informieren, wo sich die betreffende Datei nun befindet.

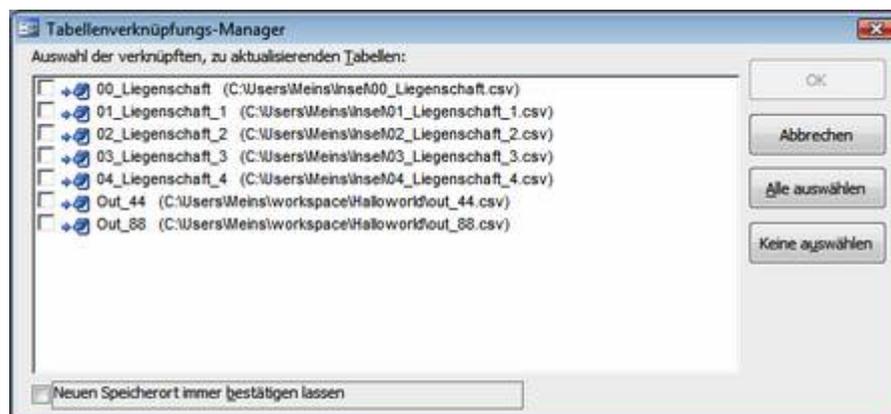


Abbildung 16: Tabellenverknüpfungs-Managers der Access Datenbank.

### **5.2.4.2 Tabellen.**

Im Grunde könnte man direkt mit den erstellten Verknüpfungstabellen weiterarbeiten, wären da nicht die relativ großen Verzögerungszeiten beim Öffnen und Weiterarbeiten mit den verknüpften Daten, die sich bei einer größeren Anzahl von Tabellen und Datenmengen bemerkbar machen. Deshalb ist es hier ratsam aus den Verknüpfungstabellen Kopien mithilfe der Tabellenerstellungsabfrage zu erstellen, die es ermöglichen, bestimmte Daten aus anderen Tabellen oder Abfragen heraus zu filtern und damit physikalisch unabhängige Tabellen zu erstellen. Dies hat außer dem eben angesprochenen Zeitvorteil den weiteren Vorteil, dass die Originaldaten nicht mehr für die Zeit in der man mit den Daten arbeiten will blockiert sind.

In den Tabellen der Insel Datenbank wurden aus den verknüpften Daten nur die Daten, die für die Weiterarbeitung relevant sind, herauskopiert. Außerdem wurden bei der Tabellenerstellung einige ergänzende Änderungen durchgeführt, wie das Eintragen der richtigen Liegenschaft\_ID und Primärschlüssel (Ausführliches dazu gibt es im Abschnitt Module).

Die Tabellen 10 und 11 zeigen eine Verknüpfung und eine Tabelle im direkten Vergleich.

01_Liegenschaft_1											
Lieg_ID	Datum	Status_OK	StartZeit	VerbindungsZeit	GesamtZeit	Fehler	FehlerAnzahl	FehlerID	TRecords	IRecords	FRecords
0	01.06.2007	Ja	00:06:31	00:01:26	00:01:42	OK	00:00:00		192	96	96
0	02.06.2007	Ja	00:06:31	00:01:26	00:01:42	OK	00:00:00		192	96	96
0	03.06.2007	Nein	00:06:27	00:00:00	00:00:00	WGET	00:01:00	1			
0	04.06.2007	Ja	00:06:28	00:01:42	00:01:58	OK	00:00:00		287	192	95
0	05.06.2007	Ja	00:06:30	00:01:24	00:01:40	OK	00:00:00		191	96	95
0	06.06.2007	Ja	00:06:27	00:01:27	00:01:42	OK	00:00:00		192	96	96
0	07.06.2007	Ja	00:06:29	00:01:29	00:01:44	OK	00:00:00		192	96	96
0	08.06.2007	Ja	00:06:30	00:01:28	00:01:43	OK	00:00:00		192	96	96
0	09.06.2007	Ja	00:06:27	00:01:26	00:01:43	OK	00:00:00		192	96	96
0	10.06.2007	Ja	00:08:00	00:01:22	00:02:30	OK	00:00:00		192	96	96

Tabelle 1: Verbindungstabelle der Liegenschaft\_1.

tab_01_Liegenschaft_1								
Lieg_ID	Datum	Status_OK	StartZeit	VerbindungsZeit	GesamtZeit	Fehler	FehlerAnzahl	FehlerID
1	01.06.2007	Ja	00:06:31	00:01:26	00:01:42	OK	00:00:00	
1	02.06.2007	Ja	00:06:31	00:01:26	00:01:42	OK	00:00:00	
1	03.06.2007	Nein	00:06:27	00:00:00	00:00:00	WGET	00:01:00	1
1	04.06.2007	Ja	00:06:28	00:01:42	00:01:58	OK	00:00:00	
1	05.06.2007	Ja	00:06:30	00:01:24	00:01:40	OK	00:00:00	
1	06.06.2007	Ja	00:06:27	00:01:27	00:01:42	OK	00:00:00	
1	07.06.2007	Ja	00:06:29	00:01:29	00:01:44	OK	00:00:00	
1	08.06.2007	Ja	00:06:30	00:01:28	00:01:43	OK	00:00:00	
1	09.06.2007	Ja	00:06:27	00:01:26	00:01:43	OK	00:00:00	
1	10.06.2007	Ja	00:08:00	00:01:22	00:02:30	OK	00:00:00	

Tabelle 2: Erstellte Tabelle der Liegenschaft\_1 aus der Verknüpfungstabelle

### 5.2.4.3 Abfragen

Abfragen sind eigentlich der häufigste Einsatzzweck von Datenbanken. Mit Abfragen lässt sich eine Auswahl der zu betrachtenden Spalten treffen, lassen sich Datensätze sortieren oder beschränken, Daten aus mehreren Tabellen zusammenstellen, neue Felder definieren, Berechnungen durchführen, neue Tabellen erstellen und weitere Dinge. Die Möglichkeit solche vordefinierten Abfragen durchführen zu können, die dann einen Report über die zu überwachenden Aspekte des GSM Netzteils darstellen war der Hauptgrund zur Auswertung der Daten mit Access (und nicht mit Excel).

In der Datenbank Insel werden die Abfragen dazu genutzt, um eine Gesamtübersicht der Verbindungszeit, der Gesamtzeit sowie der Fehler von allen Liegenschaften auf einem Blick zu ermöglichen.

Durch die Auswahlabfrage ist es möglich, die gesuchten Felder aus allen Liegenschaften herauszuziehen und in einer Tabelle darzustellen. In Abbildung 17 ist die Auswahlabfrage des *Datum* von Tabelle01 bzw. Liegenschaft\_1 und dann stets nacheinander die Gesamtzeit und die Fehleranzahl aller Tabellen bzw. Liegenschaften zu sehen.

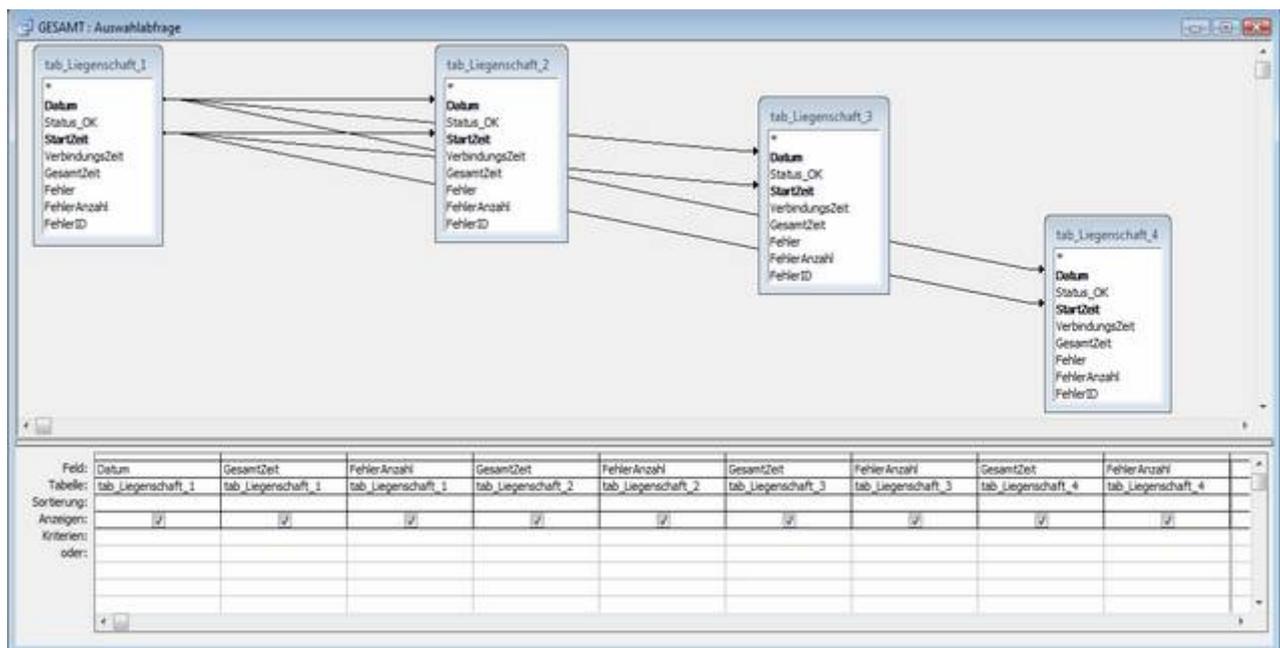


Abbildung 17: Auswahlabfrage GESAMT der Datenbank Insel.

Im oberen Abschnitt der Auswahlabfrage sind die an der Abfrage beteiligten Tabellen und die zwischen ihnen definierten Beziehungen zu sehen. Access verwendet bei der Definition einer Beziehung automatisch die Exklusiv-Verknüpfung. Bei dieser werden in zwei miteinander verknüpften Tabellen alle Datensätze beider Tabellen selektiert, bei denen die Inhalte der verknüpften Felder in beiden Tabellen identisch sind. Diese Art von Verknüpfung (bitte hier nicht zu verwechseln mit den Verknüpfungen des Unterkapitels „Verknüpfungen“) ist in unserem Fall nicht wünschenswert, weil somit Access nur die Datensätze beider Tabellen (immer die erste Tabelle mit der folgenden und dann mit der übernächsten Tabelle) selektiert.

Diese weisen sowohl in Liegenschaft\_1 als auch in Liegenschaft\_2 das gleiche Datum (bzw. Datum und Startzeit, weil die beiden den Primärschlüssel bilden) auf, also nur die Gesamtzeiten und Fehleranzahlen bei dem das Datum in beiden Liegenschaften vorhanden ist. Gesamtzeiten, Fehler und das dazugehörige Datum, wo das Datum nur in einer Liegenschaft erscheint, werden nicht angezeigt.

Wichtig für die Datenüberwachung im Projekt Insel ist die Inklusionsverknüpfung. Bei dieser werden alle Datensätze von der ersten Tabelle selektiert und zusätzlich jene Sätze von der zweiten Tabelle, die irgendeinen der in der ersten Tabelle vorhandenen korrespondierenden Werte enthalten. Hierbei gibt es zwei Varianten, eine rechte Inklusionsverknüpfung und eine linke Inklusionsverknüpfung, die beiden Varianten unterscheiden sich dabei nur darin, ob alle Datensätze aus der linken Tabelle selektiert werden sollen und nur die entsprechenden Datensätze aus der rechten Tabelle oder umgekehrt (dies wird nachstehend deutlicher). Im Fall der rechten Inklusionsverknüpfung, die auch in der Datenbank Insel verwendet wird, werden alle im Feld *Datum* der Liegenschaft\_1 eingetragenen Daten selektiert und die dazugehörigen Gesamtzeiten und Fehleranzahlen. Somit werden auch alle Gesamtzeiten und Fehleranzahlen der verknüpften Liegenschaften selektiert, sofern die Liegenschaft\_1 über den größten Zeitraum verfügt. Deshalb ist es sehr wichtig, dass die älteste Liegenschaft in der Tabelle Liegenschaft\_1 vorkommt, damit diese Verknüpfung auch einen Sinn ergibt.

Noch interessanter wäre es, wenn Access noch eine Verknüpfungsvariante hätte, die eine Kombination von Rechts- und Linksinklusionsverknüpfung darstellt, wie sie in anderen relationalen Datenbanken vorhanden ist. Dann wäre dieser Kompromiss mit der ältesten Liegenschaft an der ersten Stelle nicht mehr nötig. Abbild 18 zeigt die drei Verknüpfungsvarianten an, die Access zu Verfügung stellt.

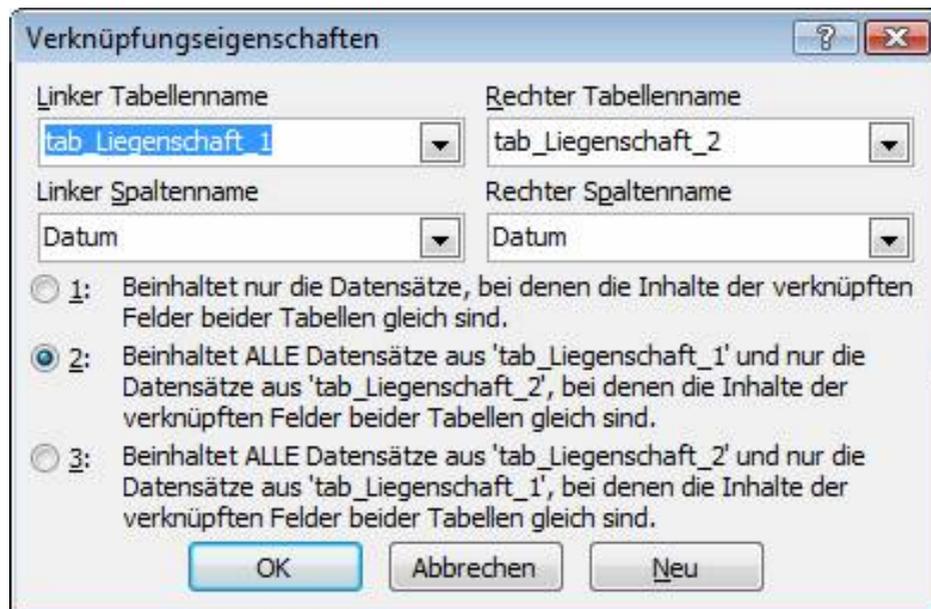


Abbildung 18: Verknüpfungseinstellungen zwischen Liegenschaft\_1 und Liegenschaft\_2.

Dass dieser Kompromiss ausreichend ist für die korrekte Funktionalität des Systems, beruht auf der Tatsache, dass alle Datenlogger täglich abgefragt werden, mit dem Unterschied, dass nicht alle Datenlogger am selben Tag in Betrieb genommen sind,

weil einige Geräte später eingebaut wurden oder werden. Solange der erste in Betrieb genommene Datenlogger an der ersten Stelle bleibt, besteht für die Funktionalität kein Bedenken.

#### **5.2.4.4 Formulare**

Nachdem wir gesehen haben, welche Funktion Tabellen und Abfragen in Access haben, kommen wir jetzt zu dem für die grafische Darstellung der Daten verantwortlichen Teils des Access Repertoires, nämlich den Formularen.

Formulare beruhen meistens auf Tabellen oder Abfragen und geben dem Benutzer die Möglichkeit, Daten bequemer zu gestalten, zu editieren und neue Datensätze einzugeben. Letzteres ist für das Projekt unwichtig, da die neuen Daten in die Datenbank Insel automatisch importiert werden. Wichtiger ist daher ein spezieller Typ von Formularen, der im Grunde auf dem Excel Pivot-Table-Assistenten beruht: Pivot-Table und Pivot-Chart.

Pivot-Table ist ein gutes Werkzeug, um große Datenmengen professionell auszuwerten und zu analysieren. Es ist eine Art Kreuztabelle, die sehr flexibel ist. Man kann damit mit wenigen Mausklicks Daten zusammenfassen, berechnen, verschieben, filtern oder individuell anpassen, ohne die Access Tabellen oder Abfragen, also die Datenbestände, selbst zu ändern.

Access bietet auch die Möglichkeit, Diagramm-Formulare zu erstellen, allerdings sind die Diagramme in Access nicht vergleichbar mit Excel Diagrammen. Sie sind etwas kleiner, umständlicher zu handhaben und bieten nicht die für das Insel-Vorhaben erwünschte Flexibilität.

##### **5.2.4.4.1 Pivot-Tabellen**

Da die Arbeit mit Pivot-Tabellen und Pivot-Diagrammen keine Selbstverständlichkeit ist, soll hier anhand einer Pivot-Tabelle beispielhaft gezeigt werden, wie das Prinzip Pivot funktioniert.

Man kann sich Pivot-Tabellen und Pivot-Diagramme als einen großen Lego-Baukasten (Feldliste) vorstellen, wo sich alle Basisdaten (aus Tabellen oder Abfragen) befinden und aus dem man nach Belieben oder Vorhaben die gewünschten Daten auf einem Spielbrett (Formulare) kombinieren kann.

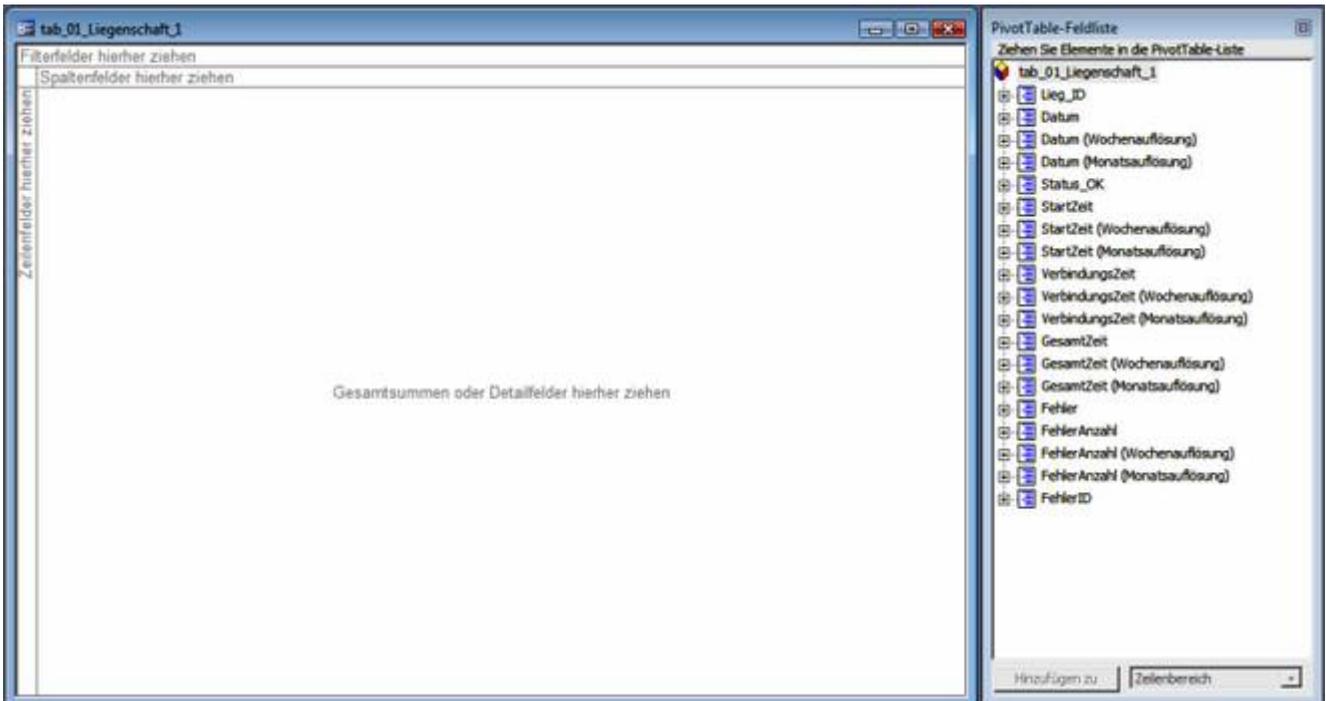


Abbildung 19: Leeres Pivot-Table-Formular der Datenbank Insel.

Abbildung 19 zeigt auf der rechten Seite die Pivot-Tabellen-Feldliste (Lego-Spielkasten) und auf der linken Seite das leere Formularblatt (Spielbrett), will man jetzt zum Beispiel den in Abbildung 20 erreichten Zustand, wo sich die ausgewählten Daten *Datum*, *Verbindungszeit*, *Gesamtzeit* und *Fehler* befinden, im Zusammenhang betrachten, so genügt es, die gesuchten Felder von der Feldliste per Drag & Drop an die gewünschte Stelle im Formularblatt hineinzuziehen.

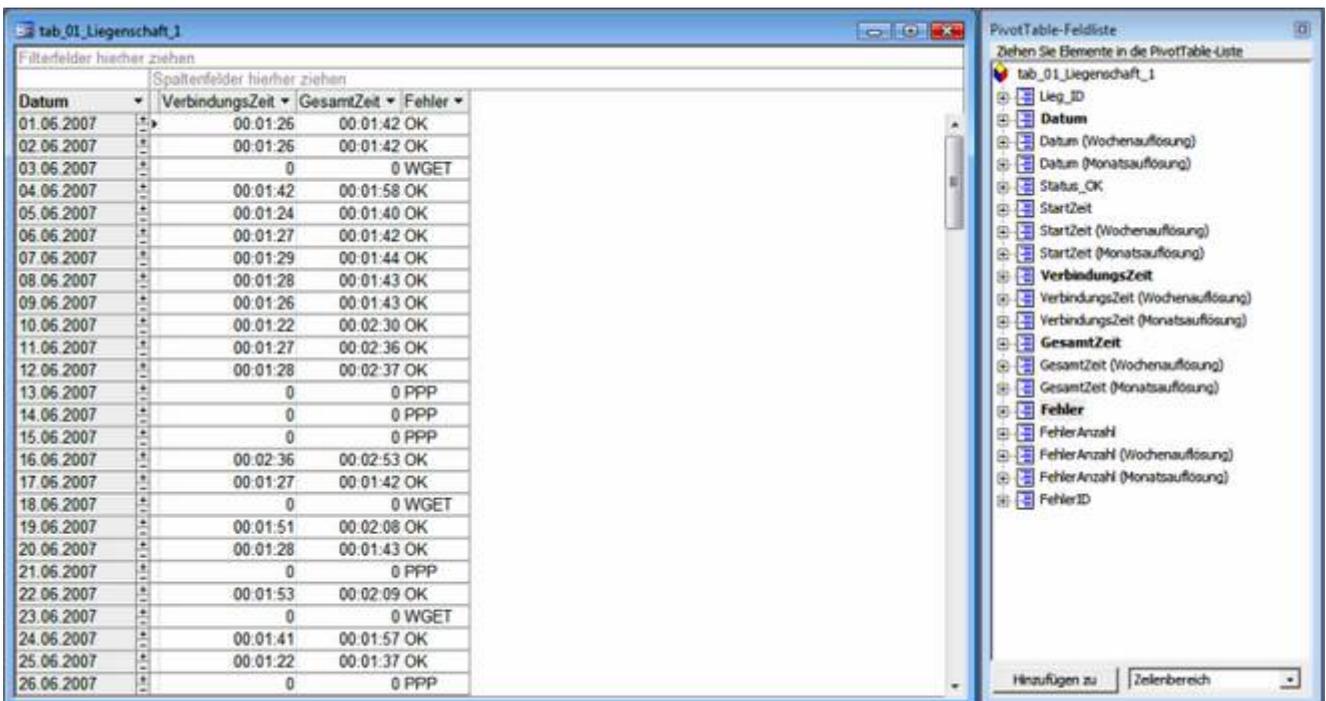


Abbildung 20: Pivot-Table-Formular der Datenbank Insel.

Dass heißt, das *Datum* in das Zeilenfeld, *Verbindungszeit*, *Gesamtzeit* und *Fehler* in die Detailfelder. Will man dort aber nur die Daten haben, wo die Verbindung erfolgreich war, dann genügt es den *Status\_OK* in das Filterfeld hineinzuziehen und anstatt „alle“ nur das Unterfeld „Ja“ anzukreuzen (Abbildung 21). Außerdem kann man leicht über das Symbol **S** in der Symbolleiste die Summe, den Mittelwert etc. für die Auswertung bilden lassen.

Um sich eine andere Konstellation anzeigen zu lassen oder zu analysieren, muss man nur die unerwünschten Felder aus der bestehenden Pivot-Tabelle entfernen und die gewünschten Felder einzufügen, ohne eine neue Pivot-Tabelle erzeugen zu müssen.

Datum	Zeit	GesamtZeit	Fehler
06.06.2007	00:01:26	00:01:42	OK
07.06.2007	00:01:26	00:01:42	OK
08.06.2007	00:01:42	00:01:58	OK
09.06.2007	00:01:24	00:01:40	OK
10.06.2007	00:01:27	00:01:42	OK
11.06.2007	00:01:29	00:01:44	OK
12.06.2007	00:01:28	00:01:43	OK
13.06.2007	00:01:26	00:01:43	OK
14.06.2007	00:01:22	00:02:30	OK
15.06.2007	00:01:27	00:02:36	OK
16.06.2007	00:01:28	00:02:53	OK
17.06.2007	00:01:27	00:01:42	OK
18.06.2007	00:01:51	00:02:08	OK
19.06.2007	00:01:28	00:01:43	OK
20.06.2007	00:01:53	00:02:09	OK
21.06.2007	00:01:41	00:01:57	OK
22.06.2007	00:01:22	00:01:37	OK
23.06.2007	00:01:51	00:02:08	OK
24.06.2007	00:01:21	00:01:37	OK
25.06.2007	00:01:22	00:01:38	OK
26.06.2007	00:01:22	00:01:39	OK
27.06.2007	00:01:25	00:01:41	OK
28.06.2007	00:01:21	00:01:37	OK
29.06.2007	00:01:21	00:01:37	OK
30.06.2007	00:01:21	00:01:37	OK
01.07.2007	00:01:21	00:01:37	OK
02.07.2007	00:01:21	00:01:37	OK
03.07.2007	00:01:21	00:01:37	OK

Abbildung 21: Pivot-Table-Formular der Datenbank Insel für Verbindungserfolgreiche Versuche.

#### 5.2.4.4.2 Formulare der Datenbank Insel

Pivot-Diagramme sind eigentlich nicht anderes als das grafisch anschauliche Gegenstück der Pivot-Tabellen. Hat man eine fertige Pivot-Tabelle, so kann davon leicht das Pivot-Diagramm erzeugt werden. Man kann aber auch Pivot-Diagramme ohne Pivot-Tabellen erstellen. Das Prinzip dabei ist dem von Pivot-Tabellen fast identisch, deshalb wird hier nur auf die verschiedenen in der Datenbank angeführten Formulare eingegangen.

Jede Liegenschaft des Projekts Insel ist mit zwei Formularen repräsentiert: Einem Zeitdiagramm, das die Verbindungszeit, die Gesamtzeit und die Fehleranzahl über den gesamten Zeitraum darstellt seit dem Daten für die Liegenschaft eingeloggt wurden, und einem Fehlerdiagramm, das über den selben Zeitraum eine genauere Fehleranalyse liefert.

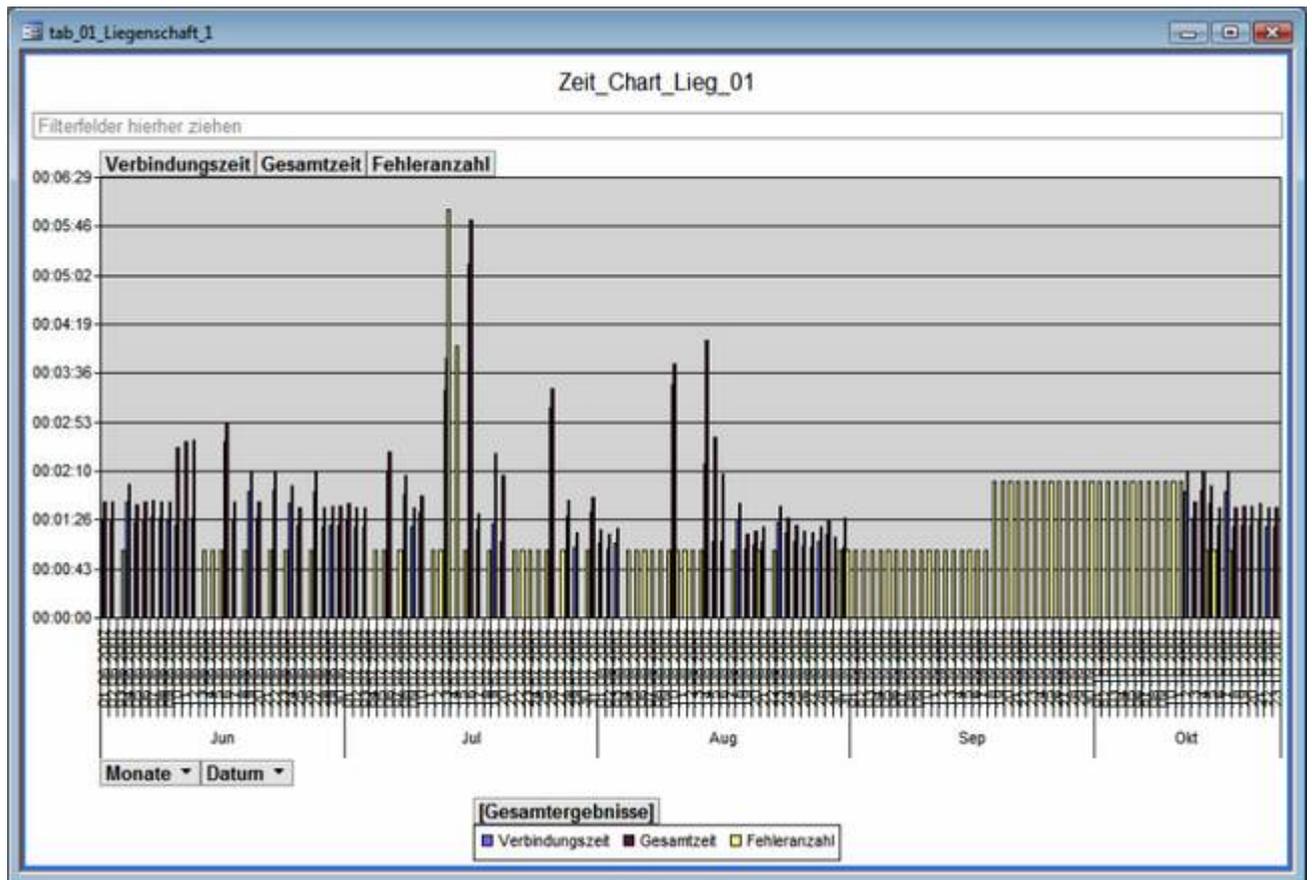


Abbildung 22: Zeit-Chart-Formular der Liegenschaft\_1.

Abbildung 22 zeigt als Beispiel das Zeitdiagramm für die Liegenschaft\_1. Man sieht hier deutlich, dass es etwas unübersichtlich ist, weil die Balken zu dünn sind und zu dicht aneinander stehen. Dieser Zustand verschlechtert sich, je mehr neue Daten hinzukommen. Hier kommt ein weiterer Vorteil der Pivot-Tabellen bzw. Pivot-Diagramme ins Spiel, denn bei Pivot-Ansichten fügt Access zusätzlich zum Datumsfeld verschiedene Datumsauflösungen nach Jahren, Quartalen, Monaten, Wochen etc. hinzu, die man gut für das Gruppieren und Filtern einsetzen kann. So kann man sich das gewünschte Quartal oder den gewünschten Monat darstellen lassen, ohne den Überblick zu verlieren, wie es die Abbildung 23 zeigt.

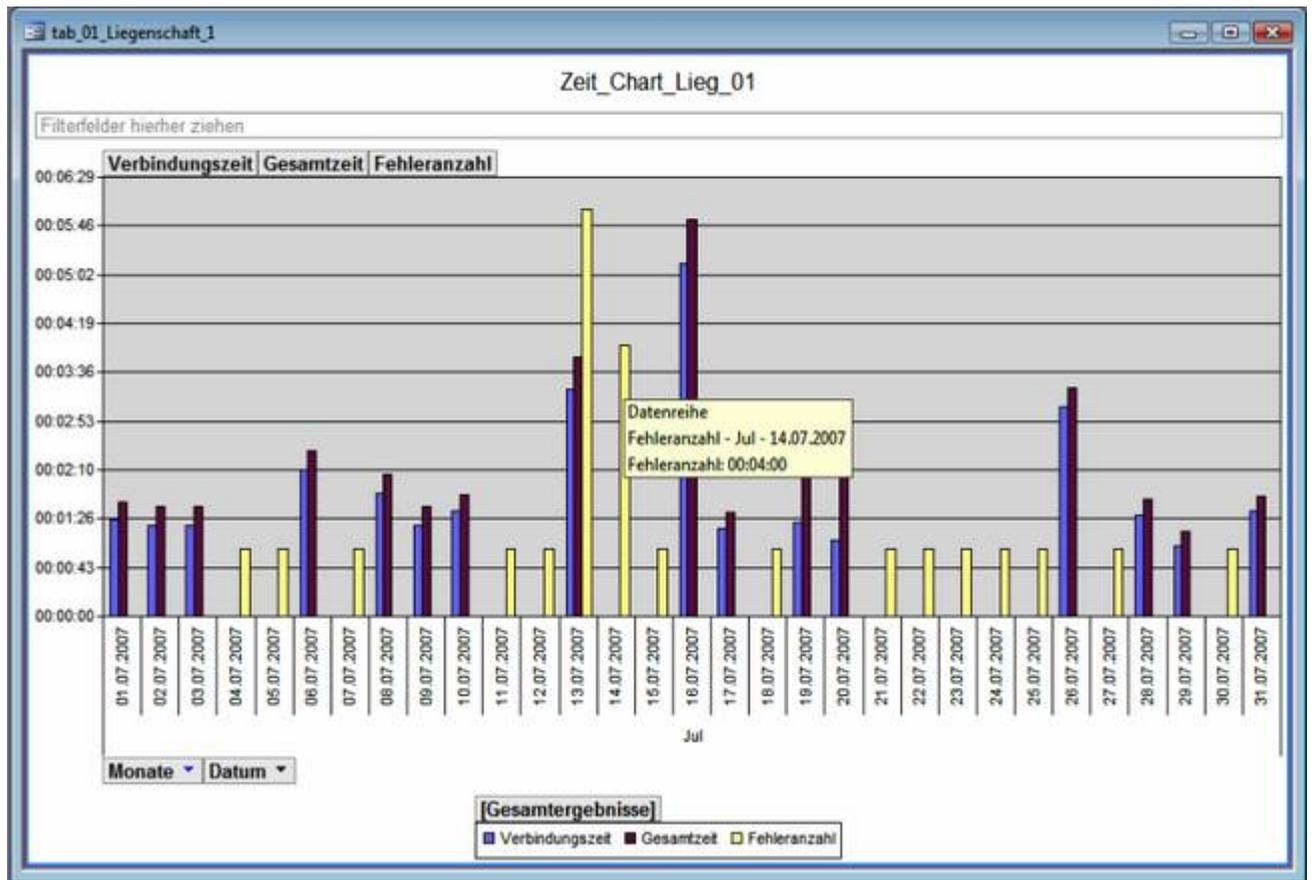


Abbildung 23: Zeit-Chart-Formular der Liegenschaft\_1 für den Monat Juli.

Beim Zeitdiagramm ist noch die Fehleranzahl zu bemerken, die auf den ersten Blick fehl am Platz scheint, zumal Zeit und Anzahl nicht dieselbe Einheit haben und zusammen nicht unbedingt auf dasselbe Diagramm passen. Hier stellt die Fehleranzahl lediglich eine Hilfestellung dar, um zu zeigen, dass an diesem Datum ein Fehler oder mehrere Fehler aufgetreten sind. Je nachdem, wie lang der Fehleranzahlbalken ist. Hat die Fehleranzahl „1“ (Minute) als Wert, heißt es an diesem Tag trat nur ein Fehler auf, ist der Wert gleich 2 (Minuten) so sind es 2 Fehler, die an diesem Tag auftraten und so weiter.

Will man unbedingt wissen, welcher Art von Fehler an diesem Tag auftrat, genügt es, das Fehlerfeld aus der Feldliste im Pivot-Diagramm neben das Datumfeld einzufügen, sodann erscheint die Fehlerart unter dem Balken (Abbildung 24). Auf diese Weise spart man die Erstellung eines zweiten Zeitdiagramms, das nur die Darstellung der Fehlerarten über den Zeitraum verteilt zeigt.

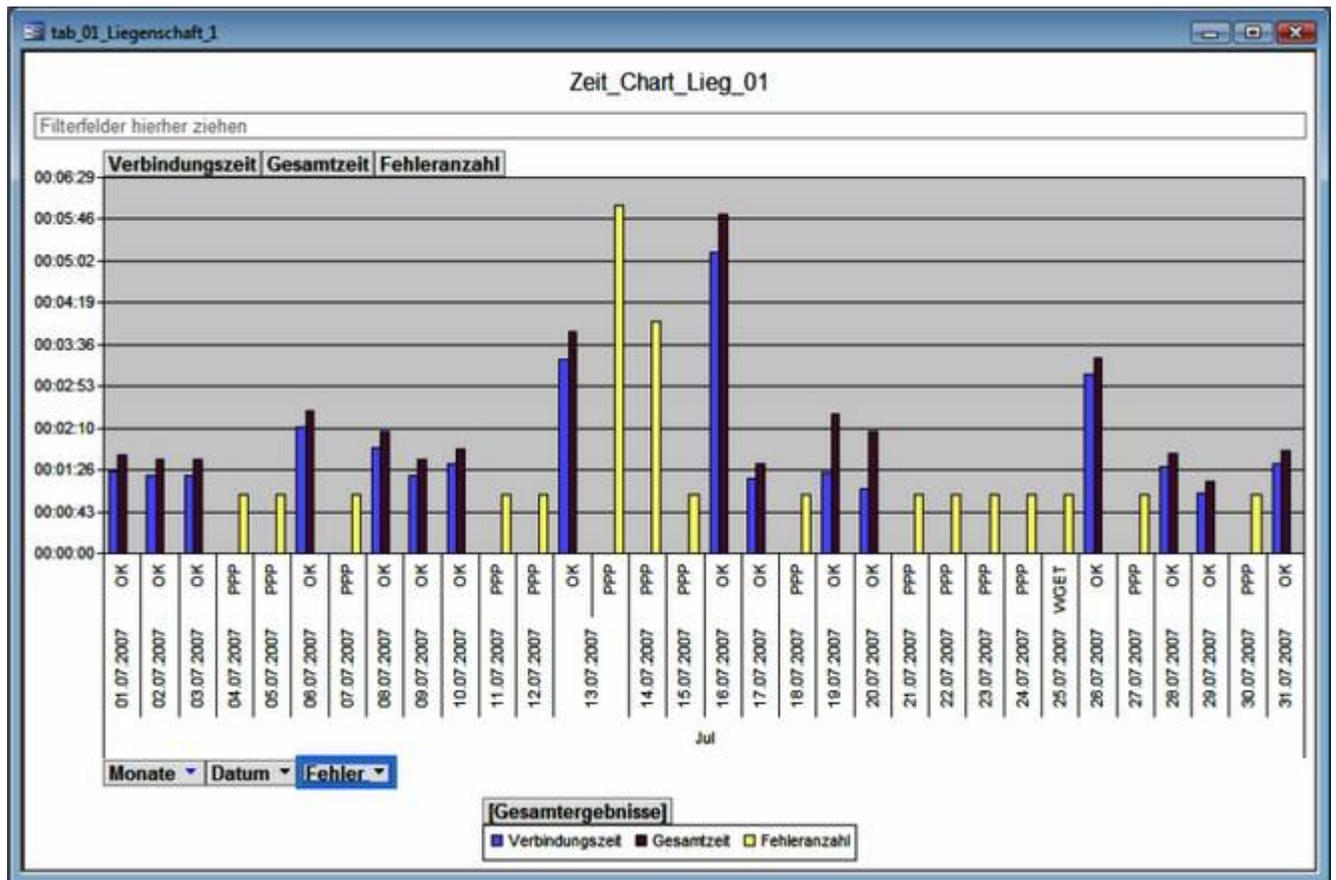


Abbildung 24: Zeit-Chart-Formular der Liegenschaft\_1 für den Monat Juli mit Fehlerart.

Beim Formular für Fehlerdiagramme repräsentiert ein Scheibendiagramm die Häufigkeitsverteilung der Fehler auf die gesamte zu betrachtende Periode. So lässt sich für jede Liegenschaft schnell herausfinden, welcher Art von Fehler wie oft in einem bestimmten Zeitraum aufgetreten ist, um darauf angemessen reagieren zu können.

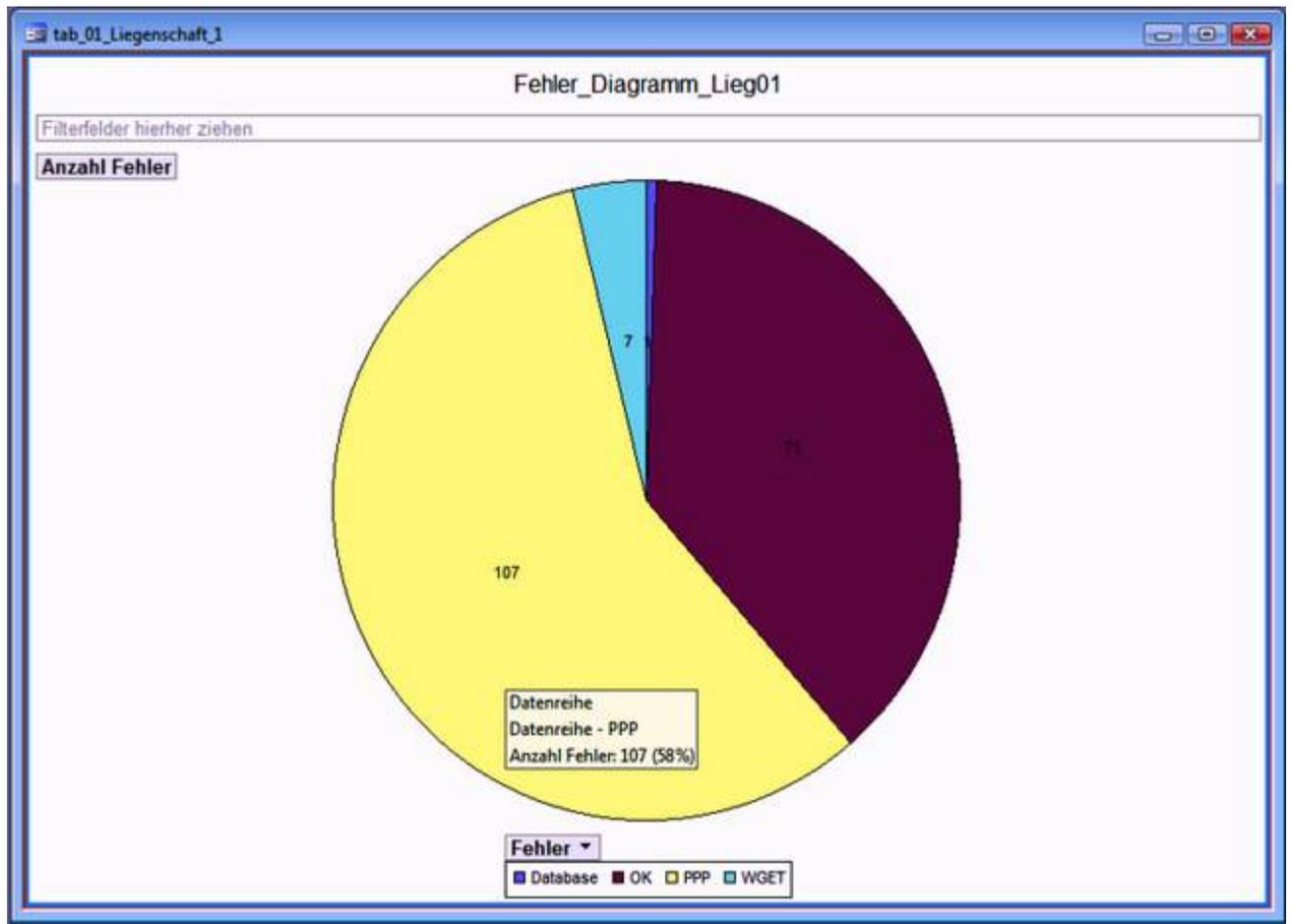


Abbildung 25: Fehler-Diagramm der Liegenschaft\_1.

Abbildung 25 zeigt als Beispiel die Fehleraufteilung der Liegenschaft\_1 für eine bestimmte Periode. In dieser Periode weist die Liegenschaft\_1 eine Häufigkeit von 107 mal (58%) den Fehler PPP, 7 mal (4%) den Fehler WGET, 1 mal (1%) den Fehler Database und nur 71 mal (38%) ist kein Fehler aufgetreten. Dies Ergebnis ist ein gutes Beispiel für eine schlecht funktionierende GSM Verbindung. Es fordert dringend zum Handeln auf. Hier hilft wahrscheinlich der Austausch des Modems.

Eine genauere Monatsaufteilung der aufgetretenen Fehler (Abbildung 26) lässt sich einfach durch das Verschieben der Monatsfelder von der Feldliste in die Mehrfachdiagrammfelder erreichen.

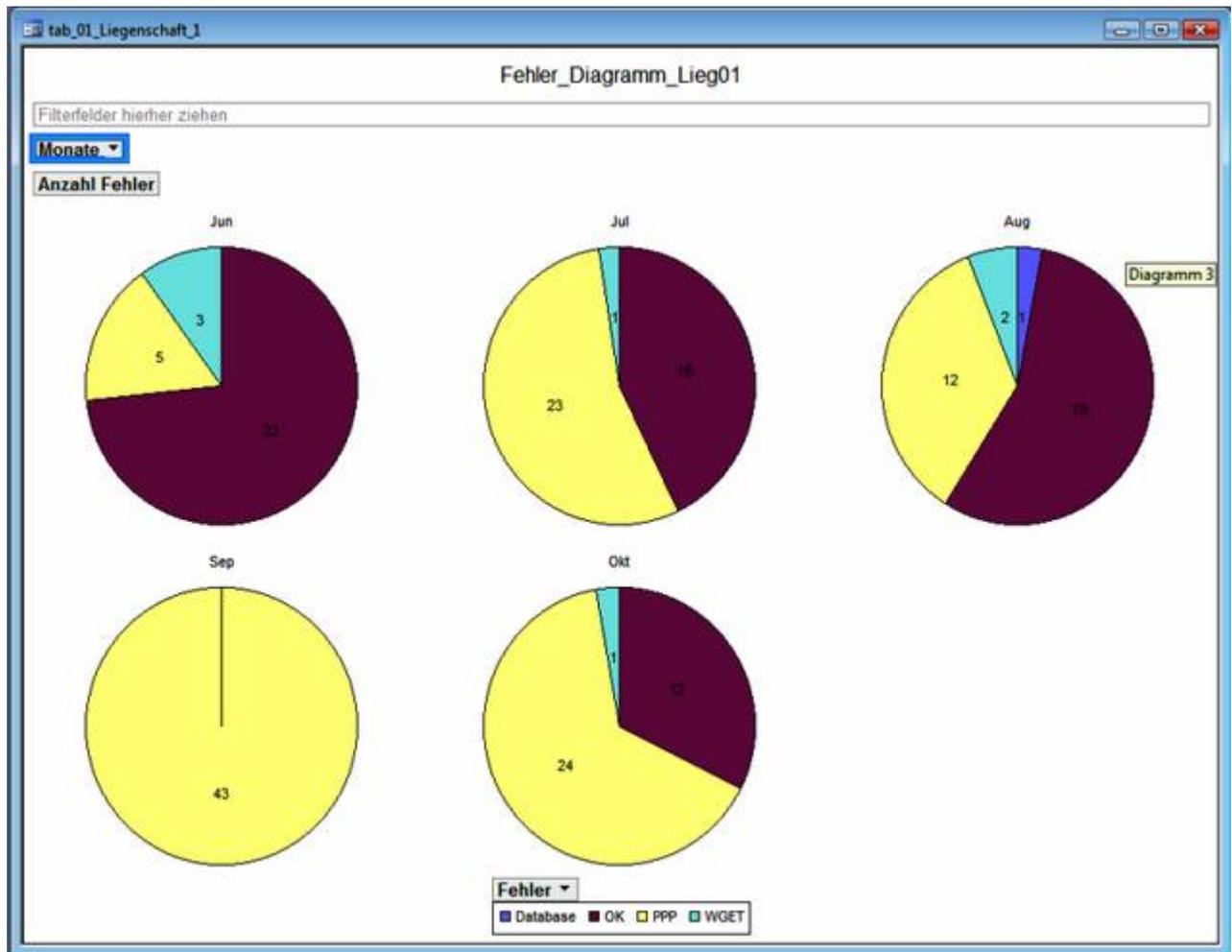


Abbildung 26: Fehler-Diagramm der Liegenschaft\_1 mit Monatsauflösung.

Für die Übersicht der gesamten Liegenschaften kommen zwei Pivot-Diagramme hinzu, die einmal die Verbindungszeit und die Gesamtzeit aller erfassten Liegenschaften nebeneinander, nach Liegenschaften sortiert in einem Pivot-Diagramm darstellen (Abbildung 27) sowie die Summe alle Verbindungszeiten und alle Gesamtzeiten in einem zweiten Pivot-Diagramm (Abbildung 28). Für die detaillierte Fehlerübersicht der gesamten Liegenschaften fungiert ein tabellarisches Formular als eine Alternative zu der sonst zu komplizierten Diagrammdarstellung aufgrund der verschiedenen Fehlerarten (Abbildung 29).



Abbildung 27: Zeit-Chart-Formular der gesamten Liegenschaften nach Liegenschaft aufgelöst.

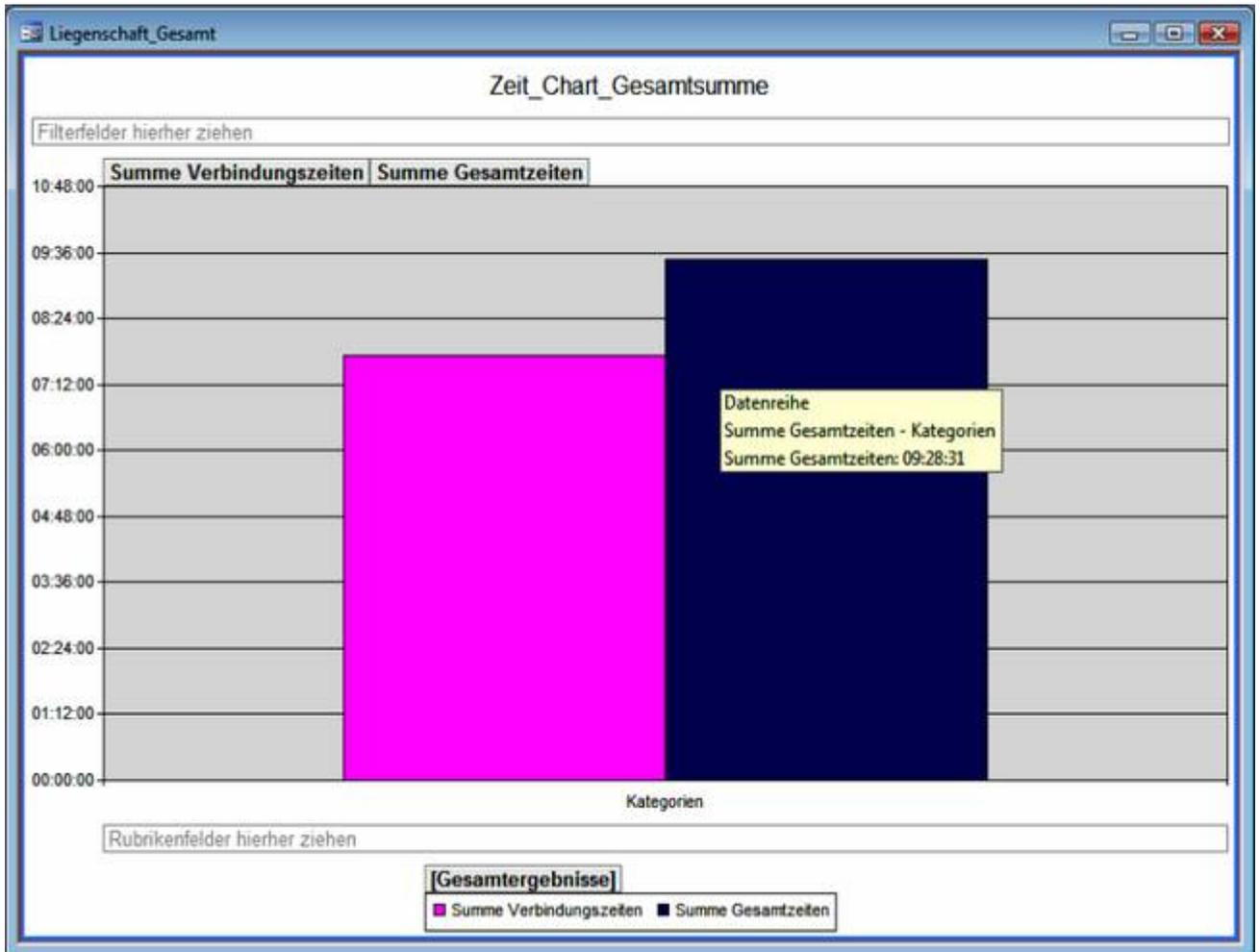


Abbildung 28: Zeit-Chart-Formular der gesamten Liegenschaften.

Liegenschaft	tab_01_Liegen	Anzahlvor	tab_02_Liegen	Anzahlvor	tab_03_Liegen	Anzahlvor	tab_04_Liegensc	Anzahlvor
Liegenschaft_1	Database	1		0		0		0
Liegenschaft_1	OK	71		0		0		0
Liegenschaft_1	PPP	107		0		0		0
Liegenschaft_1	WGET	7		0		0		0
Liegenschaft_2		0	Database	1		0		0
Liegenschaft_2		0	OK	59		0		0
Liegenschaft_2		0	PPP	105		0		0
Liegenschaft_2		0	WGET	6		0		0
Liegenschaft_3		0		0	Database	1		0
Liegenschaft_3		0		0	OK	78		0
Liegenschaft_3		0		0	PPP	86		0
Liegenschaft_3		0		0	WGET	6		0
Liegenschaft_4		0		0		0	00:00:00	1
Liegenschaft_4		0		0		0	Database	1
Liegenschaft_4		0		0		0	OK	93
Liegenschaft_4		0		0		0	PPP	86
Liegenschaft_4		0		0		0	WGET	5
Liegenschaft_5		0		0		0		0

Datensatz: 1 von 18

Abbildung 29: Tabellarische Formulare für die gesamt Fehlerübersicht.

Ein weiteres und letzteres Formular ist das in Abbildung 30 dargestellte Hauptfensterformular, welches beim Start der Datenbank erscheint und vor allem die Möglichkeit bietet, die Datenbank zu aktualisieren und zu sichern.

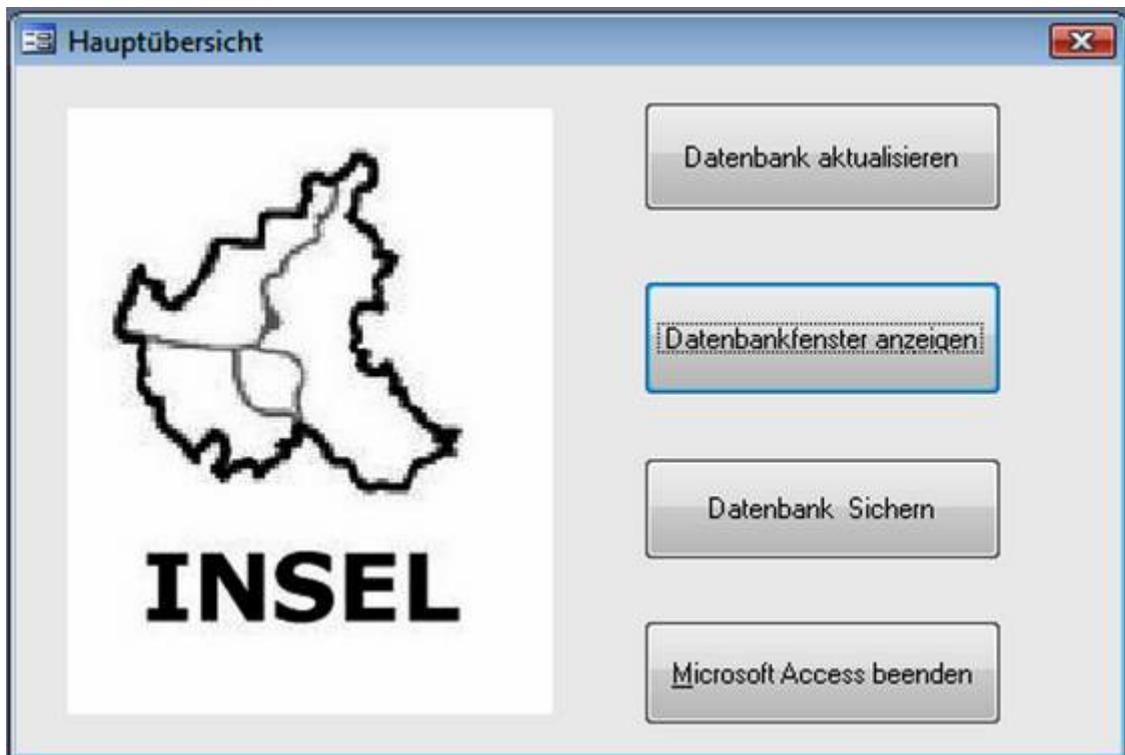


Abbildung 30: Hauptfenster der Datenbank Insel.

#### 5.2.4.5 Module (VBA)

Access bietet für spezielle Aufgaben, die sich sonst mit den interaktiven Möglichkeiten von Access nicht einfach realisieren lassen oder um bestimmte Aktionen zu automatisieren, einen Platz für programmierbare Lösungen. Die Module stellen einen Editor dar, wo kleine Funktionen, spezielle Berechnungen, automatische Abläufe etc. programmiert werden können. Das Programmieren erfolgt mit der in Access integrierten Programmiersprache VBA (Visual Basic for Applications).

In der Datenbank Insel ist der programmierte Abschnitt stark mit dem Hauptfensterformular verbunden, erst die in Module programmierten Prozeduren füllen die Bedienelemente des Hauptfensterformulars mit Leben. Beim Anklicken eines der Steuerelemente wird eine Ereignisprozedur ausgelöst, die einen bestimmten Programmabschnitt ausführt. So wird beim Anklicken auf den *Microsoft Access beenden* Button der Programmabschnitt „MicrosoftAccessBeenden“ durchgeführt, wo die *DoCmd-Methode Quit* dafür sorgt, das Access Programm zu beenden.

```
' Microsoft Access beenden.
  DoCmd.Quit
```

*DoCmd* ist ein Application-Objekt mit dem man verschiedene Access Aktionen wie Öffnen und Schließen von Formularen durchführen kann.

Ähnlich läuft es beim Betätigen des *Datenbankfenster anzeigen* Buttons. Hier sorgt der Programmabschnitt „DatenbankfensterAnzeigen“ über die *DoCmd-Methode* dafür, dass sich das Hauptfensterformular minimiert und die Datenbankfenster wieder erscheinen.

```
Dim strDocName As String

strDocName = "Hauptfenster"

' Formular "Hauptübersicht" schließen.
DoCmd.Minimize

' Focus auf Datenbankfenster setzen; Formular "Hauptfenster" auswählen.
DoCmd.SelectObject acForm, strDocName, True
```

Etwas komplizierter dagegen läuft es bei den zwei übrigen Buttons ab, beim *Datenbank aktualisieren* Button handelt sich um die Aktualisierung der Liegenschaftstabellen, die im Grunde genommen Kopien der Verknüpfungstabellen anfertigt, um mit den aktuellen Daten arbeiten zu können. In dem dazu gehörigen Programmabschnitt wird als erstes der Name jeder Tabelle aus den Logfiles Ordner abgeleitet, anschließend über einen „SQL SELEKT“ Befehl die für die weitere Betrachtung wichtigen Felder aus der Verknüpfungstabelle selektiert,

```
For Each Datei In objOrdner.Files      ' Schleife zum durchlaufen der Ordner

    intPos = InStrRev(Datei.Name, ".") ' Name der
    strVerk = Left(Datei.Name, intPos - 1) ' Tabellen aus
    strID = Left(Datei.Name, 2)          ' den Logfile Namen
    strTable = "tab_" & strVerk         ' zusammensetzen

    ' erstellen eine neue Tabelle aus den selektierten Feldern
    strSQL1 = "SELECT Lieg_ID, Datum, Status_OK, StartZeit, VerbindungsZeit,
              GesamtZeit, Fehler," & _
              " FehlerAnzahl, FehlerID INTO " & strTable & " FROM " & strVerk
```

danach werden durch ein weiteres „SQL ALTER“ Befehl die für die Abfragen wichtigen Primärschlüssel erzeugt.

*' erstellund der Primärschlüssel Datum & Startzeit*

```
strSQL2 = "ALTER TABLE " & strTable & " ADD CONSTRAINT DAT PRIMARY  
KEY (Datum, StartZeit)"
```

Abschließend sorgt ein dritter „SQL UPDATE“ Befehl dafür, dass jede Tabelle die richtige Liegenschafts\_ID bekommt statt einer provisorischen.

*' aktualisiert die liegenschaft\_ID*

```
strSQL3 = "UPDATE " & strTable & " SET " & strTable & ".Lieg_ID = " & strID
```

Beim letzten verbliebenen Button *Datenbank sichern* wird durch das File-System-Objekt dafür gesorgt, dass eine Sicherungskopie von der Datenbank Insel gemacht wird, um gewährleisten zu können, dass mit der letzten funktionierenden Datenbankzustand jeder Zeit gearbeitet werden kann, auch wenn die Verbindung zu den Originaldaten fehlgeschlagen ist.

```
Current = CurrentProject.FullName
```

*' Name der Sicherungsdatei erstellen*

```
Zieldatei = Left(Current, Len(Current) - 4) & "_" & Year(Now) & ".mdb"
```

```
Set oFSO = CreateObject("Scripting.FileSystemObject")
```

```
Quelldatei = Current
```

```
oFSO.CopyFile Quelldatei, Zieldatei, True
```

*' Sicherung erstellen*

File-System-Objekt ist eine Library, die den Zugriff auf Dateien bzw. auf das Dateisystem ermöglicht.

Es empfiehlt sich für den richtigen Gebrauch der Datenbank Insel, die Datenbank jedes Mal zunächst zu aktualisieren, bevor man mit dieser Datenbank arbeitet und sie auch immer zu sichern, bevor man es verlässt.

### **5.3 Analyse der Ergebnisdaten des Auswertungstools**

Aus den Auswertungsergebnissen lässt sich bei Problemen in der Zählerauslesung durch die Fehlerart, die Ursache analysieren. So lässt eine große Häufigkeit von Database-Fehlern darauf schließen, dass wiederholt Probleme bei der Speicherung der Lastdaten auftreten, was eine Intervention in der Oracle-Datenbank erfordert.

Wenn häufig WGET-Fehler auftreten, die das nicht erfolgreiche Herunterladen der Lastdaten anzeigen, muss eventuell darüber nachgedacht werden, ob der Installationsplatz des Auslesegeräts an einen anderen Platz verlegt werden sollte, wo der Empfang besser ist. Oder ob statt einer GSM-Verbindung auf eine Festnetz-Verbindung umgestellt werden kann, um ein Abbrechen der Verbindung während des Herunterladens zu verhindern.

Das Problem schlechter Verbindungsqualität beziehungsweise Signalstärke ist kein seltenes Problem bei den Liegenschaften. Die meisten Stromzähler befinden sich in den hinteren Gebäudeteilen oder im Keller, wo die Signalstärke des GSM-Funknetzes schwach oder manchmal gar nicht vorhanden ist.

Bei einer großen Häufigkeit der PPP-Fehler ist das Problem in der Regel schneller zu beheben, da es ein lokales Problem ist, das sich oft durch Ein- und Ausschalten beziehungsweise Auswechseln des Modems beheben lässt.

Aus den Auswertungsergebnissen lässt sich auch durch Betrachtung der Verbindungszeiten, die Notwendigkeit einer Flatrate für die Kommunikation mit den Datenloggern besser beurteilen.

## 6 Zusammenfassung und Ausblick

### 6.1 Zusammenfassung

Das Sammeln der einzelnen Liegenschaftslastdaten spielt eine Hauptrolle bei der Realisierung des Forschungsprojekts Insel. Für die Verfügbarkeitssicherung der Lastdaten hat daher insbesondere auch das Qualitätsmanagement bei der Zählerauslesung eine entscheidende Bedeutung

im Rahmen dieser Arbeit wurden daher verschiedene Aspekte betrachtet, die für ein Qualitätsmanagement bei der Zählerauslesung im Rahmen dieses Projekts wichtig sind.

Zunächst wurde anhand der Netzwerkmanagement-Standards nach [ISO 7498-4] der theoretische Aspekt des Qualitätsmanagements dargestellt und dessen Funktionsbereiche erläutert. Danach wurden Lösungen zum Qualitätsmanagement entwickelt, zuerst für die Zählerauslesung und anschließend für das gesamte Netzwerk (GSM + LAN) des Insel-Projekts.

So wurde zum Beispiel für den Funktionsbereich „Konfigurationsmanagement“ eine (Excel-) Liegenschaftsdokumentation erstellt. Darin wurde eine ausführliche Informationssammlung durchgeführt, die alle mit der Zählerauslesung verbundenen Angaben dokumentiert. Zum einen die direkten Daten über das Auslesegerät beziehungsweise den Stromzähler selbst sowie indirekte Daten wie Informationen über die Liegenschaften und die Ansprechpartner bei Vattenfall. Die gesammelten Informationen sind notwendig zur Fehlerbehebung eventuell auftretender Probleme, die durch die eingesetzten Tools festgestellt wurden.

Für die Qualitätssicherung des Insel-Projekts-Netzwerks und somit auch die Sicherung der Verfügbarkeit der Lastdaten, wurden für den LAN-Bereich des Netzwerks geeignete Produkte präsentiert und auf deren Vorteile und Einsatzgebiete hingewiesen.

Für den GSM-Bereich des Netzwerks, für den auf dem Markt bisher keine Lösungen existieren, wurde mit Hilfe der Analyse des Ereignisprotokolls (von der Firma ENVIDATEC zur Verfügung gestellt) ein Qualitätsmanagementtool entwickelt. Dieses Tool besteht aus zwei sich komplementierenden Tools. Im ersten, dem Vorbereitungstool, werden die wichtigen Informationen aus den textuellen originalen Logfiles durch ein JAVA-Programm herausgefiltert und in tabellarischer Form in CSV-Dateien gespeichert. So können sie mit dem zweiten in ACCESS erstellten Datenbanktool aggregiert und in grafischer Form ausgewertet und analysiert werden.

## **6.2 Ausblick**

Aufgrund der fehlenden Projekt-Testdaten war es nicht möglich, das Auswertungstool intensiv zu testen und echte Projektdaten auswerten zu können. Es ist daher anzunehmen, dass sich in der Praxiserprobung noch Verbesserungen in der Voreinstellung der hier entwickelten Analysetools ergeben. Außerdem muss das entwickelte Tool zukünftig an andere Zählerauslesegeräte (mit anderen Verbindungsprotokollen) angepasst werden, falls solche im Laufe des Projekts verwendet werden sollten. Darüber hinaus wäre es sicherlich von Vorteil, das aus zwei Teilen erstellte Qualitätsmanagementtool in eines zu fassen, das die Daten filtert, konvertiert und auswertet.

## 7 Anhang

### 7.1 Literaturverzeichnis

- [1] Bundesministerium für Umwelt, Naturschutz und Reaktorsicherheit/  
Erneuerbare Energien in Zahlen – nationale und internationale Entwicklung.  
Stand: November 2007.
- [2] Einsatzmöglichkeiten Regenerativer Energien für eine klimaverträgliche  
Elektrizitätsversorgung in Deutschland,  
<http://www.volker-quaschning.de/downloads/index.html>  
Einsichtnahme: 08.04.2008
- [3] Projektbeschreibung: Insel Internetbasiertes System eines erweiterbaren  
Lastmanagements zur Integration in virtuelle Kraftwerke.
- [4] E Forum,  
Magazin des Department Informations- und Elektrotechnik  
Jahrgang 2007
- [5] Kennzahlen und Qualitätsmanagement, Best Practices im IT Service  
Management,  
<http://www.presetext.at/pte.mc?pte=040415037>  
Einsichtnahme: 12.03.2008
- [6] Karl Ewald,  
Diplomarbeit, Entwicklung einer Methodik zur systematischen Gewinnung  
generischer Konfigurationsvorgänge am Beispiel ausgewählter LAN-  
Komponenten, 1995 Institut für Informatik München.
- [7] ISO7498-4,  
ISO Internationale Standardisierungsorganisation (Hrsg.): 7498-4, Information  
Processing Systems - OSI Basic Reference Model - Part 4: Management  
Framework
- [8] Heinz-Gerd Hegering, Sebastian Abeck, Bernhar Neumair  
Integriertes Management - vernetzter Systeme – Konzepte, Architekturen und  
deren betrieblicher Einsatz. (Hrsg.: dpunkt Verlag), 1 Auflage 1999, Heidelberg
- [9] Configuration management,  
[http://www.itwissen.info/definition/lexikon//\\_CMCM\\_CMconfiguration%20mana  
gementCM\\_CMKonfigurationsmanagement.html](http://www.itwissen.info/definition/lexikon//_CMCM_CMconfiguration%20managementCM_CMKonfigurationsmanagement.html)  
Einsichtnahme: 13.03.2008
- [10] Fault management,  
[http://www.itwissen.info/definition/lexikon//\\_FMFM\\_FMfault%20managementF  
M\\_FM Fehlermanagement.html](http://www.itwissen.info/definition/lexikon//_FMFM_FMfault%20managementFM_FM Fehlermanagement.html)  
Einsichtnahme: 14.03.2008

- [11] Performance management,  
[http://www.itwissen.info/definition/lexikon//\\_PMPM\\_PMperformance%20managementPM\\_PMLleistungsmanagement.html](http://www.itwissen.info/definition/lexikon//_PMPM_PMperformance%20managementPM_PMLleistungsmanagement.html)  
Einsichtnahme: 14.03.2008
- [12] Security management,  
[http://www.itwissen.info/definition/lexikon//\\_SMSM\\_SMsecurity%20managementSM\\_SMSicherheitsmanagement.html](http://www.itwissen.info/definition/lexikon//_SMSM_SMsecurity%20managementSM_SMSicherheitsmanagement.html)  
Einsichtnahme: 14.03.2008
- [13] Accounting management,  
[http://www.itwissen.info/definition/lexikon//\\_AMAM\\_AMaccounting%20managementAM\\_AMAbrechnungsmanagement.html](http://www.itwissen.info/definition/lexikon//_AMAM_AMaccounting%20managementAM_AMAbrechnungsmanagement.html)  
Einsichtnahme: 14.03.2008
- [14] OSI-Management,  
[http://www.itwissen.info/definition/lexikon//\\_OSI%20management\\_OSI-Management.html](http://www.itwissen.info/definition/lexikon//_OSI%20management_OSI-Management.html)  
Einsichtnahme: 13.03.2008
- [15] McConnell J,  
Managing Client-Server Environment: Tools and Strategies for Building Solution. (Hrsg.: Prentice-Hall Verlag), 1996
- [16] Terplan K,  
Client/Server Management, (Hrsg.: Datacom Buchverlag) 1995
- [17] Erwin Salm  
IT-Systemmanagement / Infrastruktur-Management, (Hrsg.: Diplomica Verlag),  
1 Auflage 2007, Hamburg
- [18] Free Network Monitoring Software for Small Networks,  
<http://manageengine.adventnet.com/products/opmanager/Network-Monitoring-Software.pdf>  
Einsichtnahme: 15.03.2008
- [19] Nagios,  
<http://www.nagios.org/about/>  
Einsichtnahme: 15.03.2008
- [20] Nagios,  
<http://de.wikipedia.org/wiki/Nagios>  
Einsichtnahme: 15.03.2008
- [21] OpManager, Product-verview,  
<http://www.manageengine.de/shop/media/file/products/OpManager/product-overview.pdf>  
Einsichtnahme: 15.03.2008

- [22] PPP,  
<http://www.birds-eye.net/definition/acronym.cgi?what+is+PPP=Point+to+Point+Protocol&id=1165939478>  
Einsichtnahme: 28.03.2008
- [23] PPP, (2.4.3-20050321+2sarge1)  
<http://packages.debian.org/sarge/ppp>  
Einsichtnahme: 28.03.2008
- [24] PPP, source-code  
<http://packages.debian.org/source/sarge/ppp>  
Einsichtnahme: 28.03.2008
- [25] WGET, (1.10.2-2)  
<http://packages.debian.org/stable/web/wget>  
Einsichtnahme: 28.03.2008
- [26] WGET,  
<http://de.wikipedia.org/wiki/Wget>  
Einsichtnahme: 28.03.2008
- [27] WGET, source-code  
<http://packages.debian.org/source/etch/wget>  
Einsichtnahme: 29.03.2008
- [28] JAVA, Strings  
[http://www.gailer-net.de/tutorials/java/Notes/chap26/ch26\\_20.html#literal,%20string](http://www.gailer-net.de/tutorials/java/Notes/chap26/ch26_20.html#literal,%20string)  
Einsichtnahme: 04.04.2008
- [29] JAVA, Besonderheiten  
[http://www.lorrek.de/st2/st2\\_03.htm](http://www.lorrek.de/st2/st2_03.htm)  
Einsichtnahme: 04.04.2008
- [30] Guido Krüger  
Handbuch der Java-Programmierung, (Hrsg.: Addison-Wesley), 4 Auflage  
2004, München
- [31] Christian Ullenboom  
Java ist auch eine Insel, (Hrsg.: Galileo Computing), 7 Auflage 2007
- [32] Access 2003 für Einsteiger,  
[http://www.hrz.uni-dortmund.de/S1/download/skripte/s3/aces/access2003\\_einfuehrung.pdf](http://www.hrz.uni-dortmund.de/S1/download/skripte/s3/aces/access2003_einfuehrung.pdf)  
Einsichtnahme: 20.03.2008
- [33] Relationale Datenbanken  
<https://www-rz.uni-hohenheim.de/mysql/kursunterlagen/tutorials/Relationale-Datenbanken.pdf>

Einsichtnahme: 20.03.2008

- [34] Codd E. F,  
The Relational Model for Database Management, (Hrsg.: Addison-Wesley Verlag), 2 Auflage 1990
  
- [35] Gerhard Brosius,  
Access 2002 professionell, (Hrsg.: Addison-Wesley Verlag), Auflage 2002
  
- [36] Said Baloui,  
Access 2000-Kompendium, (Hrsg.: Markt und Technik Verlag), 1 Auflage, 1999
  
- [37] Ralf Albrecht, Natascha Nicol,  
Access 2002 programmieren, (Hrsg.: Addison-Wesley Verlag), Auflage 2002
  
- [38] Said Baloui,  
Jetzt lerne ich VBA mit Access, (Hrsg.: Markt und Technik Verlag), Auflage 2004
  
- [39] Bernd Held,  
Das Access-VBA Codebook, (Hrsg.: Addison-Wesley Verlag), Auflage 2007

## 7.2 *Abbildungsverzeichnis*

Abbildung 1: Die Entwicklung der Stromerzeugung aus erneuerbaren Energien 1990 bis 2006 [1].....	5
Abbildung 2: Systeminfrastruktur des Projekts Insel[3].....	7
Abbildung 3: Die fünf Managementfunktionsbereiche in Abhängigkeit von einander[14].....	11
Abbildung 4: Screenshot_1 aus dem Netzwerkmanagementprodukt Nagios [19]. ....	15
Abbildung 5: Screenshot_2 aus dem Netzwerkmanagementprodukt Nagios [19]. ....	16
Abbildung 6: Screenshot_1 aus dem Netzwerkmanagementprodukt OpManager [21]. .....	17
Abbildung 7: Screenshot_2 aus dem Netzwerkmanagementprodukt OpManager [21]. .....	17
Abbildung 8: Struktur des Insel-Projekt-Netzwerkes.....	19
Abbildung 9: Ablaufdiagramm_1 des JAVA Programms.....	39
Abbildung 10: Ablaufdiagramm_2 des JAVA Programms.....	40
Abbildung 11: Ablaufdiagramm_3 des JAVA Programms.....	41
Abbildung 12: Ablaufdiagramm_4 des JAVA Programms.....	42
Abbildung 13: Ablaufdiagramm des Unterprogramms LastDate.....	43
Abbildung 14: Tabelle mit Mehr-Felder-Primärschlüssel in Access. ....	51
Abbildung 16: Tabellenverknüpfungs-Managers der Access Datenbank.....	54
Abbildung 17: Auswahlabfrage GESAMT der Datenbank Insel. ....	57
Abbildung 18: Verknüpfungseinstellungen zwischen Liegenschaft_1 und Liegenschaft_2. ....	58
Abbildung 19: Leeres Pivot-Table-Formular der Datenbank Insel. ....	60
Abbildung 20: Pivot-Table-Formular der Datenbank Insel. ....	60
Abbildung 21: Pivot-Table-Formular der Datenbank Insel für Verbindungserfolgreiche Versuche. ....	61
Abbildung 22: Zeit-Chart-Formular der Liegenschaft_1.....	62
Abbildung 23: Zeit-Chart-Formular der Liegenschaft_1 für den Monat Juli.....	63
Abbildung 24: Zeit-Chart-Formular der Liegenschaft_1 für den Monat Juli mit Fehlerart. ....	64
Abbildung 25: Fehler-Diagramm der Liegenschaft_1. ....	65
Abbildung 26: Fehler-Diagramm der Liegenschaft_1 mit Monatsauflösung.....	66
Abbildung 27: Zeit-Chart-Formular der gesamten Liegenschaften nach Liegenschaft aufgelöst. ....	67
Abbildung 28: Zeit-Chart-Formular der gesamten Liegenschaften. ....	68
Abbildung 29: Tabellarische Formulare für die gesamt Fehlerübersicht.....	68
Abbildung 30: Hauptfenster der Datenbank Insel. ....	69

### **7.3 Tabellenverzeichnis**

Tabelle 1: Der erste Teil der BSH Arbeitstabelle mit Bilder. ....	22
Tabelle 2: Der zweite Teil der BSH Arbeitstabelle mit den wichtigen Daten.....	23
Tabelle 3: Der dritte Teil der BSH Arbeitstabelle. ....	24
Tabelle 4: Übersichtstabelle in der gruppierten Form. ....	24
Tabelle 5: Die Antworten auf die Was? Wo? Warum? Fragen. ....	28
Tabelle 6: Tabellarisierte Form der Informationen. ....	47
Tabelle 7: Tabelle vor dem Verbund.....	48
Tabelle 8: Tabelle nach dem Verbund.....	48
Tabelle 9: Liegenschaftstabelle mit mehreren Verbindungsversuchen an einem Tag. .....	50
Tabelle 10: Verbindungstabelle der Liegenschaft_1.....	56
Tabelle 11: Erstellte Tabelle der Liegenschaft_1 aus der Verknüpfungstabelle. ....	56



```

        //erste Zeile Flag setzen
        ersteZeile = 1;
    }
    //speichert den letzten nach der formatierung
    dt = sdf2.parse(lastDate);

    //Datei schließen
    fr.close();

    //+++++//
} catch (Exception e) {
    fängt mögliche Fehlern vom readLine()
    System.out.println("Fehler: konnte Datei nicht finden!");

    e.printStackTrace();           //Fehlerbehandlung//
} //+++++//

//gibt das letzte datum im Ziel-Datei zurück
return dt;
}

//-----< main >-----//

public static void main (String[] args) {

    try{
        sourceFile = args[0];
        targetFile = args[1];

    } catch (Exception ex){
        System.out.println(ex);
    }
    InselFilter f = new InselFilter();
    f.readMyFile(sourceFile,targetFile);
}

//-----< readMyFile >-----//

void readMyFile(String filename,String zielDatei) {

    String record = null,
    startZeit    = null,
    zwischenZeit = null,
    endZeit      = null,
    zeitMarke    = null;           //hilfsvariable
    String datum = "1988-11-11",  //default Datum
    fehler       = "OK",
    tRecords     = "0",
    iRecords     = "0",
    fRecords     = "0",
    status       = "Nein",
    fehlerID     = "0";
    long gDauer  = 0, vDauer      = 0;
    int  gDauer_m = 0, gDauer_s  = 0;
    int  vDauer_m = 0, vDauer_s  = 0;

```

```

try {
    //Leser-Datei Objekt erzeugen
    FileReader fr = new FileReader(filename);

    //Zwischenspeicher Objekt erzeugen
    BufferedReader br = new BufferedReader(fr);

    //schreib-Datei Objekt erzeugen
    FileWriter fw = new FileWriter(zielDatei, true);

    //Objekt zur Zeilenorientierte Ausgabe erzeugen
    PrintWriter pw = new PrintWriter(fw);

    //Datum-formattierungsobjekt erzeugen
    SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");

    //Zeit-formattierungsobjekt erzeugen
    SimpleDateFormat sdf2 = new SimpleDateFormat("yyyy-MM-
dd");

    //String-recorderobjekt erzeugen
    record = new String();

    //String Feld-recorder erzeugen
    String[] record2 = null;

    //Datum Objekt erzeugen
    Date dt = new Date();

    //Aufruf der Funktion, die das letzte geloggte Datum zurück
gibt
    dt = lastDate("zielFile.csv");

    /*System.out.println("Datum;Status;Startzeit;Zwischenzeit;Endzeit;" +
    "Fehler;FehlerID;T.records;I.records;F.records");*/

    if(ersteZeile !=1)

        //shreibt die Feldnamen in der ersten Zeile im Datei
        pw.println("Lieg_ID;Datum;Status_OK;StartZeit;VerbindungsZeit;G
esamtZeit;"
        +"Fehler;FehlerAnzahl;FehlerID;TRecords;IRecords;FRecords");

    /*****
    Hier wird jeder zeile im Logfie eingelesen. in jeder Zeile die den
    Begriff "ADF session" enthält wird das Datum und die Startzeit
    festgehalten. Steht anstelle des Datums oder die Startzeit etwas
    anders im Logfile, wird Stattdessen einen default Datum bzw. eine
    default Uhrzeit eingesetzt
    *****/

    //liest jede Zeile im logfile solange das Ende der Datei nicht
    erreicht ist
    while (( record = br.readLine()) != null) {

```

```

//testet die eingelesene Zeile auf das gesuchte begriff
if( record.contains("ADF session" )){

    //splittet die ausgewählten Zeilen nach jeder leer
    Zeichen
    record2 = record.split(" ");

    //speichert das Datum aus der ausgewählten Zeile
    datum = record2[0];

    try{//+++++
        Date test = new Date();

    //testet ob das Datum diese Format "yyyy-mm-dd" hat.
        test = sdf2.parse(datum);

        }catch (ParseException e){ //Fehlerbehandlung
            datum = "1888-11-11";

        }//+++++

    //speichert die Startzeit aus der ausgewählten Zeile
    startZeit = record2[1];

    try{//+++++
        Date test = new Date();

    //testet ob die Startzeit diese Format "hh:mm:ss" hat.
        test = sdf.parse(startZeit);

        }catch (ParseException e){ //Fehlerbehandlung
            startZeit = "00:00:00";

        }//+++++
    }

/*****
    hier wird der "PPP" Fehler behandelt, dabei wird der Status_OK auf
    "Nein" und die Fehlerart auf "PPP" gesetzt. Der Fehler_ID wird aus
    der Fehlerzeile entnommen.
*****/

//testet ob das eingelesene Datum neu oder alt ist
if (dt.before(sdf2.parse(datum))){

    //testet die eingelesene Zeile auf das gesuchte
begriff

        if(record.contains("FATAL" )){

            fehler = "PPP";
            status = "Nein";

            //splittet die ausgewählten Zeilen nach
            jedes leeres Zeichen

```

```

        record2 = record.split(" ");

        //speichert die Fehler_ID bei PPP-Fehlern
        fehlerID = record2[record2.length-1];

        /*System.out.println(datum + " ;" + status + " ;" +
        startZeit + " ;" + 0 + " ;" + 0 + " ;" + fehler + " ;" +
        fehlerID);*/

pw.println("0 ;" +datum + " ;" + status + " ;" + startZeit + " ;" +
"00:00:00" + " ;" + "00:00:00" + " ;" + fehler +
" ;" + "00:01:00 ;" + fehlerID);

    }

/*****
    Hier wird der "Database" Fehler behandelt, dabei wird der Status_OK
    Auf "Nein", die Fehlerart auf "Database" und die Fehler_ID auf "0"
    gesetzt.
*****/

//testet die eingelesene Zeile auf das gesuchte begriff
    if(record.contains( "ERROR" ) && !record.contains( "unexpected"
)) {
        //testet ob die selbe Zeile den begriff "database"
        enthält
            if(record.contains("database")) {
                fehler = "Database";
                fehlerID = "0";
                status = "Nein";

                //betrachtet nur die erste Fehler beim schreiben in
                der
                Datenbank, Duplikate werden ausgeschlossen.
                if(zeitMarke != startZeit){

                    /*System.out.println(datum + " ;" + status + " ;"
                    + startZeit + " ;" + vDauer_m+":" +vDauer_s +
                    ;" + 0 + " ;" + fehler + " ;" + fehlerID + " ;" +
                    tRecords + " ;" + iRecords + " ;"+fRecords);*/

                    pw.println("0 ;" +datum + " ;" + status + " ;" +
                    startZeit + " ;"+"0:" +vDauer_m+":" +vDauer_s +
                    ;" + "00:00:00"+ " ;" + fehler + " ;" +
                    "00:01:00 ;" + fehlerID);
                }
                //merkt sich die Startzeit für die if Anweisung
                zeitMarke = startZeit;
            }

/*****
    Hier wird der "WGET" Fehler behandelt, dabei wird der
    Status_OK auf "Nein" und die Fehlerart auf "WGET" gesetzt.
    Aus der Fehlerzeile werden die Fehler_ID und die
    Verbindungszeit entnommen.
    Steht anstelle des Verbindungszeit etwas anders im Logfile
    wird Stattdessen eine default Uhrzeit eingesetzt.
*****/

```

```

        else{
            fehler = "WGET";
            status = "Nein";

            //splittet die ausgewählten Zeilen nach jeder leer
            Zeichen
            record2 = record.split(" ");

            //speichert die Fehler_ID bei WGET-Fehlern
            fehlerID = record2[record2.length-1];

            //speichert die Verbindungszeit bei WGET-Fehlern
            zwischenZeit = record2[1];

        }

    try{
        //+++++
        Date test = new Date();

        //testet ob die Startzeit diese Format "hh:mm:ss" hat.
        test = sdf.parse(startZeit);

        }catch (ParseException e){
        //Fehlerbehandlung
            zwichentZeit = "00:00:00";

        }//+++++

        vDauer = sdf.parse(zwischenZeit).getTime()-
sdf.parse(startZeit).getTime();
//Umrechnung der Verbindungsdauer
        vDauer_m = (int) vDauer/60000;           // in Minuten
        vDauer_s = (int) (vDauer%60000)/1000;   // und Sekunden

        /*System.out.println(datum + " ;" + status + " ;" + startZeit + " ;" + 0
        + " ;" + 0 + " ;" + fehler + " ;" + fehlerID);*/

        pw.println("0 ;" +datum + " ;" + status + " ;" + startZeit + " ;" +
"0:" +vDauer_m+" : " +vDauer_s + " ;" + "00:00:00" + " ;" + fehler +
" ;" + "00:01:00 ;" + fehlerID);

    }
}

/*****
Hier wird der fehlerfreie Teil behandelt, dabei wird an erste stelle
Die Verbindungszeit und die Gesamtzeit entnommen. Fehlern im Logfile
werden durch default Uhrzeiten behandelt.
an zweite Stelle werden die total-, iportierten- und die
fehlerhaften-records entnommen.

*****/

//testet die eingelesene Zeile auf das gesuchte begriff
if(record.contains( "Successfully")){

//splittet die ausgewählten Zeilen nach jeder leer Zeichen
record2 = record.split(" ");

//speichert die Verbindungszeit bei erfolgreicher Verbindungsaufbau

```

```

zwischenZeit = record2[1];

try{
//+++++
    Date test = new Date();

    //testet ob die Startzeit diese Format "hh:mm:ss" hat.
    test = sdf.parse(startZeit);

    }catch (ParseException e){
//Fehlerbehandlung
        zwichentZeit = "00:00:00";

    }//+++++

    vDauer = sdf.parse(zwischenZeit).getTime()-
sdf.parse(startZeit).getTime();
//Umrechnung der Verbindungsdauer
    vDauer_m = (int) vDauer/60000;           // in Minuten
    vDauer_s = (int) (vDauer%60000)/1000;   // und Sekunden

    //testet die eingelesene Zeile auf das gesuchte begriff
    if(record.contains( "total")){

    //splittet die ausgewählten Zeilen nach jeder leer Zeichen
    record2 = record.split(" ");

    //speichert die Anzahl der total Records
    tRecords = record2[2];
    }

    //testet die eingelesene Zeile auf das gesuchte begriff
    if(record.contains("records imported")){

    //splittet die ausgewählten Zeilen nach jeder leer Zeichen
    record2 = record.split(" ");

    //speichert die Anzahl der importierten Records
    iRecords = record2[2];
    }

    //testet die eingelesene Zeile auf das gesuchte begriff
    if(record.contains("records failed")){

    //splittet die ausgewählten Zeilen nach jeder leer Zeichen
    record2 = record.split(" ");

    //speichert die Anzahl der fehlgeschlagenen Records
    fRecords = record2[2];
    }

    //testet die eingelesene Zeile auf das gesuchte begriff
    if(record.contains("Conversion finished")){
        status ="Ja";

        //splittet die ausgewählten Zeilen nach jeder leer Zeichen

```

```

record2 = record.split(" ");

//speichert die Gesamtzeit bei erfolgreichem Vorgang
endZeit = record2[1];

    try{
/+++++
        Date test = new Date();

        //testet ob die Startzeit diese Format "hh:mm:ss" hat.
        test = sdf.parse(startZeit);

    }catch (ParseException e){
//Fehlerbehandlung
        endZeit = "00:00:00";

    }//+++++

        gDauer = sdf.parse(endZeit).getTime()-
sdf.parse(startZeit).getTime();

                                                    //Umrechnung der
Gesamtsdauer
        gDauer_m = (int) gDauer/60000;           // in Minuten
        gDauer_s = (int) (gDauer%60000)/1000;   // und Sekunden

        fehler = "OK";
        fehlerID = "0";
        //betrachtet nur die Gesamtzeiten bei erfolgreichem Vorgang
Gesamtzeiten
        bei Database-Fehlern werden ausgeschlossen.
        if(zeitMarke != startZeit){

                /*System.out.println(datum + " ;" + status + " ;" + startZeit + "
;"+
                vDauer_m+":" +vDauer_s + " ;"+gDauer_m+":" +gDauer_s + " ;" +
                fehler + " ;" + " ;"+ tRecords + " ;"+ iRecords + " ;"+fRecords);*/

                pw.println("0 ;" +datum + " ;" + status + " ;" + startZeit + " ;"+
                "0:"+vDauer_m+":" +vDauer_s + " ;" + "0:"+ gDauer_m+":" +gDauer_s
                + " ;" + fehler + " ;" + "00:00:00 ;"+ " ;"+ tRecords + " ;"+
                iRecords + " ;"+fRecords);
                }
                zeitMarke = startZeit;
        }
    }
}

//schließen der Ziel-Datei
pw.close();

//+++++
}catch (Exception e) {
    fängt mögliche Fehlern vom readLine()
    System.out.println("Fehler: konnte Datei nicht finden!");

    e.printStackTrace();
//Fehlerbehandlung//

```

```
    }catch (ParseException e){
        e.printStackTrace();
    }
    //++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++//
}
}
```

## 7.5 Source Code des VBA-ACCESS-Programms

```

VERSION 1.0 CLASS
BEGIN
    MultiUse = -1 'True
END
Attribute VB_Name = "Form_Hauptfenster"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False

Option Compare Database ' Die Sortierreihenfolge der Datenbank für
                        Zeichenfolgenvergleiche verwenden.
Option Explicit ' Erfordert, dass alle Variablen vor ihrer Verwendung
                deklariert werden.
-----

Private Sub Aktualisierung_Click()

Dim b As Boolean

    b = OrdnerDurchsuchen("C:\Users\Meins\Insel")

End Sub
-----

Sub DatenbankfensterAnzeigen_Click()
' Dieser Code wurde teilweise vom Befehlsschaltflächen-Assistenten
erstellt.
On Error GoTo Err_DatenbankfensterAnzeigen_Click

    Dim strDocName As String

    strDocName = "Hauptfenster"

    ' Formular "Hauptübersicht" schließen.
    DoCmd.Minimize

    ' Focus auf Datenbankfenster setzen; Formular "Hauptfenster" auswählen.
    DoCmd.SelectObject acForm, strDocName, True

Exit_DatenbankfensterAnzeigen_Click:
    Exit Sub

Err_DatenbankfensterAnzeigen_Click:
    MsgBox Err.Description
    Resume Exit_DatenbankfensterAnzeigen_Click

End Sub
-----

Private Sub Sicherung_Click()

DB_Sicher

End Sub
-----

Sub MicrosoftAccessBeenden_Click()
' Dieser Code wurde vom Befehlsschaltflächen-Assistenten erstellt.

```

```

On Error GoTo Err_MicrosoftAccessBeenden_Click

    ' Microsoft Access beenden.
    DoCmd.Quit

Exit_MicrosoftAccessBeenden_Click:
    Exit Sub

Err_MicrosoftAccessBeenden_Click:
    MsgBox Err.Description
    Resume Exit_MicrosoftAccessBeenden_Click

End Sub
-----

-----

Attribute VB_Name = "Modul1"

Option Compare Database
-----

Function OrdnerDurchsuchen(strV As String) As Boolean

    Dim fs As Object
    Dim objOrdner As Object
    Dim Datei As Object
    Dim intz As Integer
    Dim intPos As Integer
    Dim strVerk As String
    Dim strTble As String
    Dim strID As String
    Dim strSQL1 As String
    Dim strSQL2 As String
    Dim strSQL3 As String

On Error GoTo fehler
    If CurrentProject.Name <> "Insel.mdb" Then
        MsgBox "Die Aktualisierung kann nicht auf eine Kopie" & _
            " angewendet werden."
    Exit Function
    End If

    Set fs = CreateObject("Scripting.FileSystemObject")
    Set objOrdner = fs.GetFolder(strV)

    For Each Datei In objOrdner.Files
        ' Schleife zum
durchlaufen
        ' Name der
        intPos = InStrRev(Datei.Name, ".")
        strVerk = Left(Datei.Name, intPos - 1)
        strID = Left(Datei.Name, 2)
        ' Tabellen aus
        ' den Logfile
        Namen
        strTable = "tab_" & strVerk
        '
zusammensetzen

        'erstellen eine neue Tabelle aus den selektierten Feldern

```

```

        strSQL1 = "SELECT Lieg_ID, Datum, Status_OK, StartZeit,
VerbindungsZeit, GesamtZeit, Fehler," & _
                " FehlerAnzahl, FehlerID INTO " & strTable & " FROM " &
strVerk

        ' erstellund der Primärschlüssel Datum & Startzeit
        strSQL2 = "ALTER TABLE " & strTable & " ADD CONSTRAINT DAT PRIMARY
KEY (Datum, StartZeit)"

        ' aktualisiert die liegenschaft_ID
        strSQL3 = "UPDATE " & strTable & " SET " & strTable & ".Lieg_ID = "
& strID

DoCmd.SetWarnings False
DoCmd.RunSQL strSQL1
DoCmd.RunSQL strSQL2
DoCmd.RunSQL strSQL3
DoCmd.SetWarnings True

fehler:

If Err.Number <> 0 Then
    MsgBox Err.Number & " bitte überprüfen Sie ob alle Verknüpfungen
vorhanden! " & _
        Err.Description
End If

    Next
    Set objOrdner = Nothing
    Set objFSO = Nothing
End Function
-----

Public Function DB_Sicher()
    Dim Quelldatei As String
    Dim Zieldatei As String
    Dim oFSO As Variant
    Dim Current As String

    ' stellt fest ob die Datenbank eine Kopie ist oder nicht
    If CurrentProject.Name <> "Insel.mdb" Then
        MsgBox "Die Sicherung kann nicht auf eine Kopie" & _
            " angewendet werden."
    Exit Function
    End If

    Current = CurrentProject.FullName

    ' Name der Sicherungsdatei erstellen
    Zieldatei = Left(Current, Len(Current) - 4) & "_" & Year(Now) & ".mdb"
    '
        Set oFSO = CreateObject("Scripting.FileSystemObject")
        Quelldatei = Current
        oFSO.CopyFile Quelldatei, Zieldatei, True           ' Sicherung
erstellen

        MsgBox "Es wurde eine Sicherheitskopie unter " & _
            Zieldatei & " erstellt"

    'End If
End Function

```

### **7.6 Excel-Liegenschaftsdocumentaion**

Auf der mitgelieferte CD.

### **7.7 JAVA-Projektdateien (Vorbereitungstool)**

Auf der mitgelieferte CD.

### **7.8 ACCESS-Datenbanktool (Auswertungstool)**

Auf der mitgelieferte CD.

### **7.9 Erklärung der Selbständigkeit**

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(4) bzw. §25(4) ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, den 12.05.08

-----

Unterschrift