

**Diplomarbeit**

**Datenaustausch-**  
**und**  
**Analysetool**

**Stefan Boehning**

September 2006

**Titel der Diplomarbeit:** Datenaustausch- und Analyse Tool

**Stichworte:** MES, Web Service, Datenbank, ASP, C#, TCP/IP

**Zusammenfassung:**

Im Zuge der starken Zunahme von Automatisierung in Fertigungsbereichen steigt auch der Bedarf an produktionsrelevanten Daten (z.B. Stillstandszeiten, Qualität, etc.) für Beurteilungen und Entscheidungen an. Die E&K Automation Gruppe bietet mit ihren MES-CON-OS Produkten zurzeit die Möglichkeit, diese Daten aus SPS (Siemens S7) über ein VB-Tool in eine Datenbank zu schreiben und über ein MS Access-Frontend zu betrachten und auszuwerten. Im Zuge der Diplomarbeit soll diese Technik modernisiert und durch entsprechend neu zu erstellende Tools ersetzt werden. Der Datenaustausch wird durch das Tool MES-CON-OS COM vorgenommen. Es überträgt die Daten von einer Steuerung (SPS od. auch PC) in eine Datenbank. Die Datenanalyse teilt sich auf in MES-CON-OS WS, welches die gesammelten Daten für andere Programme zur Verfügung stellt und MES-CON-OS OEE, welches über ein Web-Frontend den Zugriff auf eine Statistikdarstellung für diese Produktionsdaten ermöglicht.

**Betreuender Professor:** Prof. Dr.-Ing. Ralf Haderler

**Zweitprüfer:** Prof. Dr. rer. nat. Helmut Faasch

**Hochschule:** Universität Lüneburg

**Fachbereich:** Automatisierungstechnik

**Studiengang:** Ingenieur-Informatik

**Firma:** E&K AUTOMATION - Eilers & Kirf GmbH

**Betreuender Mitarbeiter:** Dipl.-Ing. Dieter Sagewitz

**Abgegeben am:** 11.09.2006

## Danksagung

Mein besonderer Dank gilt meinen Eltern, **Günter und Christine Boehning**, die meine letzten Schritte dieses Lebensabschnittes leider nicht mehr begleiten konnten. Trotz schwerer Krankheit standen sie mir bis zum Schluss stets zur Seite und haben mich in allen Belangen tatkräftig unterstützt.

Weiterer Dank gilt der **Firma E&K Automation** für die Unterstützung und Betreuung meines beruflichen Werdeganges seit meinem ersten Praxissemester.

Besonders danke ich **Herrn Dieter Sagewitz** für die fachliche und persönliche Unterstützung während meiner Tätigkeit im Hause E&K Automation. Auch **Herrn Jürgen Kirf** möchte ich erwähnen, der mit großem Engagement trotz knapper Zeitressourcen immer zur Verfügung stand.

Auf Seiten der Universität Lüneburg möchte ich **Herrn Prof. Dr.-Ing. Ralf Haderler** meinen Dank aussprechen, der mich seit meinem ersten Praxissemester begleitet hat. Auch **Herrn Prof. Dr. rer. nat. Helmut Faasch** möchte ich danken, für die Übernahme der Zweitkorrektur, für die fachliche Betreuung meines Studiums, jedoch vor allem für die persönliche Unterstützung während des letzten Jahres.

Abschließend bedanke ich mich bei meiner Frau **Nadine**, die mir in den letzten Jahren viel Kraft und Halt gegeben hat und mir ständig zur Seite stand. Sie schenkte mir neuen Mut und Freude, ohne die ich diesen Lebensabschnitt sicherlich nicht gut bewältigt hätte.



1.	Die E&K AUTOMATION Gruppe .....	6
1.1.	Ist-Zustand und Problemdefinition .....	6
1.2.	Zielsetzung .....	8
2.	Grundlagen .....	9
2.1.	Abkürzungen / Fachbegriffe .....	11
2.2.	MES .....	12
2.2.1.	OEE .....	14
2.2.2.	Produktpalette E&K Automation .....	15
2.2.2.1.	MES-CON .....	15
2.2.2.2.	CONLOG-AG .....	16
2.3.	Webtechnologien .....	17
2.3.1.	Web Services .....	17
2.3.2.	ASP.NET .....	18
2.4.	Programmiersprachen / Arbeitsmittel .....	19
2.4.1.	Die Programmiersprache C# .....	19
2.4.1.1.	Dienste .....	20
2.4.2.	SQL .....	22
2.4.3.	HTML .....	23
2.4.4.	XML .....	23
2.4.5.	Dotnetcharting 2005 .....	24
3.	Datenaustausch- und Analysetool .....	27
3.1.	Datenaustauschtool .....	27
3.1.1.	Konzept, Funktionsüberblick .....	27
3.1.2.	Verwendete Nutzdaten .....	29
3.1.2.1.	XML-Konfigurationsdatei .....	29
3.1.2.2.	E&K Standard Telegramme .....	30
3.1.3.	Umsetzung .....	31
3.1.3.1.	Klasse cls_Telegramm .....	31
3.1.3.2.	Konfigurationstool .....	32
3.1.3.2.1.	Klassendiagramm .....	33
3.1.3.2.2.	Klasse cls_Config .....	34
3.1.3.2.3.	Klasse frm_MesconComController .....	35
3.1.3.2.4.	Screenshot und Kurzbeschreibung .....	36
3.1.3.3.	Serverdienst .....	38

3.1.3.3.1.	Klassendiagramm .....	38
3.1.3.3.2.	Klasse MesconComServer.....	39
3.1.3.3.3.	Klasse cls_Communication .....	40
3.1.3.3.4.	Klasse cls_Telegrammserver .....	41
3.1.3.3.5.	Klasse cls_DBVerbindung.....	41
3.2.	Analysetool.....	42
3.2.1.	Konzept, Funktionsüberblick.....	42
3.2.1.1.	Screenshots des alten MES-CON-OS OEE Access-Frontend.....	44
3.2.2.	Verwendete Nutzdaten .....	47
3.2.2.1.	Herkunft .....	47
3.2.2.2.	Struktur und Inhalt.....	48
3.2.3.	Umsetzung.....	50
3.2.3.1.	Web Service MES-CON-OS WS.....	50
3.2.3.1.1.	Klassendiagramm .....	51
3.2.3.1.2.	Service.asmx.....	52
3.2.3.1.3.	Service.cs.....	52
3.2.3.2.	Webseite MES-CON-OS OEE .....	54
3.2.3.2.1.	Seitenaufbau in HTML.....	54
3.2.3.2.2.	Codebehind C# .....	55
3.2.3.2.3.	Web.Config .....	56
3.2.3.2.4.	Screenshots und Funktionsbeschreibungen .....	57
4.	Vergleich MES-CON-OS und Wonderware .....	61
5.	Zusammenfassung / Ausblick / Bewertung.....	63
6.	Quellen .....	65
6.1.	Literaturverzeichnis.....	65
6.2.	Tabellenverzeichnis .....	66
6.3.	Abbildungsverzeichnis .....	67
7.	Erklärung zur Diplomarbeit .....	68
8.	Anhang .....	69
8.1.	Ablageort der Diplomarbeit .....	69
8.2.	Ablageort der Visual Studio - Projekte.....	69

## **1. Die E&K AUTOMATION Gruppe**

Die E&K Automation Gruppe ist spezialisiert auf Fahrerlose Transportsysteme und die Steuerung von industriellen Großanlagen (bspw. Brauereien, Molkereien, etc.). Die Gruppe schließt mehrere Unternehmen in ganz Europa zusammen, darunter auch die an der Gründung im Jahre 1980 beteiligte Eilers & Kirf GmbH in Rosengarten-Nenndorf.

Die gesamte Unternehmensgruppe befindet sich momentan in einer Neuorganisation der Strukturen. Die Unternehmensführung löst sich von der Ablauforganisation und strukturiert das Unternehmen gänzlich in Prozesse.

Auch die technischen Gegebenheiten werden in diesem Zuge neu ausgerichtet, so dass bestehende Softwarelösungen auf neue Technologien umgesetzt werden sollen und somit auch neue Funktionen und Bereiche abdecken müssen.

In diesem Zusammenhang entstand auch diese Diplomarbeit, die auf Basis von Webtechnologien arbeitet.

### **1.1. Ist-Zustand und Problemdefinition**

Die Erfahrungen der E&K Automation Gruppe haben in den letzten Jahren einen steigenden Bedarf an produktionsrelevanten Daten aufgezeigt. Zwischen den so genannten ERP-Systemen (Enterprise Resource Planning) und der Produktionsebene befindet sich jedoch eine Lücke, die durch so genannte Manufacturing Execution Systems (MES) geschlossen werden soll.

E&K Automation hat eine eigene Produktreihe namens MES-CON für diesen Bereich, welche aus einer Vielzahl verschiedener Programmkomponenten besteht. Eine Beschreibung der MES-CON Produktpalette befindet sich im Kapitel Grundlagen im Abschnitt 2.2.2.1. Mit diesen Tools muss es im Bedarfsfall möglich sein, relevante Daten schnell und übersichtlich zur Verfügung zu stellen.

Im speziellen werden hier zur Zeit zwei Tools verwendet:

- **MES-CON-OS COM**, welches als Schnittstelle zwischen einer SPS (Siemens SIMATIC S7) und einem PC dient. Zurzeit können hier maximal 2 Kopplungen gleichzeitig eingerichtet und konfiguriert werden. Das Tool empfängt Daten aus der SPS und speichert diese in einer SQL-Datenbank ab. MES-CON-OS COM läuft nicht automatisiert als Windowsdienst, sondern als normale Applikation, welche mit Visual Basic programmiert wurde.
- **MES-CON-OS OEE**, welches als MS Access-Frontend arbeitet. Über Abfragen wertet es Daten aus und stellt diese tabellarisch sowie grafisch für einen Endbenutzer dar. Die Nutzdaten für das Frontend liegen auf einer SQL-Datenbank. Für die nötigen Abfragen verwendet das Frontend eine Vielzahl zusätzlicher Tabellen, um das gewünschte Ergebnis zu liefern. Für die grafische Darstellung werden die aus MS Excel bekannten Diagramme aus dem MS Office Paket verwendet. Aus diesem Grund ist eine vorhandene Installation des MS Office Pakets auf dem jeweiligen Computer unbedingt erforderlich.

## **1.2. Zielsetzung**

Ziel der Diplomarbeit ist es, einen Grundstein für die Einführung neuer Technologien in der Softwareentwicklung im Hause E&K Automation zu legen. In diesem Zuge sollen erste Applikationen im Rahmen des MES-CON-OS Softwarekonzeptes erstellt werden. Mit diesen Applikationen soll es möglich sein, Produktionsdaten in einer zentralen Datenbank zu speichern (MES-CON-OS COM) und diese in aufbereiteter Form (MES-CON-OS WS) dem Kunden zur Verfügung stellen zu können (MES-CON-OS OEE). Die teilweise schon vorhandenen Softwaretools sind deshalb auf Basis von Webtechnologien und der Programmiersprache C#.NET komplett neu zu erstellen. Insbesondere legt hier das Unternehmen E&K Automation großen Wert auf eine zukünftige Vereinheitlichung der vielen unterschiedlichen Softwaretools, die zurzeit im Einsatz sind. Im Zuge dieser Diplomarbeit entstehen die ersten Programme in der Programmiersprache C#.NET, welche in naher Zukunft möglichst unternehmensweit verwendet werden soll. Bei der zu verwendenden Webtechnologie handelt es sich um so genannte Web Services, welche im Abschnitt 2.3.1 näher erläutert werden.

Dabei teilt sich die Realisierung der Aufgabe in verschiedene Bereiche ein:

- Einarbeitungen in folgende Themengebiete:
  - C#.NET
  - ASP.NET
  - XML
  - Web Services
  - SQL
  - MES speziell MES-CON-OS Produkte von E&K Automation
- Erstellung MES-CON-OS COM (Datenübertragung)
- Erstellung MES-CON-OS WS (Datenaufbereitung)
- Erstellung MES-CON-OS OEE (Datendarstellung)
- Test der Applikationen
- Erstellung einer Dokumentation



## 2. Grundlagen

Um das Zusammenspiel der verschiedenen Komponenten dieser Diplomarbeit zu verstehen (dargestellt in Abbildung 1: Zusammenspiel der MES-CON Komponenten), werden im Folgenden wichtige Grundlagen beschrieben sowie sich ständig wiederholende Abkürzungen erläutert.

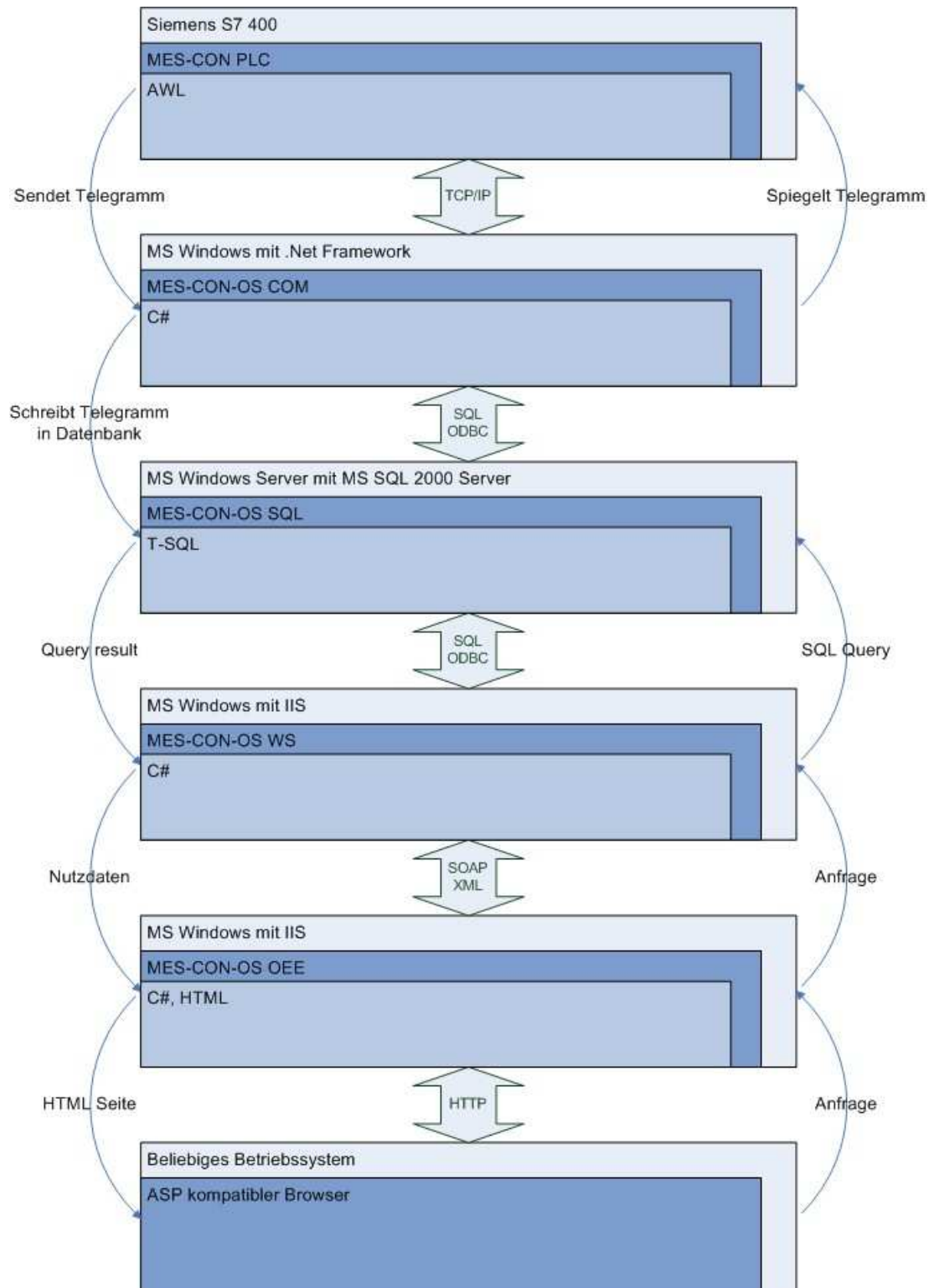
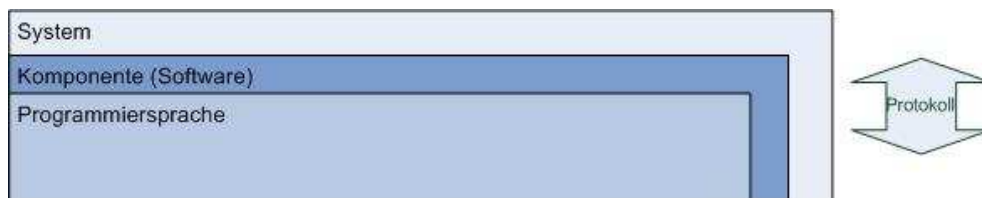


Abbildung 1: Zusammenspiel der MES-CON Komponenten

In dieser Abbildung ist das Zusammenspiel der MES-CON Komponenten erläutert. Dabei ist nicht nur die Komponente allein dargestellt, sondern auch die Systemumgebung sowie die Programmiersprache. Zusätzlich sind in Pfeilen die jeweiligen Kommunikationsprotokolle dargestellt. Die Pfeile seitlich an der Grafik geben Aufschluss über das inhaltliche Zusammenspiel.



**Abbildung 2: Legende**

## **2.1. Abkürzungen / Fachbegriffe**

Abk.	Bedeutet	Erklärung
ASP	Active Server Pages	siehe 2.3.2
CLR	Common Language Runtime	siehe 2.4.1
CP	Communication Processor	Kommunikationsbaugruppe einer SPS
ERP	Enterprise Resource Planning	Meist Softwaretools zur Planung von Unternehmensressourcen
HTML	Hypertext Markup Language	siehe 2.4.3
HTTP	Hypertext Transport Protocol	Protokoll zur Übertragung von Daten in einem Netzwerk
IIS	Internet Information Server	Webserver innerhalb der Microsoft Betriebssysteme
MES	Manufacturing Execution System	siehe 2.2
OEE	Overall Equipment Effectiveness	siehe 2.2.1
PLC	Programmable Logic Controller	entspricht einer SPS
SOAP	Simple Object Access Protocol	siehe 2.3.1
SPS	Speicherprogrammierbare Steuerung	Beispielsweise Siemens Simatic S7 300/400
SQL	Structured Query Language	siehe 2.4.2
VB	Visual Basic	Programmiersprache
XML	Extensible Markup Language	siehe 2.4.4

**Tabelle 1 : Abkürzungen**

## 2.2. MES

Nach einer Festlegung einer Reihe von Gremien, die letzten Endes in der ISA95 festgeschrieben wurde, definiert sich der Minimalumfang eines MES wie folgt: „Ein Manufacturing Execution System ist eine Informationsdrehscheibe innerhalb des Unternehmens, die fertigungsnahe Informationsquellen miteinander verknüpft.“<sup>1</sup>

Bei den „Informationsquellen“ handelt es sich um die zwei entscheidenden Ebenen eines Unternehmens:

- die Unternehmensleitebene (ERP), bspw. SAP
- die Automatisierungs- und Fertigungsebene, bspw. Sensoren / Aktoren

welche miteinander verbunden werden müssen. Diese „vertikale Integration“ ist in der nachfolgenden Grafik der Firma Siemens schematisch dargestellt.



Abbildung 3: Simatic Pyramide – Vertikale Integration

Verschiedene Faktoren haben in den letzten Monaten und Jahren eine starke Veränderung in der fertigenden Industrie mit sich gebracht. Allem voran eine entscheidende Gesetzesänderung, die es vorschreibt, Produktionsdaten einer ganzen Produktionskette für mögliche nötige Nachverfolgungen vorrätig zu halten,

<sup>1</sup> IT Production 03/06 Hintergrund MES-Wissen

um beispielsweise bei einer verdorbenen Ware eine ganze Charge von der Auslieferung zu stoppen oder Rückrufe zu ermöglichen.

Des Weiteren erfordert die Wirtschaft eine immer transparentere reaktionsfähigere Produktion, in der durch ständige Kontrolle und Planung schnell Entscheidungen getroffen werden können um unnötige Fehler und Stillstandszeiten einer Anlage zu minimieren.

Klassische Insellösungen der Fertigungsindustrie wie Betriebsdatenerfassung, Maschinendatenerfassung, Personalzeit, Management und Qualität müssen nun in einem möglichst interdisziplinären System zentral organisiert und verbunden werden, um den neuen Anforderungen gerecht zu werden.

Auf Seiten der Automatisierungsebene ist nicht zwingend eine Integration eines MES in eine Steuerung nötig, vielmehr geht es dabei um die Kopplung von Steuerungen mit einem MES, um möglichst exakt die anfallenden Produktionsdaten zu sammeln und auszuwerten. Auf Basis dieser Datensammlungen und -auswertungen ist es für die Unternehmensleitebene möglich, Entscheidungen zu treffen.

Ein Beispiel für eine Datensammlung und –Analyse aus dem Bereich MES ist das sog. OEE, welches im Abschnitt 2.2.1 erläutert wird.

### 2.2.1. OEE

„Der Begriff **Gesamtanlageneffektivität** (GAE) oder englisch **Overall Equipment Effectiveness** (OEE) bezeichnet eine vom Japan Institute of Plant Maintenance erstellte Kennzahl. Sie ist eines der Ergebnisse im Zuge der jahrzehntelangen Entwicklung des TPM-Konzeptes (Total Productive Maintenance). In die Gesamtanlageneffektivität werden Verluste mit einbezogen, die an einer Anlage entstehen und Auswirkungen auf die Produktion haben.

Die OEE einer Anlage ist als das Produkt der folgenden drei Faktoren definiert:

- dem *Nutzungsgrad* (Availability)
- dem *Leistungsgrad* (Performance Rate)
- der *Qualitätsrate* (Quality Rate)

Sein Wertebereich liegt zwischen 0 und 1."<sup>2</sup>

Der Nutzungsgrad definiert sich durch das Verhältnis der Nettoproduktionszeit zur Gesamtlaufzeit.

Der Leistungsgrad entspricht etwa dem bisherigen Wirkungsgrad. Er zeigt das Verhältnis der produzierten Istmenge zur Sollmenge, die in der Nettoproduktionszeit möglich wäre.

Die Qualitätsrate ist das Verhältnis Gutmenge – Ausschuss zur Gutmenge.

Für die OEE-Berechnung werden Daten der zu analysierenden Maschine oder Anlage benötigt. Im einfachsten Fall können diese manuell erfasst werden oder durch die fortgeschrittene Automatisierung direkt aus der Maschine oder Anlage übernommen werden.

So entsteht eine Art Logbuch, das für eine ganzheitliche Betrachtung über beliebige Zeiträume (Jahr, Quartal, Monat oder Woche) ausgewertet werden kann.

---

<sup>2</sup> <http://de.wikipedia.org/wiki/OEE> (11.09.06)

## **2.2.2. Produktpalette E&K Automation**

In den folgenden Abschnitten werden einige wichtige Produkte des Unternehmens E&K Automation vorgestellt.

### **2.2.2.1. MES-CON**

MES-CON ist das MES – Produkt des Unternehmens E&K Automation. Genau wie auch in der Definition arbeitet MES-CON als „Vermittlungsschicht“ zwischen der Sensor- / Aktor- und der ERP-Ebene.

Das Produkt MES-CON ist wie ein modularer Baukasten aufgebaut. So kann der Kunde das MES-CON nach seinen Wünschen und Aufgaben spezifizieren und in Auftrag geben. Grundsätzlich lässt sich das MES-CON Produkt in den Bereich PLC und OS aufteilen.

Der Bereich MES-CON PLC enthält viele fertige Programme und Standardbausteine, die für Siemens S7 Steuerungen erstellt wurden. Mit diesen Modulen lassen sich beispielsweise Sensoren und Aktoren steuern, Regelungsaufgaben lösen, Schrittketten steuern, Produkthistorien verfolgen und vieles mehr.

Einige Beispielkomponenten sind:

- MES-CON VM (Ventil- und Motorauswertung)
- MES-CON PID (PID-Regelkreise)
- MES-CON PRO (Schrittkettensteuerung)
- MES-CON REP (Datenauswertung, Reporting)

Teile der MES-CON PLC Bausteine sind in der Steuerung der CONLOG-AG enthalten, deren Funktion im folgenden Abschnitt 2.2.2.2 näher beschrieben wird.

Der Bereich MES-CON-OS umfasst Softwaretools, die auf einem Microsoft Windows 2000 Server laufen. Diese Tools agieren nicht in der Steuerung. Sie dienen vielmehr der übergeordneten Planung und Bewertung der Produktionsprozesse, um Prozesse transparenter zu gestalten und mit Hilfe geeigneter Kennzahlen zu verbessern.

Einige Beispielkomponenten sind:

- MES-CON-OS SQL (Microsoft SQL 2000 Server)
- MES-CON-OS ERP (ERP / PPS Schnittstelle)

- MES-CON-OS VIS (Prozessvisualisierung)

aber auch die durch diese Diplomarbeit neu entstandenen Komponenten

- MES-CON-OS COM (Datenaustausch)
- MES-CON-OS WS (Datenaufbereitung)
- MES-CON-OS OEE (Datenanalyse)

### **2.2.2.2. CONLOG-AG**

Die CONLOG-AG ist eine simulierte Anlage, welche für Messevorführungen und Entwicklungsaufgaben ins Leben gerufen wurde. Simuliert wird eine Anlage, in der Getränke abgefüllt werden. Hierzu liefert ein Tankfahrzeug Getränke an, welche in diversen Tanks zwischengelagert werden und später über zwei Füller abgefüllt werden. An dieses System gekoppelt ist ein Fahrerloses Transportsystem (FTS), welches die auf Paletten verpackten Getränke in einen Lagerbereich fährt. In der Steuerung für das Getränkeabfüllsystem ist ein von E&K Automation entwickeltes Eventsystem programmiert, mit dem es möglich ist, Systemmeldungen in einen Puffer der Steuerung zu schreiben, der bei vorhandener Netzwerkverbindung diese Daten an einen Computer sendet, und nach erfolgter Quittierung diese Daten aus dem Puffer löscht. Bei diesen Daten handelt es sich zum jetzigen Entwicklungsstand nicht um Produktionsdaten, sondern um willkürlich zusammengestellte Daten, die lediglich einen realen Zeitstempel enthalten. Diese Meldungen werden alle 5 Sekunden in den Sendepuffer der SPS geschrieben.

Hardwareseitig ist die CONLOG-AG mit 2 Steuerungen (FTS und PLC) und zwei Servern ausgestattet. Auf den Servern ist eine SQL-Datenbank und ein Internet Information Server installiert. Diese Hardware steht für Tests zur Verfügung.



## **2.3. Webtechnologien**

In den folgenden Abschnitten werden Webtechnologien beschrieben, auf denen diese Arbeit aufbaut. Diese Webtechnologien sind für E&K Automation ein wichtiger Anreiz, um auf diesem Gebiet Entwicklungsarbeit zu leisten. Sie haben für das Unternehmen einen hohen Stellenwert.

### **2.3.1. Web Services**

Der Sinn eines Web Service (dt: Webdienst) ist es, die Funktionalität vorhandener Applikationen auf eine einfache Art über das Internet zugänglich zu machen. Genauer genommen ist es eine RPC, eine Remote Prozedur, eine Prozedur eines Programms, die auf einen anderen Rechner ausgelagert wurde. Ein Client ruft diese Prozedur bei Bedarf auf und übermittelt Eingabeparameter. Die anhand der Parameter berechnete Ausgabe wird dann an den Aufrufer zurückgesendet. Mit der Übermittlung zwischen den beiden Softwarekomponenten braucht sich der Programmierer nicht zu befassen.

Man könnte einen Webdienst auch mit einer DLL vergleichen, die auf einem Webserver im Internet liegt und die in ihr ausprogrammierten Funktionen für andere Anwendungen global zur Verfügung stellt.

Treibende Kraft hinter der Entstehung der heutigen Web Service Technik war die Firma Microsoft, die eine Ablösung für die bis dahin bekannteren aber auch komplexeren Protokolle auf Basis von CORBA und D-COM suchte. Als komplett neues und vereinfachtes Protokoll entstand in diesem Zusammenhang das SOAP-Protokoll. SOAP steht für Simple Object Access Protocol. SOAP basiert auf dem Standard Übertragungsprotokoll des Internets, nämlich http. Es dient der Beschreibung von Typen, Parametern und Rückgabewerten sowie der Definition der selbigen. Außerdem benennt es Methoden, die ein Client aufrufen kann. SOAP verwendet eine XML-Grammatik. Da sowohl das Hypertext Transfer Protokoll (HTTP) als auch XML mittlerweile unabhängige Standards sind, lassen sich auch andere Programmierumgebungen für die Thematik der Web Services verwenden, beispielsweise Java. Mit Hilfe dieser Protokolle weiß auch ein Client, welche Funktionalität ein Webdienst zur Verfügung stellt.

Als Beispiel lässt sich an dieser Stelle die Firma Ebay nennen, welche im Zuge ihrer Auktionsplattform einen Web Service zur Verfügung stellt, mit dem es möglich ist, in selbst erstellten Programmen in der Datenbank des Auktionshauses nach Artikeln zu suchen.

### **2.3.2. ASP.NET**

„ASP.NET (Active Server Pages .NET, aktuelle Version 2.0 (2006)) ist eine serverseitige Technologie von Microsoft zum Erstellen von Webanwendungen auf Basis des Microsoft .NET-Frameworks.

Mit ASP.NET können Webanwendungen in beliebigen von .NET unterstützten Sprachen (theoretisch alle CLR-kompatiblen Sprachen) erstellt werden, zum Beispiel C#, VB.NET, [...].“<sup>3</sup>

ASP.NET-Seiten werden auf einem Internet Information Server (IIS) gehostet, welcher ein Microsoft Betriebssystem ab mindestens Windows 2000 voraussetzt. Im Gegensatz zu Unix/Linux gehosteten Sprachen, wie beispielsweise PHP, werden ASP.NET-Anwendungen auf dem Server kompiliert und nicht schrittweise interpretiert. HTML und Programmcode können getrennt werden. Dies steigert die Übersichtlichkeit des Programms. Programmierfehler werden durch das Kompilieren bereits sichtbar, nicht erst bei einem Aufruf der Applikation/Seite. ASP.NET ist aufgrund des .NET Frameworks wesentlich komplexer und somit auch schwieriger zu erlernen als Skriptsprachen wie PHP oder Perl.

Web Services in einer Microsoft Windows Betriebssystem-Umgebung benötigen ebenfalls einen ASP.NET Interpreter und müssen somit auch auf einem IIS betrieben werden.

---

<sup>3</sup> [http://de.wikipedia.org/wiki/Active\\_Server\\_Pages](http://de.wikipedia.org/wiki/Active_Server_Pages) (11.09.06)

## **2.4. Programmiersprachen / Arbeitsmittel**

Bedingt durch die schon vorhandenen Programme und Entwicklungsumgebungen im Hause Eilers & Kirf GmbH fiel die Wahl der Arbeitsmittel in den Bereich der durch das Microsoft Visual Studio abgedeckten Programmiersprachen. Durch die bereits angesprochene Modernisierung soll zukünftige Software möglichst in der Programmiersprache C#.NET entstehen, wobei zusätzlich für Internetprogrammierung die ASP.NET - Technologie zum Einsatz kommen soll. Als Datenbankserver wurde ein Microsoft SQL 2000 Server eingesetzt, der sich mittlerweile als Standard gegen Access-Datenbanken durchgesetzt hat und somit bei einer Vielzahl von Kunden bereits produktiv im Einsatz ist.

### **2.4.1. Die Programmiersprache C#**

Microsoft Visual C#.NET ist eine noch relativ junge und sehr leistungsfähige objektorientierte Programmiersprache, die erst seit dem Jahr 2003 durch die ISO standardisiert wurde. Sie ist fester Bestandteil des Microsoft .NET Frameworks und in die Entwicklungsumgebung Microsoft Visual Studio 2005 integriert, welche für diese Diplomarbeit verwendet wurde.

C# weist eine nicht zu bestreitende Ähnlichkeit mit der Programmiersprache Java auf. Geschweifte Klammern und Semikolons zur Syntaxstrukturierung sind hier genau wie in Java zu setzen. Schleifen und IF-Strukturen sind zum Verwechseln ähnlich.

Dennoch hat man bei der Erstellung des C#-Compilers versucht, möglichst viele Schwächen auszuhebeln und nützliche Funktionen einzubinden, um einen modernen sicheren Compiler zu gestalten. Als einige Beispiele wären anzuführen:

- Variablendeklarationen ziehen immer eine Initialisierung mit sich (Bei Nichtinitialisierung folgt ein Compilerfehler)
- Zuweisungen in IF-Bedingungen sind unzulässig (Falsch: IF(a=0) ergibt einen Compilerfehler)
- Speicherbereinigung erfolgt durch den Automatismus der Garbage Collection (GC) zur Vermeidung von überflüssiger Belegung durch nicht mehr verwendete Objekte
- Generics (Methodenaufrufe mit Parameterübergabe ohne Typfestlegung)

Zu Beginn einer C# Codedatei sind sog. Using-Direktiven eingebracht. Hier werden Bestandteile des .NET Frameworks mit eingebunden, ähnlich den „Package-Anweisungen“ in Java. Hier müssen eventuell Bestandteile des Frameworks, beispielsweise für SQL Abfragen eingebaut werden. Ohne diese Using-Direktiven stehen diese Funktionen sonst nicht zur Verfügung. An dieser Stelle müssen auch DLL's oder Web Services eingebunden werden.

Ausschlaggebend für die Verwendung von C# ist das Microsoft .NET Framework, mittlerweile in der Version 2.0, welches für fast alle aktuellen Microsoft Betriebssysteme zur Verfügung steht und somit eine plattformunabhängige Entwicklung ermöglicht. Das .NET Framework versucht die verschiedenen Programmiersprachen, Anwendungen, Entwicklungsumgebungen und Plattformen zu vereinheitlichen. Das .NET Framework stellt somit eine Klassenbibliothek zur Verfügung, die mit jeder Sprache aus dem Visual Studio .NET auf jedem Windowsbetriebssystem kompatibel ist. Dies ist das Hauptkriterium für das Unternehmen E&K Automation, diese Programmiersprachen zu verwenden.

#### **2.4.1.1. Dienste**

Dienste stellen eine etwas andere Art von Programmen unter einem Windows Betriebssystem dar. Dienste werden auch als kompilierte Exe-Datei ausgeführt, jedoch ist die Ausführung fest im System verankert. Dienste können nur durch das Betriebssystem gestartet werden oder durch einen manuellen Eingriff in die Management Konsole des Windows Betriebssystems. Wird das Programm durch das Betriebssystem automatisch gestartet, kann es auch nach einem Reboot des Systems wieder ohne weitere Eingriffe anlaufen. Diese Art der Programme bergen aber auch ein paar Nachteile. So muss das Programm für die Behandlung durch das System erst einmal in selbigem verankert werden. Für diesen Fall gibt es die Möglichkeit, Installationsroutinen mit dem Microsoft Visual Studio zu erstellen, die das kompilierte Projekt im System installieren und es in die System Management Konsole einbinden. Nun kann das System das Programm steuern bzw. einem Benutzer die Möglichkeit zum Eingriff geben. Eine manuelle Ausführung durch „Doppelklick“ der Programmdatei ist wirkungslos. Des Weiteren ist es möglich, das Programm über die Systemsteuerung über den Punkt „Software“ wieder zu deinstallieren.

Bei der Erstellung solcher Dienste ist eine gesonderte Art der Programmierung und Planung notwendig. Windowsdienste müssen nach spätestens 30 Sekunden eigenständig laufen und dem Betriebssystem eine Rückmeldung geben. Sollte dies nicht der Fall sein, wird der Dienst automatisch wieder gestoppt.

So ist es erforderlich, auf das Prinzip der Threadprogrammierung zurückzugreifen.

„Ein **Thread** (auch **Aktivitätsträger**), in der deutschen Literatur vereinzelt auch als **Faden** bezeichnet, ist in der Informatik ein Ausführungsstrang beziehungsweise eine Ausführungsreihenfolge der Abarbeitung der Software.“

(Zitat Wikipedia)

Das eigentliche Programm wird in separate Threads unterteilt. Die Startroutine des Dienstes ist nur dazu da, die einzelnen Threads zu starten. So ist es möglich, nach erfolgreichem Thread-Start die erforderliche Rückmeldung binnen 30 Sekunden an das System zu liefern und die Programmausführung aufrecht zu halten.

Somit steckt in der Startroutine des Dienstes der Aufruf der benötigten Threads. Die eigentliche Programmlogik verbirgt sich in den einzelnen Threads.

Dienste bieten jedoch keine grafische Programmoberfläche, wie sie durch Windows-Forms gegeben ist, da sie meist keine Beeinflussung durch einen Benutzer erfahren und stetig die gleichen Aufgaben erfüllen.

### **2.4.2. SQL**

„SQL [...] ist eine deklarative Datenbanksprache für relationale Datenbanken. SQL (im allgemeinen Sprachgebrauch als Abkürzung für „Structured Query Language“ aufgefasst, obwohl laut ANSI-Standard ein eigenständiger Name) ist aus SEQUEL ([...], Structured English Query Language) hervorgegangen, das von IBM in den 1970er Jahren auf der Grundlage des bahnbrechenden Artikels „A Relational Model of Data for Large Shared Data Banks“ (1970) von Edgar F. Codd entworfen wurde.

SQL hat eine relativ einfache Syntax, die an die englische Umgangssprache angelehnt ist, und stellt eine Reihe von Befehlen zur Definition von Datenstrukturen nach der relationalen Algebra, zur Manipulation von Datenbeständen (Anfügen, Bearbeiten und Löschen von Datensätzen) und zur Abfrage von Daten zur Verfügung. Durch ihre Rolle als Quasi-Standard ist SQL von großer Bedeutung, da eine weitgehende Unabhängigkeit von der benutzten Software erzielt werden kann. Die meisten SQL-Implementierungen bieten darüber hinaus allerdings noch herstellerspezifische Erweiterungen, die nicht dem Standard-Sprachumfang entsprechen, was zur Folge hat, dass von den Herstellern parallel entwickelte gleiche Funktionen unterschiedliche Sprachelemente benutzen. [...]“<sup>4</sup>

---

<sup>4</sup> <http://de.wikipedia.org/wiki/Sql> (11.09.06)

### **2.4.3. HTML**

Um Inhalte, wie beispielsweise Texte, Bilder oder auch Hyperlinks, auf Internetseiten zu strukturieren verwendet man die Hypertext Markup Language (HTML). Dabei handelt es sich um eine Auszeichnungssprache. Die Struktur erhält eine Webseite durch das verwenden von Tags (z.B. `<b>` für Fettschrift, `<font>` für Schrifteinstellungen, `<img>` für Grafiken, `<a>` für Hyperlinks, `<h1>` für Überschriften, etc.). Es gibt immer ein öffnendes und ein schließendes Element (z.B. `<b>Dies wird fett geschrieben</b>`).

Folgender Beispielcode gibt eine einfache Webseite mit dem Satz „Hello World!“ aus:

```
<html>
<head><title>Webseite Titel</title></head>
<body>Hello World!</body>
</html>
```

### **2.4.4. XML**

„Die Extensible Markup Language (engl. für „erweiterbare Auszeichnungs-Sprache“), abgekürzt XML, ist ein Standard zur Erstellung maschinen- und menschenlesbarer Dokumente in Form einer Baumstruktur, der vom World Wide Web Consortium (W3C) definiert wird. XML definiert dabei die Regeln für den Aufbau solcher Dokumente. Für einen konkreten Anwendungsfall ("XML-Anwendung") müssen die Details der jeweiligen Dokumente spezifiziert werden. Dies betrifft insbesondere die Festlegung der Strukturelemente und ihre Anordnung innerhalb des Dokumentenbaums. XML ist damit ein Standard zur Definition von beliebigen in ihrer Grundstruktur jedoch stark verwandten Auszeichnungssprachen. Eine Sprache wie XML zur Definition anderer Sprachen nennt man Metasprache. XML ist eine vereinfachte Teilmenge von SGML.“<sup>5</sup>

---

<sup>5</sup> <http://de.wikipedia.org/wiki/XML> (11.09.2006)

### **2.4.5. Dotnetcharting 2005**

Für die Darstellung von Diagrammen gibt es Softwareprodukte von externen Firmen, die verschiedene Funktionalitäten bieten. Dabei standen drei Produkte zur näheren Auswahl: Dotnetcharting 2005, CrystalReports, List&Label. Für die Darstellung von Diagrammen sind alle Produkte geeignet, jedoch bieten CrystalReports und List&Label einen sehr viel größeren Funktionsumfang. Hierzu gehören z.B. automatisch generierte Reports, Tabellen, Barcodes. Dieser Funktionsumfang macht diese Produkte unnötig teuer. Sie eignen sich somit nicht für den Einsatz bei E&K Automation.

Dotnetcharting in der Version 2005 besteht aus einer DLL, welche in eigene Softwareprojekte eingebunden werden kann. So können Diagramme als Objekte in jeder beliebigen .NET-Programmiersprache verwendet werden. Dabei ist es egal, ob es sich um eine Form-basierende Windowsanwendung oder um eine ASP Webseite handelt. Dotnetcharting bietet eine Vielzahl von Diagrammtypen (horizontal, vertikal, Balken, Säulen, etc.), die in Funktion und Optik im Quelltext angepasst werden können. Zur Verdeutlichung folgt ein Ausschnitt aus den Programmierbeispielen zu Dotnetcharting 2005:

Einbindung in die Webseite (mit C# Quellcode):

```
<%@ Page Language="C#" Description="dotnetCHARTING Component"%>
<%@ Register TagPrefix="dotnet" Namespace="dotnetCHARTING"
Assembly="dotnetCHARTING"%>
<%@ Import Namespace="System.Drawing" %>
<script runat="server">
```

```
SeriesCollection getRandomData()
{
    SeriesCollection SC = new SeriesCollection();
    Random myR = new Random();
    for(int a = 0; a < 4; a++)
    {
        Series s = new Series();
        s.Name = "Series " + a;
        for(int b = 0; b < 4; b++)
        {
            Element e = new Element();
```



```
        e.Name = "E " + b;
        e.YValue = myR.Next(50);
        s.Elements.Add(e);
    }
    SC.Add(s);
}
// Set Different Colors for our Series
SC[0].DefaultElement.Color = Color.FromArgb(49,255,49);
SC[1].DefaultElement.Color = Color.FromArgb(255,255,0);
SC[2].DefaultElement.Color = Color.FromArgb(255,99,49);
SC[3].DefaultElement.Color = Color.FromArgb(0,156,255);
return SC;
}
void Page_Load(Object sender,EventArgs e)
{
    // Set the title.
    Chart.Title="My Chart";
    // Set the chart Type
    Chart.Type = ChartType.ComboHorizontal;
    // Set 3D
    Chart.Use3D = true;
    // Set a default transparency
    Chart.DefaultSeries.DefaultElement.Transparency = 20;
    // Set the x axis label
    Chart.XAxis.Label.Text="X Axis Label";
    // Set the y axis label
    Chart.YAxis.Label.Text="Y Axis Label";
    // Set the x axis scale
    Chart.XAxis.Scale = Scale.FullStacked;
    // Set the directory where the images will be stored.
    Chart.TempDirectory="temp";
    // Set the bar shading effect
    Chart.ShadingEffect = false;
    // Set the chart size.
    Chart.Width = 600;
    Chart.Height = 350;
    // Add the random data.
    Chart.SeriesCollection.Add(getRandomData());
}
</script>
<html xmlns="http://www.w3.org/1999/xhtml"><head><title>Gallery Sample</title></head>
```

```
<body>  
<div style="text-align:center">  
<dotnet:Chart id="Chart" runat="server"/>  
</div>  
</body>  
</html>
```

Diagrammansicht:

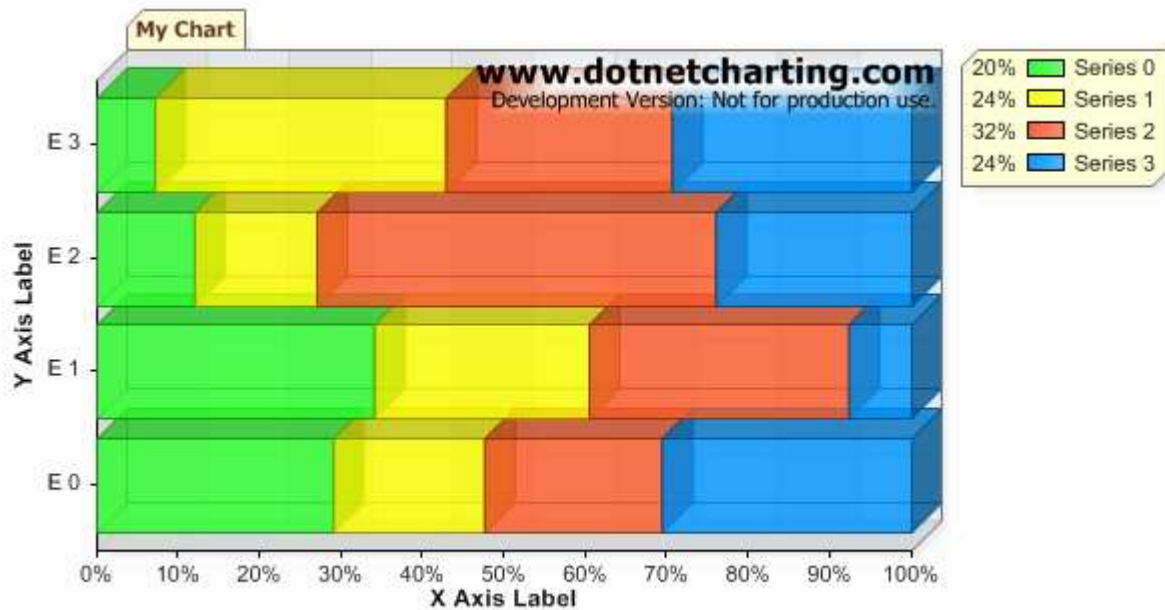


Abbildung 4: Balkendiagramm aus der Beispielsammlung von dotnetcharting2005

### **3. Datenaustausch- und Analysetool**

In den folgenden Abschnitten werden die Softwarebestandteile dieser Diplomarbeit beschrieben. Es wird auf die Funktion und das Konzept der Tools eingegangen, wichtige Methoden und Klassen beschrieben, Klassendiagramme dargestellt sowie die wichtigen Nutzdaten in Struktur und Inhalt beschrieben.

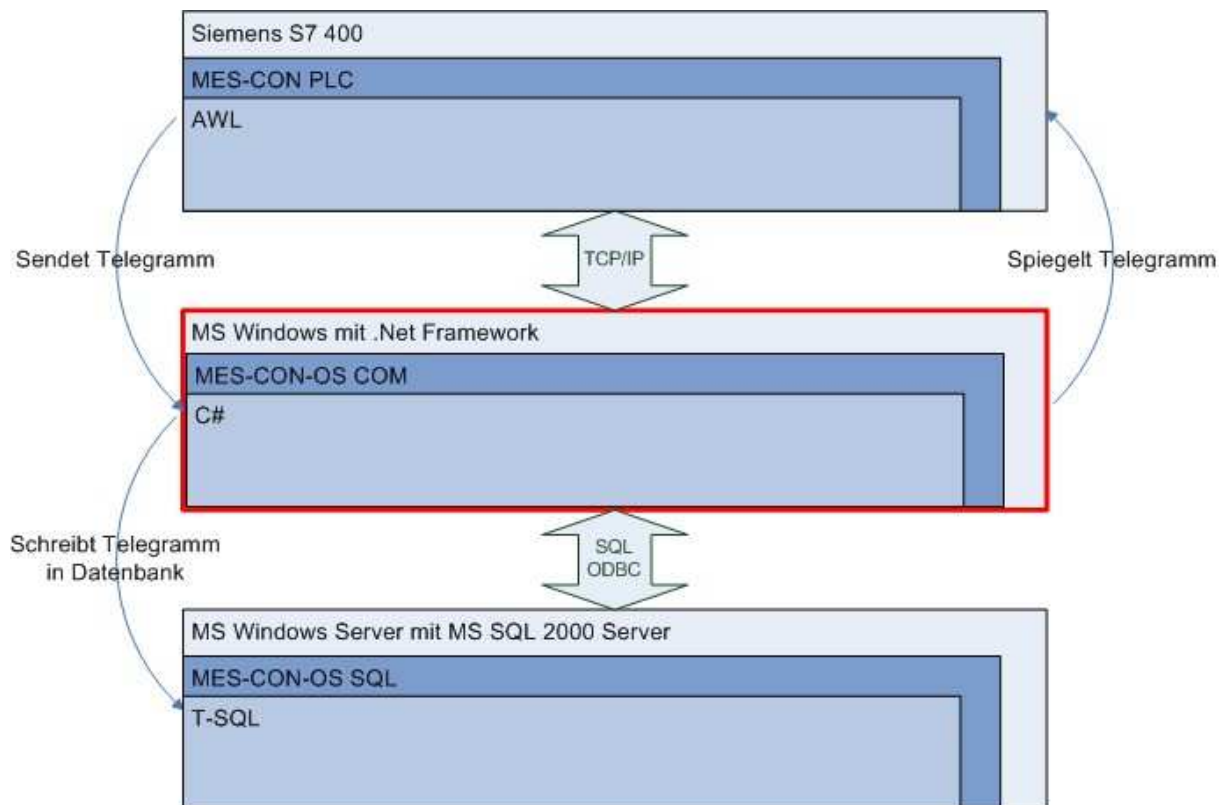
#### **3.1. Datenaustauschtool**

In diesem Abschnitt wird das neu erstellte Datenaustauschtool, im folgenden MES-CON-OS COM genannt, beschrieben. Um dieses Tool in einer geeigneten Umgebung zu testen, wurde die Umgebung der CONLOG-AG verwendet. Auf einer zur CONLOG-AG gehörenden Workstation befindet sich eine SQL-Datenbank mit einer Tabelle, in der die aus der Steuerung empfangenen Telegramme gespeichert werden sollen.

##### **3.1.1. Konzept, Funktionsüberblick**

MES-CON-OS COM dient als Kommunikationsschnittstelle. Es verbindet einen PC mit einer SPS und empfängt Daten, die die Steuerung versendet. Die bereitgestellten Daten werden in dem folgenden Abschnitt 3.1.2 erläutert. Die Bereitstellung der Daten auf Seiten der SPS ist nicht Teil dieser Arbeit. Sie ist bereits durch das Unternehmen E&K Automation realisiert, wie im Abschnitt 2.2.2.2 beschrieben. Die Kommunikation wird durch einen in der Steuerung befindlichen Kommunikationsprozessor (CP) aufgebaut. Er verbindet sich aktiv mit einem Server. Der Verbindungsaufbau lässt sich nicht beeinflussen, er wird durch die Steuerung selber verwaltet. Bei bestehender Verbindung werden die Daten aus einem Puffer gelesen und via TCP/IP an den Verbindungspartner gesendet. Der Verbindungspartner muss diese Daten zwingend spiegeln, da sie ansonsten nicht aus dem Datenpuffer der SPS entfernt werden.

Der folgende Ausschnitt der zu Beginn präsentierten Abbildung 1: Zusammenspiel der MES-CON Komponenten gibt Aufschluss über die Verbindung der am Datenaustausch teilnehmenden Systeme. Hervorgehoben ist das in diesem Bereich beschriebene Programm.



**Abbildung 5: Ausschnitt zum Bereich Datenaustausch**

Somit war es erforderlich, das neue MES-CON-OS COM so aufzubauen, das es eine Verbindung einer Steuerung erwartet (passiver Verbindungsaufbau) und bei erfolgreichem Verbindungsaufbau die Daten der Steuerung empfängt, in eine Datenbank schreibt und danach spiegelt.

MES-CON-OS COM wurde im Gegensatz zum alten, bereits bestehenden Tool als Dienst realisiert, um die Möglichkeit zu bieten, nach einem System(neu)start eigenständig anzulaufen.

Zusätzlich wurde eine grafische Oberfläche geschaffen, mit der es möglich ist, den Dienst zu stoppen, zu starten und die benötigten Konfigurationsdaten (IP-Adressen, Datenbanken, etc.) in einer XML-Datei zu speichern. Dieses Programm wird im Folgenden als „Konfigurationstool“ bezeichnet.

### 3.1.2. Verwendete Nutzdaten

Im folgenden Abschnitt werden die verschiedenen Nutzdaten und ihre Struktur erläutert, wie sie im Datenaustauschtool verwendet werden. Dabei handelt es sich um eine XML-Datei, die dazu dient, die Programmkonfiguration auf Dauer zu sichern. Bei den anderen handelt es sich um Standardtelegramme der Firma E&K Automation.

#### 3.1.2.1. XML-Konfigurationsdatei

Die XML-Konfigurationsdatei wird als Schnittstelle zwischen dem Dienst und der grafischen Oberfläche verwendet. Sowohl der Dienst als auch die grafische Oberfläche nutzen dieselbe Datei. Der Dienst liest lediglich in seiner Einstiegsroutine die Daten der XML-Konfigurationsdatei aus, um mit diesen Informationen die nötigen Verbindungen aufzubauen.

Die grafische Oberfläche hingegen kann Informationen verändern und diese in die XML-Konfigurationsdatei sichern. Die Verbindungen sind somit konfigurierbar und nicht fest im Quellcode verankert.

Folgende Daten werden in der XML-Konfigurationsdatei verwendet:

Name	Daten-Typ	Erklärung
ControlSetNumber1	Integer	Enthält die Nummer des Datensatzes
ControlSetName1	String	Ermöglicht die Bezeichnung eines Datensatzes
IsServer1	Boolean	Aktiver oder passiver Verbindungsaufbau
TimeSync1	Boolean	Aktiviert Zeitsynchronisation zwischen den Teilnehmern
Identifizier1	String	Identifizier zur Kennzeichnung des Teilnehmers
LocalPort1	Integer	Lokale Portnummer des Dienstrechners
LocalIP1	String	Lokale IP des Dienstrechners
RemotePort1	Integer	Remote Portnummer des Verbindungspartners
RemoteIP1	String	Remote IP des Verbindungspartners
DatabaseIP1	String	Name bzw. IP des Datenbankservers
DatabaseUser1	String	Datenbank Benutzer
DatabasePassword1	String	Datenbank Passwort
DatabaseName1	String	Name der Datenbank
DatabaseTableName1	String	Name der Tabelle

**Tabelle 2 : Struktur der XML-Konfigurationsdatei**

### 3.1.2.2. E&K Standard Telegramme

E&K Automation hat für den Datenaustausch zwischen verschiedenen Steuerungen und Computern eine feste Telegrammstruktur definiert. Generell lassen sich drei verschiedene Telegrammtypen unterscheiden:

- MF-Telegramm (für Meldungen in einem Fahrerlosen Transportsystem, die Transportaufträge enthalten)
- PA-Telegramm (für Meldungen in der Prozessautomatisierung, enthält 20 Real-Werte)
- OS-Telegramm (Format für die Verarbeitung von Meldungen für Fremdsysteme)

Alle drei Telegrammarten haben eine fest definierte Länge von 108 Byte. Dabei haben alle einen gemeinsamen Telegrammkopf. Er besteht aus 28 Byte und enthält alle in der nachfolgenden „Tabelle 3: Aufbau Telegrammkopf“ dargestellten Informationen.

Byte	Inhalt	Datentyp	Erklärung
0+1	Kennung	Char	Bestimmung der Datenstruktur
2+3	Länge	Integer	Gesamtlänge des Telegramms
4+5	Nummer	Integer	laufende Nummer
6+7	PLC_Jahr	Integer	Datum, Jahr
8+9	PLC_Monat	Integer	Datum, Monat
10+11	PLC_Tag	Integer	Datum, Tag
12+13	PLC_Stunde	Integer	Zeit, Stunde
14+15	PLC_Minute	Integer	Zeit, Minute
16+17	PLC_Sekunde	Integer	Zeit, Sekunde
18+19	PLC_ms	Integer	Zeit, Millisekunde
20+21	Modul-ID	Integer	Kategorisiert Telegramme tiefer (E&K Function Identifier)
22+23	Sub_ID	Integer	Entscheidet über Verarbeitung der Telegramme
24+25	Source	Integer	Organisatorische Nummer des Senders
26+27	Object	Integer	Modulnummer innerhalb des Senders

**Tabelle 3: Aufbau Telegrammkopf**

An diesen gemeinsamen Telegrammkopf schließt sich ein Bereich von 80 Bytes an, der je nach Telegramm anders aufgeteilt ist. Im Bereich des PA-Telegramms handelt es sich um 20 Real-Werte, die in 80 Bytes aufgeteilt sind. Somit belegt ein Real-Wert 4 Byte. Die beiden anderen Telegrammtypen spielen für diese Diplomarbeit keine entscheidende Rolle und werden an dieser Stelle nicht weiter betrachtet.

### 3.1.3. Umsetzung

Im folgenden Abschnitt wird die Umsetzung des Datenaustauschtools MES-CON-OS COM beschrieben. Die Umsetzung ist in drei Abschnitte unterteilt. Der Bereich 3.1.3.2 befasst sich mit der Umsetzung der Windows-Forms Anwendung, um den Dienst zu steuern und die Konfigurationsdaten zu verändern. Der Bereich 3.1.3.3 beschreibt die Umsetzung des Windowsdienstes. Vor diesen Abschnitten wird jedoch eine Klasse zur Erstellung von Telegramm-Objekten beschrieben. Diese Klasse entstand zusammen mit einem Mitarbeiter der Firma E&K Automation als Übung zur Objektorientierte Programmierung im Zuge einer Einarbeitungsphase.

#### 3.1.3.1. Klasse cls\_Telegramm

Die Klasse cls\_Telegramm dient dazu, Daten der Standardtelegramme in einem Objekt zu behandeln. Die Klasse behandelt alle drei Telegrammtypen. Wie bereits in Abschnitt 3.1.2.2 beschrieben, sind die Telegrammköpfe identisch. Aus diesem Grunde wurde eine abstrakte Klasse cls\_Telegramm generiert, die nicht instanziiert werden kann. Neben dieser abstrakten Klasse gibt es nun drei Klassen, passend zu den drei Telegrammtypen. Diese Klassen lassen sich instanziiieren. Sie erben die Eigenschaften der übergeordneten Klasse cls\_Telegramm und somit die identische Struktur. Aufgrund dieser Struktur können die einzelnen Telegramme gut verändert und an bestimmte Umgebungen (kundenspezifisch) angepasst werden.

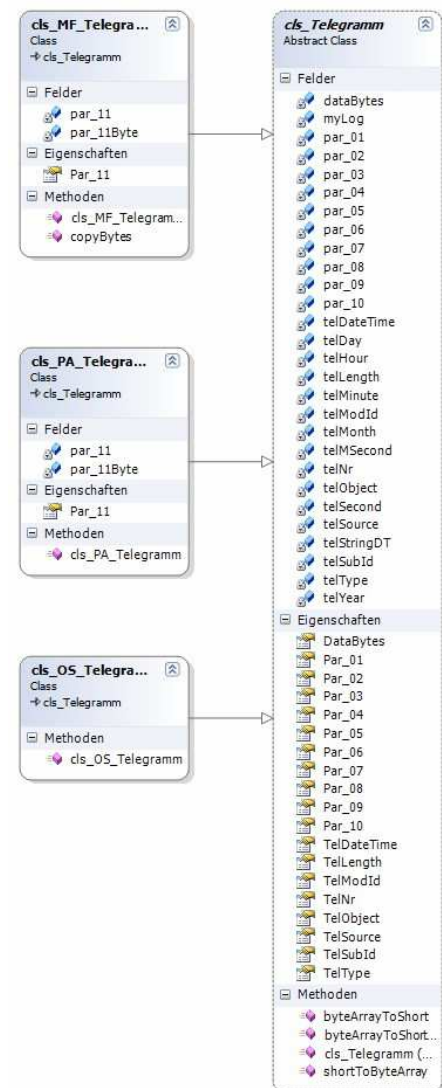


Abbildung 6: Klassendiagramm cls\_Telegramm



### **3.1.3.2. Konfigurationstool**

Das Konfigurationstool entstand als Übung und Einarbeitung in die Aufgabenstellung und den Umgang mit dem Microsoft Visual Studio 2005 und der Programmiersprache C#. Das Konfigurationstool ist eine Windows-Forms Anwendung, also eine Applikation, die aus Fenstern besteht. In diesem Fall gibt es ein einziges Hauptfenster. In diesem Fenster befinden sich Textfelder, mit denen die Konfiguration der Verbindungsteilnehmer für MES-CON-OS COM möglich ist. Über Button-Steurelemente ist es möglich, diese Konfigurationsdaten in eine XML-Konfigurationsdatei zu speichern.



### 3.1.3.2.1. Klassendiagramm

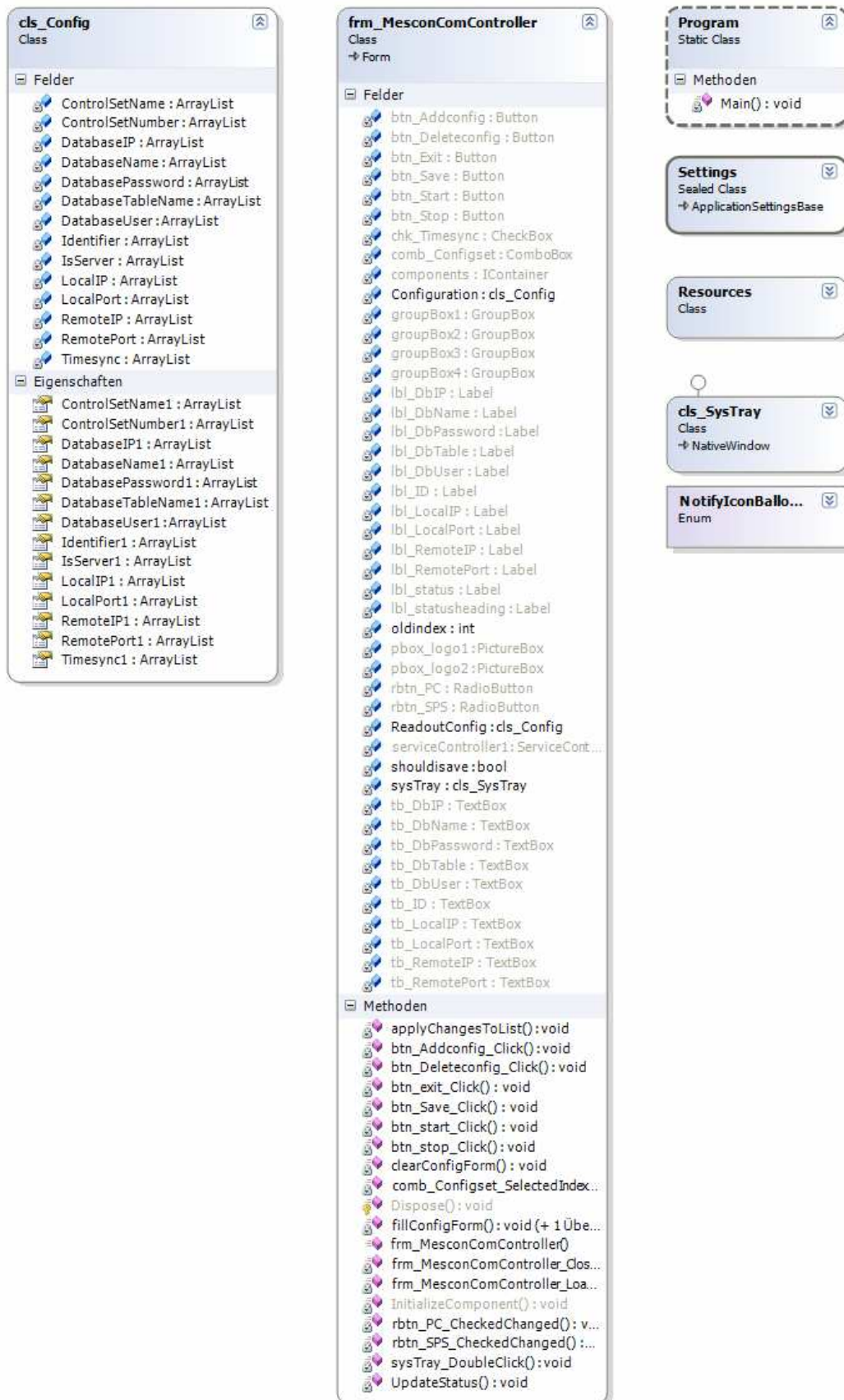


Abbildung 7: Klassendiagramm Konfigurationstool für MES-CON-OS COM

### 3.1.3.2.2. Klasse `cls_Config`

Diese Klasse beinhaltet alle wichtigen Konfigurationsparameter. Diese sind als `ArrayList` deklariert, um die Möglichkeit zu bieten, eine beliebige Anzahl an Datensätzen zu realisieren. Ein normales `Array` braucht eine definierte Anzahl von Datensätzen zur Initialisierung. Dies ist nicht möglich, da zu Programmstart die Anzahl dieser Datensätze nicht feststeht. Die Klasse `cls_Config` beinhaltet keine Methoden. Sie beinhaltet lediglich alle nötigen Felder, um eine Verbindung zu definieren. Diese Felder sind folgende:

Name	Daten-Typ	Erklärung
<code>ControlSetNumber</code>	<code>ArrayList</code>	Enthält die Nummer des Datensatzes
<code>ControlSetName</code>	<code>ArrayList</code>	Ermöglicht die Bezeichnung eines Datensatzes
<code>IsServer</code>	<code>ArrayList</code>	Aktiver oder passiver Verbindungsaufbau
<code>TimeSync</code>	<code>ArrayList</code>	Aktiviert Zeitsynchronisation zwischen den Teilnehmern
<code>Identifizier</code>	<code>ArrayList</code>	Identifizier zur Kennzeichnung des Teilnehmers
<code>LocalPort</code>	<code>ArrayList</code>	Lokale Portnummer des Dienstrechners
<code>LocalIP</code>	<code>ArrayList</code>	Lokale IP des Dienstrechners
<code>RemotePort</code>	<code>ArrayList</code>	Remote Portnummer des Verbindungspartners
<code>RemoteIP</code>	<code>ArrayList</code>	Remote IP des Verbindungspartners
<code>DatabaselP</code>	<code>ArrayList</code>	Name bzw. IP des Datenbankservers
<code>DatabaseUser</code>	<code>ArrayList</code>	Datenbank Benutzer
<code>DatabasePassword</code>	<code>ArrayList</code>	Datenbank Passwort
<code>DatabaseName</code>	<code>ArrayList</code>	Name der Datenbank
<code>DatabaseTableName</code>	<code>ArrayList</code>	Name der Tabelle

**Tabelle 4 : Felder der Klasse `cls_Config`**

Alle Variablen wurden als `privat` deklariert. Entsprechend wurden für alle Variablen `Get-` und `Set-`Methoden programmiert, um den Zugriff aus einer anderen Klasse zu ermöglichen. Die Klasse `cls_Config` wurde somit gekapselt. Die Variablen sind nicht direkt zugänglich, sondern nur über ihre `Get-` und `Set-`Methoden.

Diese Klasse wird sowohl im Konfigurationstool als auch im Dienst verwendet. Zusammen mit der XML-Konfigurationsdatei bildet sie eine Schnittstelle zwischen beiden Programmen. Während der Dienst nur lesend auf die XML-Konfigurationsdatei zugreift, schreibt das Konfigurationstool auch in diese Datei.

### 3.1.3.2.3. Klasse frm\_MesconComController

Die Klasse frm\_MesconComController bildet das eigentliche Fenster der Anwendung. Folgende Methoden bilden die Funktionalität der Anwendung:

Name	Beschreibung
applyChangesToList	Fügt alle Änderungen der Form in die Arraylist des instanziierten cls_Config Objekt ein.
btn_Addconfig_Click	Fügt einen neuen Konfigurationsdatensatz mit Standartwerten ein.
btn_Deleteconfig_Click	Löscht den aktuell ausgewählten Konfigurationsdatensatz.
btn_exit_Click	Eventhandler beim Klicken des Button-Steurelements: Schließt die Anwendung.
btn_Save_Click	Eventhandler beim Klicken des Button-Steurelements: Speichert die
btn_start_Click	Eventhandler beim Klicken des Button-Steurelements: Startet den Dienst, sofern er nicht schon gestartet ist.
btn_stop_Click	Eventhandler beim Klicken des Button-Steurelements: Stoppt den Dienst, sofern er noch nicht gestoppt ist.
clearConfigForm (int removeindex1)	Hilfsmethode zum Löschen eines bestimmten Datensatzes.
comb_Configset_ SelectedIndexChanged	Eventhandler Combobox-Steurelement: Belegt die Form mit dem aus der Combobox ausgewählten Datensatz.
fillConfigForm(int cfgindex)	Hilfsmethode zum Füllen der Form. Es wird der durch den Index bestimmte Datensatz verwendet.
frm_MesconComController_Closing	Eventhandler beim Schließen der Form: Programmende über dieses Ereignis wird unterdrückt. Schließen nur durch den „Stop-Button“ möglich.
frm_MesconComController_Loading	Eventhandler beim Laden der Form: Lädt die XML-Konfigurationsdatei, belegt die Form vor.
rbtn_PC_CheckedChanged	Eventhandler Radiobutton-Steurelement: Umschaltung zwischen den Radiobuttons.
rbtn_SPS_CheckedChanged	Eventhandler Radiobutton-Steurelement: Umschaltung zwischen den Radiobuttons.
sysTray_DoubleClick	Eventhandler SystemTray Doppelklick: Maximiert die Form bei Doppelklick, wenn diese verborgen war.
UpdateStatus	Eventhandler Timerüberlauf: Aktualisiert die Anzeige des Dienststatus (Stopped/Running).

**Tabelle 5 : Methoden der Klasse frm\_MesconComController**

Auf eine Auflistung aller verwendeten Variablen im Konfigurationstool wird an dieser Stelle verzichtet. Es finden sich hauptsächlich Steuerelemente für Windows-Forms (Labels, Buttons, Radiobuttons, etc.) sowie einige Hilfsvariablen und ein Objekt vom Typ `cls_Config`, in dem die Konfigurationsdatensätze gespeichert sind, in dieser Applikation. Dies ist auch in der Abbildung 7: Klassendiagramm Konfigurationstool für MES-CON-OS COM zu erkennen.

### 3.1.3.2.4. Screenshot und Kurzbeschreibung

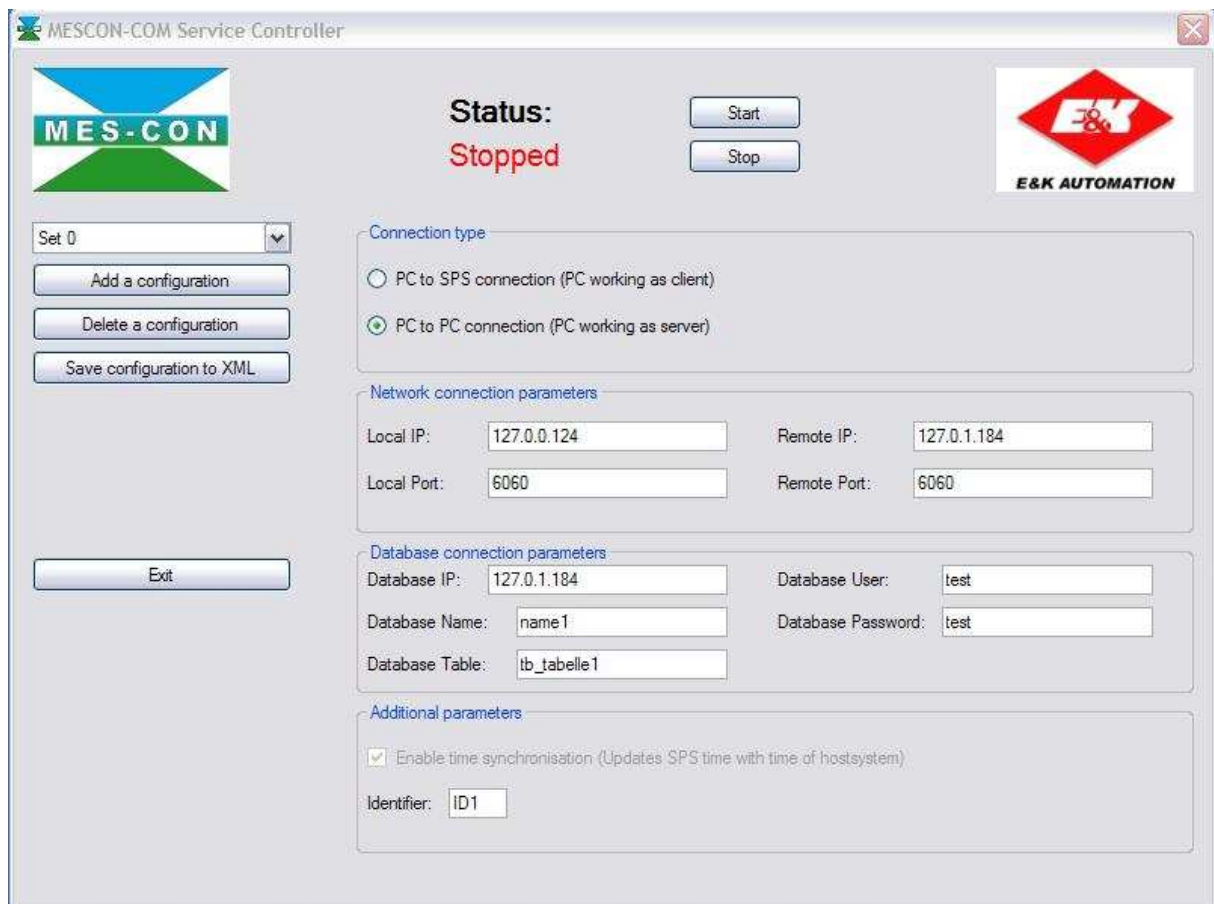


Abbildung 8: Screenshot des Konfigurationstools

Die Abbildung zeigt das laufende Konfigurationstool. Zentral am oberen Rand befindet sich die Steuerung des Dienstes. Über die Button-Steuerelemente kann er gestartet und gestoppt werden. Die Anzeige aktualisiert sich durch einen Timer. Im rechten Bereich des Fensters befinden sich die Steuerelemente für die Konfiguration. Sie bestehen hauptsächlich aus Labels und Textboxen. Die Möglichkeit zur Zeitsynchronisation wurde lediglich für mögliche Folgeprojekte bzw. Updates eingeplant, jedoch nicht im Rahmen dieser Arbeit behandelt. Über die Button-Steuerelemente am linken Bildrand lassen sich die Konfigurationen speichern,

löschen, neu hinzufügen sowie die Anwendung schließen. Bei Veränderungen an der Konfiguration muss diese explizit mit einem Klick auf den Button „Save configuration to XML“ gespeichert werden. Ansonsten gehen Einstellungen verloren. Das Schließen der Form über das Kreuz in der rechten oberen Fensterecke wurde unterbunden. Die Anwendung wird in diesem Fall lediglich in das Systemtray minimiert. So werden Fehlbedienungen ausgeschlossen, ein versehentliches Schließen der Anwendung ohne Klick auf den Button „Exit“ ist nicht möglich. Um die Anwendung wieder zu maximieren, ist ein Doppelklick auf das Symbol im Systemtray notwendig.



### 3.1.3.3. Serverdienst

Der Serverdienst ist das Kernstück der MES-CON-OS COM Anwendung. Im Visual Studio wird dieser bei Erstellung der Projektmappe als Windows Dienstanwendung ausgewählt.

#### 3.1.3.3.1. Klassendiagramm

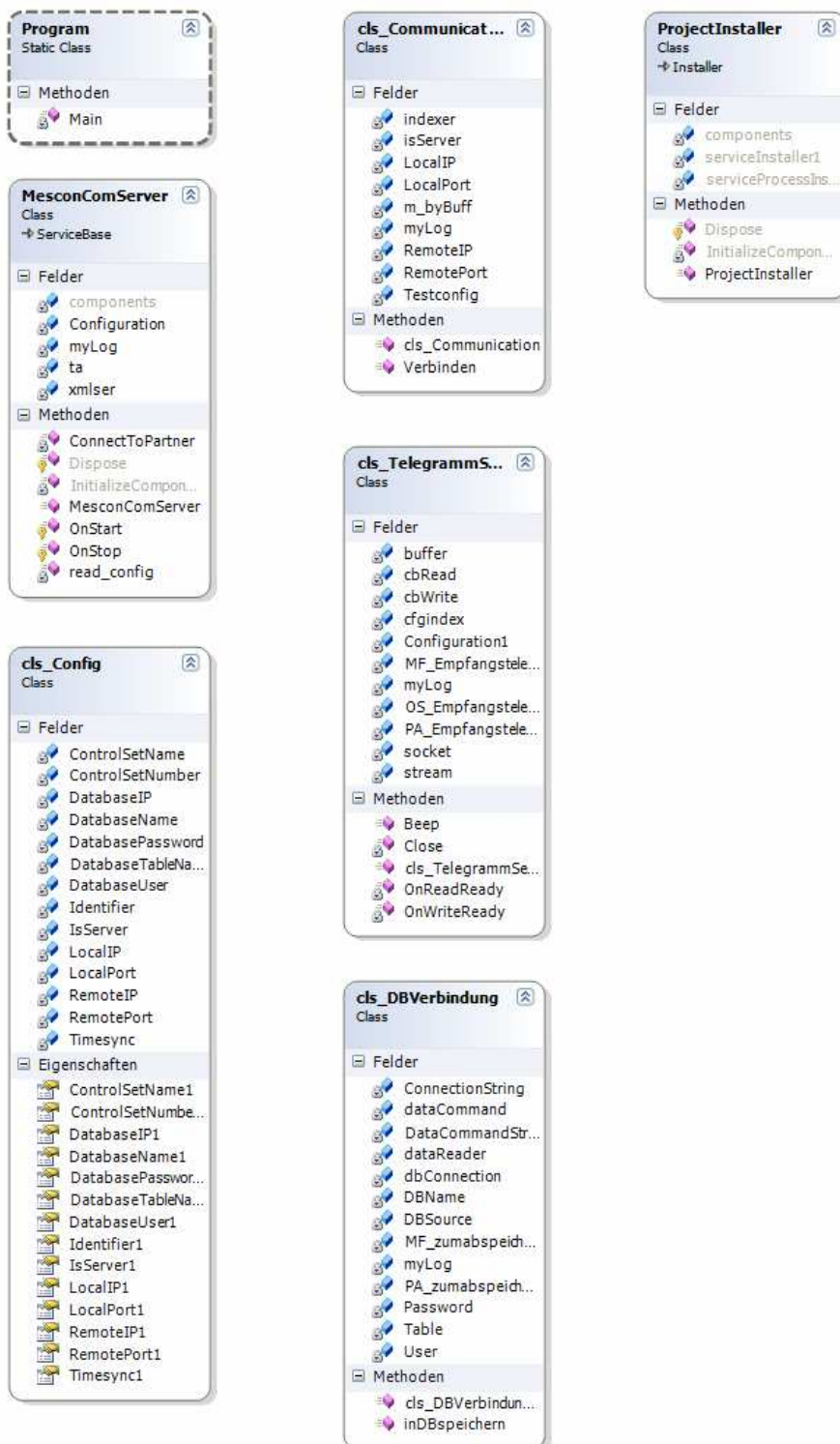


Abbildung 9: Klassendiagramm MES-CON-OS COM

### 3.1.3.3.2. Klasse MesconComServer

In der Klasse „MesconComServer“ befindet sich die Einstiegsmethode. Hier wird zuerst die XML-Konfigurationsdatei beim Start des Dienstes ausgelesen. Dies geschieht nur einmal. Wenn die Konfiguration durch das Konfigurationstool verändert wird, muss der Dienst neu gestartet werden.

Nach erfolgreichem Einlesen wird für jeden Datensatz, der eine Verbindung darstellt, ein eigener Thread gestartet, der die weitere Kommunikation übernimmt. Somit lässt sich der Dienst innerhalb 30 Sekunden aufstarten. Es kommt nicht zu einer Fehlermeldung. Die Klasse „MesconComServer“ bietet folgende Methoden:

Name	Beschreibung
OnStart()	Führt die Funktion read_config() aus, startet pro Datensatz einen Thread, der dann die Methode ConnectToPartner() aufruft.
OnStop()	Beendet alle laufenden Threads.
read_config()	Liest die XML-Konfigurationsdatei aus und speichert diese in einem Objekt vom Typ cls_Config (beschrieben in 3.1.3.2.2)
ConnectToPartner	Startmethode eines Threads. Instanziert ein neues Objekt vom Typ cls_Communication und startet deren Methode Verbinden()

**Tabelle 6 : Methoden der Klasse „MesconComServer“**

### **3.1.3.3.3. Klasse cls\_Communication**

Die Klasse cls\_Communication dient der Kommunikation mit dem Verbindungspartner. Sie bekommt das cls\_Config Objekt übergeben und mit ihm einen Index für den jeweiligen Datensatz, den diese Instanz bearbeitet. Durch die Threadprogrammierung wird je Datensatz ein Thread gestartet. Jeder Thread instanziiert genau ein cls\_Communication Objekt mit dem dazugehörigen Datensatz. In der Instanziierung wird eine Endlosschleife gestartet, in der ein sog. TCP-Listener auf einen Verbindungsaufbau wartet. Dieser Listener läuft solange, bis der Thread der Anwendung beendet wird. Somit ist jederzeit ein Verbindungsaufbau möglich. Bei erfolgreicher Verbindung wird ein Socket für die Kommunikation gebildet. Nun wird ein Objekt vom Typ cls\_Telegrammserver erstellt, welches den Socket übergeben bekommt.

Name	Beschreibung
Verbinden()	Wartet in einer Endlosschleife auf Verbindung. Erstellt dann einen Socket und ein neues Objekt zum weiteren Datenaustausch

**Tabelle 7 : Methoden der Klasse „cls\_Communication“**



#### 3.1.3.3.4. Klasse cls\_Telegrammserver

In der Klasse cls\_Telegrammserver findet der eigentliche Datenaustausch der Verbindsteilnehmer statt. Der Datenaustausch ist asynchron programmiert. Die Methoden für das Senden und Empfangen werden somit „angestoßen“. Diese Methoden werden ebenfalls in einem separaten Thread ausgeführt. Somit wird das System nicht blockiert, während diese Methoden noch ausgeführt werden. Bei erfolgreichem Empfang gibt es eine weitere Methode, welche den kompletten Empfang „bestätigt“. In dieser Methode wird das Datensenden asynchron gestartet. Bei erfolgreichem Senden, wechselt dies, wie in einem Schneeballsystem, hin und her. Bei erfolgreichem Empfang von Daten wird ein Objekt vom Typ des jeweiligen Telegramms mit der Basisklasse cls\_Telegramm erstellt (3.1.3.1). Gleichzeitig wird eine Instanz der Klasse DBVerbindung erstellt und das Telegramm übergeben.

Name	Beschreibung
OnWriteReady(IAsyncResult iar)	Bei abgeschlossenem Schreibvorgang wird das Lesen der Daten angestoßen.
OnReadReady(IAsyncResult iar)	Bei abgeschlossenem Lesevorgang wird die Datenbankverbindung instanziiert und ein Telegramm erstellt und übergeben. Write-Methode wird angestoßen, um die Daten zu spiegeln
Close()	Schließt die Verbindung bei Fehlern.

**Tabelle 8 : Methoden der Klasse „cls\_Telegrammserver“**

#### 3.1.3.3.5. Klasse cls\_DBVerbindung

In der Klasse cls\_DBVerbindung wird anhand der Telegrammdaten und des Konfigurationsdatensatzes die Verbindung zur Datenbank aufgebaut. Es wird eine Insert SQL Anweisung erstellt und der Telegramminhalt in die Datenbank gespeichert.

Name	Beschreibung
inDBspeichern	Speichert Telegrammdaten in die Datenbank

**Tabelle 9 : Methoden der Klasse „cls\_Communication“**

## **3.2. Analysetool**

In den folgenden Abschnitten werden der Webservice, im Folgenden als MES-CON-OS WS, und die Webseite zur Darstellung der Analyse, weiterhin als MES-CON-OS OEE beschrieben. Um auch diese Programme testen zu können, wurde auf reale Produktionsdaten der Firma Friesland Foods zurückgegriffen, welche Im Abschnitt 3.2.2 beschrieben werden. Die Struktur dieser Daten und die damit verbundene Telegrammstruktur weicht von der Struktur in der CONLOG-AG ab. Diese Abweichung ist für diese Diplomarbeit nicht relevant, da es um das Aufzeigen der Möglichkeiten ging. Das Analysetool ist somit auf die Umgebung der Firma Friesland Foods zugeschnitten. Es findet also ein Schnitt in der Testumgebung statt. Datenaustausch und Datenanalyse sind in der Testumgebung CONLOG-AG zum jetzigen Entwicklungsstand nicht durchgängig

### **3.2.1. Konzept, Funktionsüberblick**

Der Web Service MES-CON-OS WS soll Funktionen bereitstellen, mit denen aus einer vorgegebenen Datenbankstruktur Daten aus dem Bereich OEE verarbeitet und an eine sich anschließende Applikation weitergeleitet werden können. Hierbei sollen drei Bereiche abgedeckt werden:

OEE Event:

Bei der Funktion OEE Event handelt es sich um eine tabellarische Darstellung aller Datenbankinhalte, welche als OEE Event Meldungen eingestuft werden. Diese Daten sollen in einer geeigneten Form an eine Applikation zur tabellarischen Darstellung weitergesendet werden. Es soll die Möglichkeit gegeben sein, einen Anfangs- und einen Enddatumswert zu übergeben, um die OEE Events zeitlich einzugrenzen.

OEE Analyse:

Bei der Funktion OEE Analyse sollen anhand der OEE Event Meldungen für einen per Datum ausgewählten Zeitraum alle zugehörigen OEE Werte berechnet werden und in einer geeigneten Struktur an eine weitere Applikation zur Darstellung weitergesendet werden.

OEE Trend:

Die Funktion OEE Trend soll es ermöglichen, für einen ausgewählten Zeitraum den OEE Wert wöchentlich kumuliert zu berechnen und in einer geeigneten Form an eine weitere Applikation weiterzusenden, um diese Daten dort als Trendgrafik darstellen zu können. Die folgende Abbildung zeigt den Ausschnitt aus dem Zusammenspiel der Komponenten für den Bereich Datenanalyse. Hervorgehoben sind die in diesem Abschnitt beschriebenen Programme.

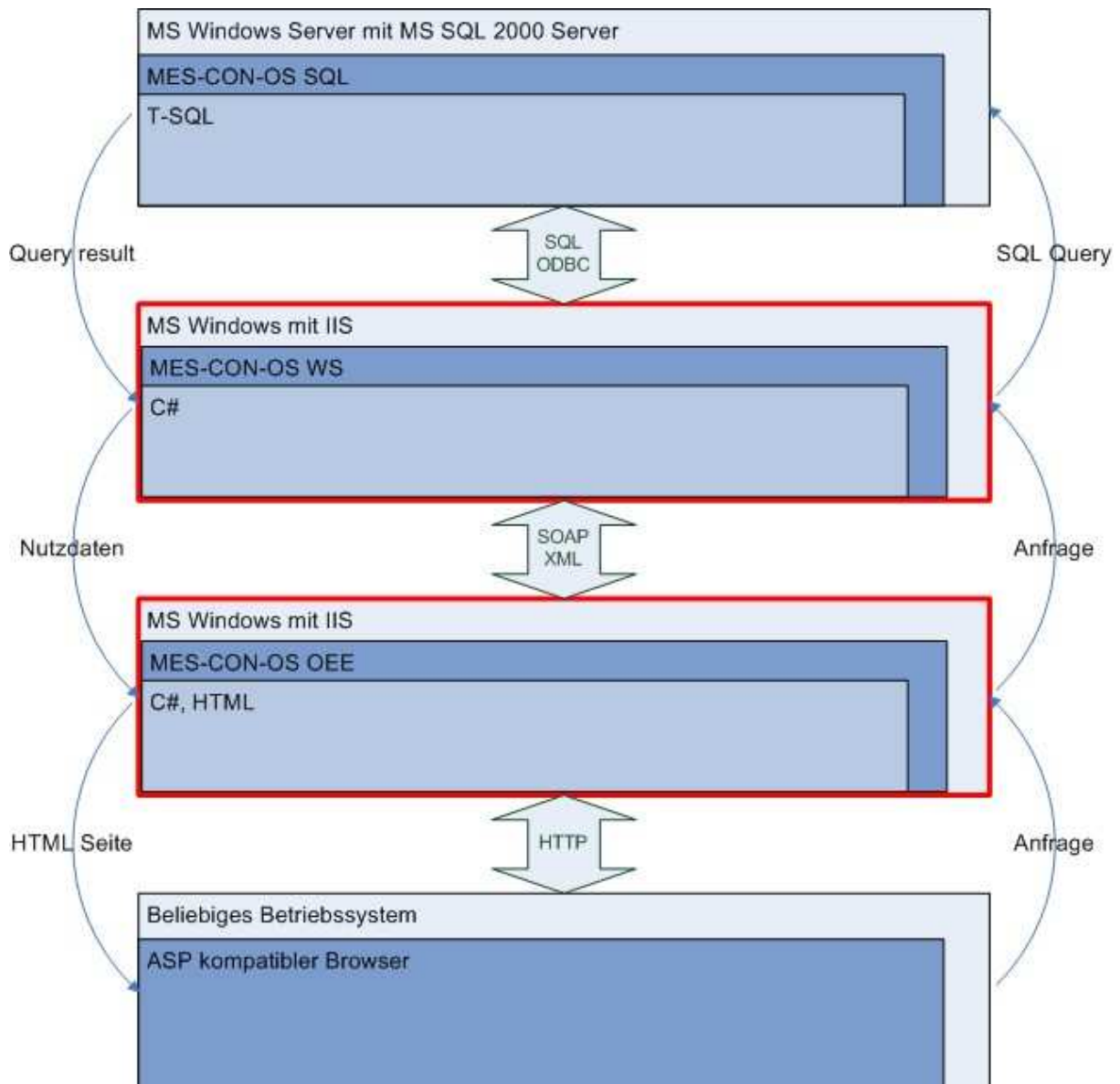


Abbildung 10: Ausschnitt zum Bereich Datenanalyse

### 3.2.1.1. Screenshots des alten MES-CON-OS OEE Access-Frontend

An dieser Stelle werden Screenshots des alten Access-Frontend dargestellt, um den Funktionsüberblick des MES-CON-OS OEE Tools, der im vorigen Abschnitt erläutert wurde, zu verdeutlichen. Diese vorhandenen Funktionen sollen dem Access-Frontend in der Funktion nachgebildet werden.

Start-Time	Stop-Time	Duration	OEE-Status	OEE-Group	Charge	Product	Total	Good	Correct	Com	
2006.04.30 23:00:01	01.05.2006 01:00:00	01:00:00	21	Einde Productie	AU	Available unused tin	558	10054	0,00	0,00	0,00
2006.05.01 00:00:00	01.05.2006 01:00:00	01:00:00	21	Einde Productie	AU	Available unused tin	558	10054	0,00	0,00	0,00
2006.05.01 01:00:00	01.05.2006 02:00:00	01:00:00	21	Einde Productie	AU	Available unused tin	558	10054	0,00	0,00	0,00
2006.05.01 02:00:00	01.05.2006 02:16:53	00:16:53	21	Einde Productie	AU	Available unused tin	558	10054	0,00	0,00	0,00
2006.05.01 02:16:53	01.05.2006 02:20:50	00:04:03	116	Orderwissel-geen ombouw	RPS	Routine Production	406973	10202	0,00	0,00	0,00
2006.05.01 02:20:50	01.05.2006 02:21:00	00:00:10	50	Produceert	P	Production time [P]	406973	10202	0,00	0,00	0,00
2006.05.01 02:21:00	01.05.2006 03:00:00	00:38:55	11	Bedrijf Gesloten	UA	Unavailable time	406973	10202	0,00	0,00	0,00
2006.05.01 03:00:00	01.05.2006 04:00:00	01:00:00	11	Bedrijf Gesloten	UA	Unavailable time	406973	10202	0,00	0,00	0,00
2006.05.01 04:00:00	01.05.2006 05:00:00	01:00:00	11	Bedrijf Gesloten	UA	Unavailable time	406973	10202	0,00	0,00	0,00
2006.05.01 05:00:00	01.05.2006 05:02:59	00:02:59	11	Bedrijf Gesloten	UA	Unavailable time	406973	10202	0,00	0,00	0,00
2006.05.01 05:02:59	01.05.2006 05:03:14	00:00:15	21	Einde Productie	AU	Available unused tin	406973	10202	0,00	0,00	0,00
2006.05.01 05:03:14	01.05.2006 05:03:29	00:00:11	11	Bedrijf Gesloten	UA	Unavailable time	406973	10202	0,00	0,00	0,00
2006.05.01 05:03:29	01.05.2006 05:03:44	00:00:18	21	Einde Productie	AU	Available unused tin	406973	10202	0,00	0,00	0,00
2006.05.01 05:03:44	01.05.2006 05:04:17	00:00:36	118	Ombouw inpakker	RPS	Routine Production	406973	10202	0,00	0,00	0,00
2006.05.01 05:04:17	01.05.2006 05:39:22	00:35:06	124	Opstarten na ombouw lage	RPS	Routine Production	406973	10202	0,00	0,00	0,00
2006.05.01 05:39:22	01.05.2006 05:39:30	00:00:08	125	Reiniging Vulmachine	RPS	Routine Production	406973	10202	0,00	0,00	0,00
2006.05.01 05:39:30	01.05.2006 05:45:44	00:06:14	121	Opstarten na reiniging (CIP)	RPS	Routine Production	406973	10202	0,00	0,00	0,00
2006.05.01 05:45:44	01.05.2006 05:45:50	00:00:10	50	Produceert	P	Production time [P]	406973	10202	0,00	0,00	0,00
2006.05.01 05:45:50	01.05.2006 05:47:24	00:01:30	11	Bedrijf Gesloten	UA	Unavailable time	406973	10202	0,00	0,00	0,00
2006.05.01 05:47:24	01.05.2006 05:47:34	00:00:10	50	Produceert	P	Production time [P]	406973	10202	0,00	0,00	0,00
2006.05.01 05:47:34	01.05.2006 05:48:00	00:00:26	11	Bedrijf Gesloten	UA	Unavailable time	406973	10202	0,00	0,00	0,00
2006.05.01 05:48:00	01.05.2006 05:49:53	00:01:53	21	Einde Productie	AU	Available unused tin	406973	10202	0,00	0,00	0,00
2006.05.01 05:49:53	01.05.2006 05:49:55	00:00:02	50	Produceert	P	Production time [P]	406973	10202	0,00	0,00	0,00
2006.05.01 05:49:55	01.05.2006 05:49:55	00:00:04	11	Bedrijf Gesloten	UA	Unavailable time	406973	10202	0,00	0,00	0,00
2006.05.01 05:49:55	01.05.2006 05:50:30	00:00:32	21	Einde Productie	AU	Available unused tin	406973	10202	0,00	0,00	0,00
2006.05.01 05:50:30	01.05.2006 05:50:40	00:00:11	50	Produceert	P	Production time [P]	406973	10202	0,00	0,00	0,00
2006.05.01 05:50:40	01.05.2006 05:54:30	00:03:51	137	Botertoevoer geen boter	USF	Unexpected supply f	406973	10202	0,00	0,00	0,00
2006.05.01 05:54:30	01.05.2006 05:54:30	00:00:00	137	Botertoevoer geen boter	USF	Unexpected supply f	406973	10202	0,00	0,00	0,00
2006.05.01 05:54:30	01.05.2006 05:55:00	00:00:36	50	Produceert	P	Production time [P]	406973	10202	0,00	0,00	0,00
2006.05.01 05:55:00	01.05.2006 05:55:00	00:00:01	50	Produceert	P	Production time [P]	406973	10202	0,00	0,00	0,00
2006.05.01 05:55:00	01.05.2006 05:55:00	00:00:03	50	Produceert	P	Production time [P]	406973	10202	0,00	0,00	0,00
2006.05.01 05:55:00	01.05.2006 05:55:22	00:00:15	50	Produceert	P	Production time [P]	406973	10202	0,00	0,00	0,00
							1.356,46	253.983,94	249.212,56	0,00	

Abbildung 11: MES-CON-OS OEE (alt) - Events

Diese Abbildung zeigt die tabellarische Übersicht über alle OEE-Events.

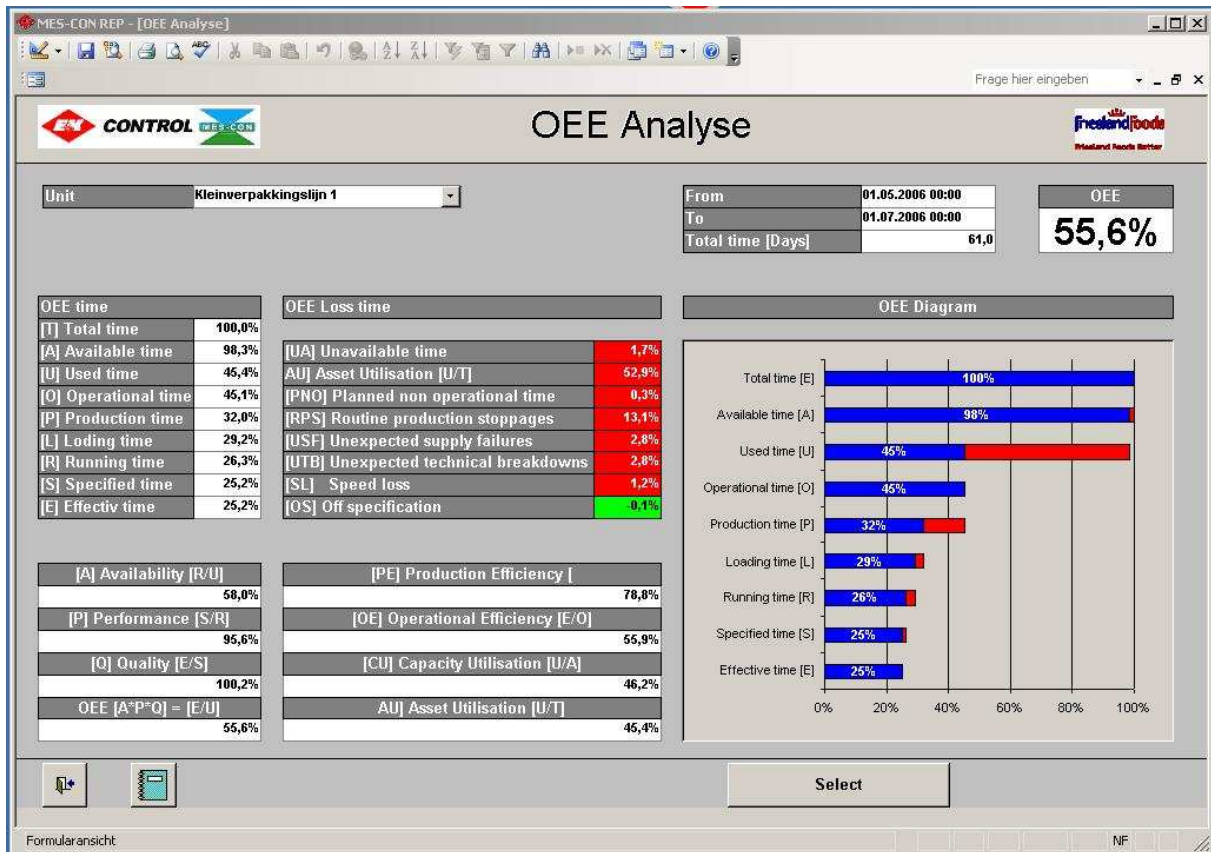


Abbildung 12: MES-CON-OS OEE (alt) - Analyse

In dieser Abbildung ist die OEE-Analyse dargestellt. Sämtliche wichtigen Werte sind auf dieser Oberfläche zugänglich und werden in einem Diagramm dargestellt.

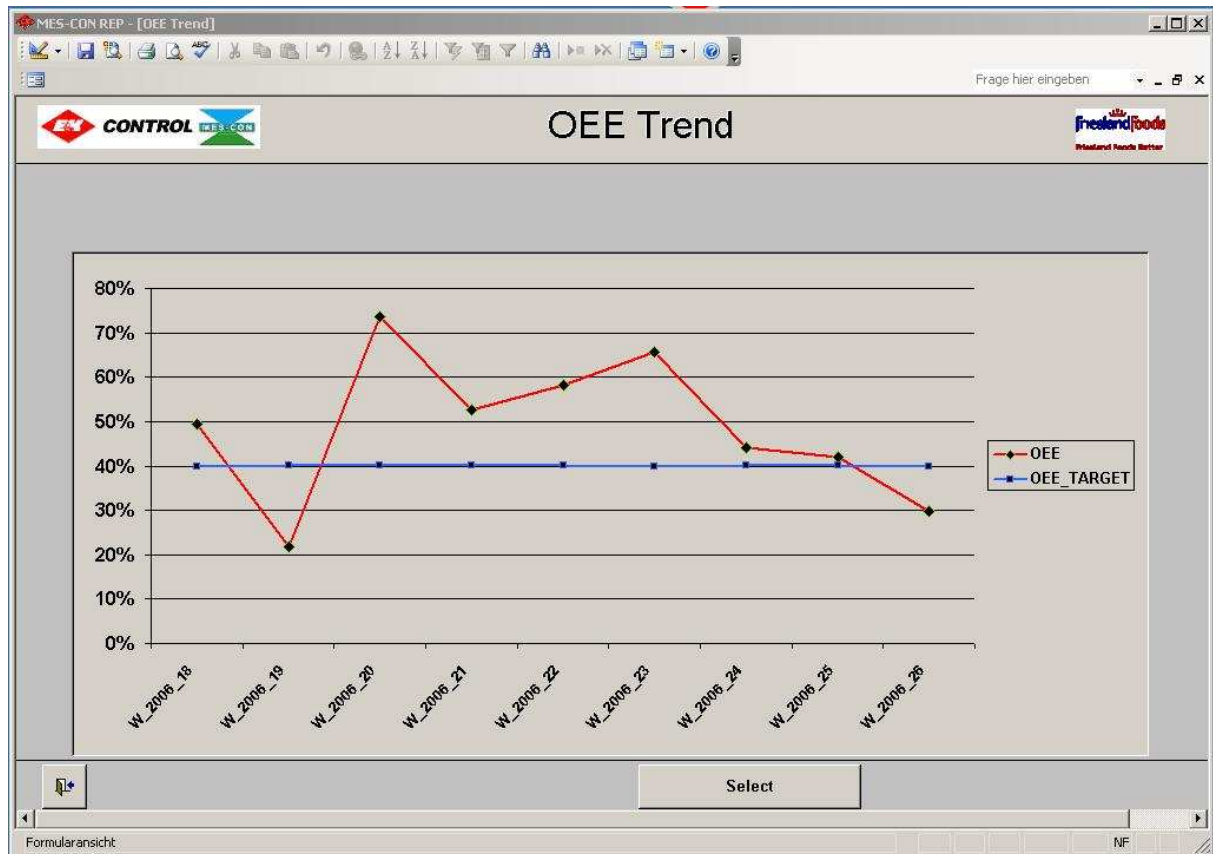


Abbildung 13: MES-CON-OS OEE (alt) - Trend

Diese Abbildung gibt einen Überblick über die grafische Trendbetrachtung der wochenweise kumulierten OEE-Werte.



### **3.2.2. Verwendete Nutzdaten**

In diesem Abschnitt werden die für den Bereich OEE verwendeten Daten vorgestellt und erläutert.

#### **3.2.2.1. Herkunft**

Bei den für diese Diplomarbeit zur Verfügung gestellten Daten handelt es sich um reale Produktionsdaten der Firma Friesland Foods in den Niederlanden. Bei der Produktionsumgebung handelt es sich um eine Maschine, die Butter herstellt. Hierbei sind verschiedene Butterprodukte möglich, die in verschiedenen Größen abgepackt sein können. Am Ende der Maschine befinden sich zwei Zähler. Der erste Zähler zählt den gesamten Ausstoß der Buttermaschine. Mit dem zweiten Zähler werden alle Einheiten gezählt, welche nach der Qualitätskontrolle noch weiter verwendet werden können. Somit findet zwischen Zähler 1 und 2 eine Qualitätskontrolle statt, welche für diese Diplomarbeit keine weitere Rolle spielt.

Die Daten werden halbautomatisch erstellt. Sowohl die Produktionsmaschine als auch ein Bediener füllen somit die Datenbank mit Meldungen über den jeweiligen Anlagenstatus.

### 3.2.2.2. Struktur und Inhalt

Die Produktionsdaten liegen in Form von Tabellen in einer SQL-Datenbank vor. Die wichtigste Tabelle ist T\_OEE\_MES\_CON\_TELE. Hier liegen alle Meldungen vor, die nach der Datenübertragung zur Verfügung stehen. In der nachfolgenden Tabelle wird ein Ausschnitt der wichtigsten Datenfelder dargestellt.

Name	Daten-Typ	Erklärung
ID	Integer	ID des Telegramms
EventTime	Datetime	Zeitstempel des Events (Ende)
Modul_ID	Smallint	Modul_ID
Modul_Sub_ID	Smallint	Modul_Sub_ID
Source	Smallint	Nummer der Quell PLC
Object	Smallint	Nummer der Maschine innerhalb der PLC
Tagname	Nvarchar(64)	Zusammensetzung der vier vorangegangenen Felder
Par_01	Float	OEE Status (Fremdschlüssel T_OEE_STATUS)
Par_02	Float	Nummer der aktuellen Charge
Par_03	Float	Produktnummer (Fremdschlüssel T_OEE_PRODUCT)
Par_04	Float	Zähler 1 (Total, vor Qualitätskontrolle)
Par_05	Float	Zähler 2 (Nach Qualitätskontrolle)
Par_10	Float	Zeitstempel des Events (Beginn), Tag
Par_11	Float	Zeitstempel des Events (Beginn), Monat
Par_12	Float	Zeitstempel des Events (Beginn), Jahr
Par_13	Float	Zeitstempel des Events (Beginn), Stunde
Par_14	Float	Zeitstempel des Events (Beginn), Minute
Par_15	Float	Zeitstempel des Events (Beginn), Sekunde
Par_16	Float	Zeitstempel des Events (Beginn), Millisekunde

**Tabelle 10 : Ausschnitt aus der Struktur der Tabelle T\_OEE\_MES\_CON\_TELE**

In der Datenbank befinden sich noch weitere Tabellen, in denen die benötigten Klartextdefinitionen für verwendete ID's stehen. Hier werden beispielsweise Meldungsnummern klare Begriffe zugeordnet. In einer Meldung steht beispielsweise im Feld Par\_01 die Zahl 50.

Über die Beziehung zur Tabelle T\_OEE\_STATUS wird der Zahl 50 der Klartext „Produziert“ (jedoch auch holländisch) zugewiesen. Hier werden ebenfalls die OEE-Meldungen in verschiedene Bereiche eingeteilt. Jede OEE-Meldung gehört in eine





Kategorie, die später einem bestimmten Wert in der Berechnung der OEE-Werte zugewiesen werden kann.

Diese Gruppierung findet sich in der Tabelle T\_OEE\_GROUP.

In der Tabelle T\_OEE\_PRODUCT finden sich Kennzahlen, die die Sollgeschwindigkeit, Packungseinheiten und Namen für ein bestimmtes Produkt vorlegen.

Somit liegen alle für die Berechnung benötigten Informationen in der Datenbank.

### **3.2.3. Umsetzung**

In den folgenden Abschnitten wird die Umsetzung des Analysetools dargestellt. Die Beschreibung der Umsetzung teilt sich analog in beide Bestandteile des Analysetools auf. Im Abschnitt 3.2.3.1 wird die Umsetzung des Web Service MES-CON-OS COM beschrieben. Der Abschnitt 3.2.3.2 beschreibt die Umsetzung der Webseite MES-CON-OS OEE.

#### **3.2.3.1. Web Service MES-CON-OS WS**

Der Web Service MES-CON-OS WS ist als eine Schnittstelle zwischen Datenbank und Webseite anzusehen. Sämtliche relevanten Datenbankabfragen zum Erhalt der erforderlichen Werte für die OEE Berechnung. Dies hat den Vorteil, dass eventuelle Änderungen nur an dieser Stelle zu beachten sind. Für die Umsetzung der drei geforderten OEE Abfragen sind somit drei Hauptmethoden erforderlich. Diese Methoden sind in Abschnitt 3.2.3.1.3 beschrieben. Außerdem werden noch ein paar Hilfsmethoden zur Ermittlung und Berechnung der Werte benötigt.

### 3.2.3.1.1. Klassendiagramm

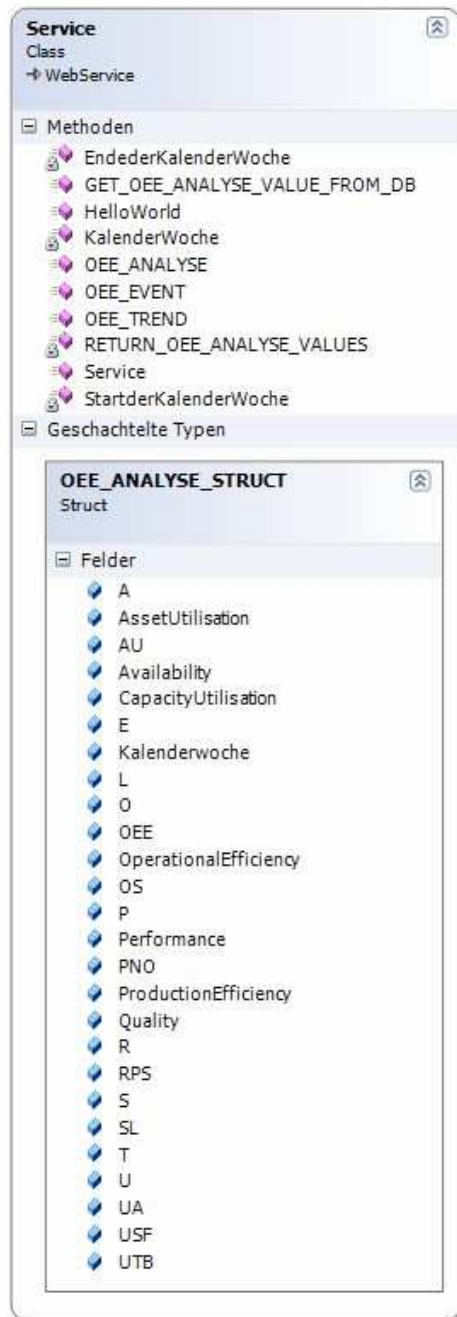


Abbildung 14: Klassendiagramm MES-CON-OS WS

### 3.2.3.1.2. Service.asmx

```
<%@ WebService Language="C#" CodeBehind="~/App_Code/Service.cs"
Class="Service" %>
```

Dies ist der einzige Inhalt der Webseite Service.asmx. Hier wird lediglich auf die Codebehind Datei „Service.cs“ verwiesen, in welcher die gesamte Funktionalität programmiert ist.

### 3.2.3.1.3. Service.cs

Name	Beschreibung
String HelloWorld()	Visual Studio Testfunktion, gibt den bekannten „Hello World!“ String aus
System.Data.DataSet OEE_EVENT(DateTime start, DateTime end)	Gibt die OEE Events als DataSet zurück.
OEE_ANALYSE_STRUCT[] OEE_TREND(DateTime start, DateTime end)	Liefert ein Array des OEE-ANALYSE_STRUCT mit den Werten der Trendabfrage.
OEE_ANALYSE_STRUCT OEE_ANALYSE(DateTime start, DateTime end)	Liefert das OEE_ANALYSE_STRUCT mit den Werten der Analyse eines Zeitraums. Ruft dazu RETURN_OEE_ANALYSE_VALUES() auf.
RETURN_OEE_ANALYSE_VALUES(start, end)	Ruft GET_OEE_ANALYSE_VALUE_FROM_DB() auf und berechnet aus den Ergebnissen die einzelnen OEE-Werte für die Analyse
System.Data.DataSet GET_OEE_ANALYSE_VALUE_FROM_DB(DateTime start, DateTime end)	Fragt die Werte aus der Datenbank ab.
DateTime StartderKalenderWoche(DateTime start)	Hilfsfunktion: Berechnet den Montag einer Kalenderwoche
DateTime EndederKalenderWoche(DateTime start)	Hilfsfunktion: Berechnet den Sonntag einer Kalenderwoche

**Tabelle 11 : Steuerelemente der Webseite „Default.aspx“**

Des Weiteren ist in der Datei „Service.cs“ eine Struktur angelegt, in der alle für den Bereich OEE benötigten Werte abgespeichert werden können. Die Variablen dieser Struktur sind der Abbildung 14: Klassendiagramm MES-CON-OS WS zu entnehmen.

Diese Struktur ist öffentlich deklariert und kann in einem Programm, welches diesen Web Service verwendet, ebenfalls benutzt werden.

Des Weiteren kann man im Web Service mit dem Kommentar „[WebMethod]“ festlegen, ob eine Methode öffentlich über den Web Service zugänglich ist, oder ob sie nur privat innerhalb des Web Service verwendet werden kann.

In diesem Fall sind drei Methoden (vier mit der „Hello World!“ Testfunktion) öffentlich:

- OEE\_EVENT()
- OEE\_ANALYSE()
- OEE\_TREND()

Mit diesen drei Funktionen sind alle geforderten OEE Abfragen abgedeckt.

### **3.2.3.2. Webseite MES-CON-OS OEE**

In diesem Abschnitt wird die Webseite MES-CON-OS OEE beschrieben. Sie teilt sich in zwei Bereiche auf. Die Datei „Default.aspx“ bildet die eigentliche Webseite, in der das Layout und die Steuerelemente festgelegt sind. ASP.NET Seiten besitzen auch eine Codebehind Datei, in der Methoden für die Funktionalität der Webseite programmiert werden können. Die Webseite verwendet die im Web Service vorhandenen offen deklarierten Methoden und Variablen.

#### **3.2.3.2.1. Seitenaufbau in HTML**

Die Datei „Default.aspx“ beinhaltet sämtlichen HTML und ASP Code. Hier wird das Layout der Seite bestimmt sowie alle wichtigen Steuerelemente hinzugefügt. Diese Steuerelemente können dann in der Codebehind Datei verwendet werden.

Folgende Steuerelemente befinden sich in der Datei „Default.aspx“:

<b>Name</b>	<b>Beschreibung</b>
diverse Label	Diverse Label sind auf der Webseite platziert, um Beschriftungen und Werte zu übernehmen.
GridView1	Das Gridview ist das Tabellen-Steuerelement auf der Webseite. Es ermöglicht die Anbindung eines Dataset, wie es durch den Web Service bereitgestellt wird.
diverse Buttons	Die Webseite enthält drei Buttons, über die die Anwahl der verschiedenen OEE Bereiche möglich ist.
Chart1	dotnetcharting Diagramm zur Anzeige der Analysewerte
Chart2	dotnetcharting Diagramm zur Anzeige der Trenddaten
Calender_Start	Kalender Steuerelement zur Auswahl des Startdatums.
Calender_End	Kalender Steuerelement zur Auswahl des Enddatums.

**Tabelle 12 : Steuerelemente der Webseite „Default.aspx“**

### 3.2.3.2.2. Codebehind C#

In der Codebehind Datei „Default.aspx.cs“ befinden sich alle wichtigen Methoden für die Funktionalität der Seite. Neben den durch die Webseite „Default.aspx“ bereits vorhandenen Steuerelementen werden noch zwei weitere Objekte instanziiert. Dabei handelt es sich jeweils um Objekte des Typs OEE\_ANALYSE\_STRUCT, welche durch den Web Service vorgegeben ist. Das erste Objekt umfasst die Struktur mit den OEE-Werten für einen ausgewählten Zeitraum. Das zweite Objekt stellt ein Array dieser Struktur dar, welches ebenfalls OEE-Werte enthält, jedoch wochenweise für den ausgewählten Zeitraum. Somit steht für jede Kalenderwoche des Zeitraums ein Arrayelement zur Verfügung. In der folgenden Tabelle befinden sich alle wichtigen Methoden der Codebehind Datei der Webseite.

Name	Beschreibung
Page_Load	Vorbelegung der Steuerelemente beim Laden der Webseite (Elemente sichtbar / unsichtbar, Setzen der Beschriftungen).
btn_OEE_Events_Click	Eventhandler: OEE Eventansicht wird aktiviert, Gridview für die Tabellenansicht wird sichtbar, Daten werden gebunden.
btn_OEE_Analyse_Click	Eventhandler: OEE Analyseansicht wird aktiviert, die Grafik wird aktiviert und durch die Methode getData() mit Werten versorgt. Die Labels werden mit den OEE Werten belegt.
SeriesCollection getData(OEE_ANALYSE_STRUCT Analyse_Struct1)	Erzeugt eine SeriesCollection für das Diagramm. Die Werte erhält diese Methode aus dem OEE_ANALYSE_STRUCT
GridView1_PageIndexChanged	Eventhandler: Lädt eine neue Seite des Gridviews bei verwenden der Navigationselemente am Gridview
btn_OEE_Trend_Click	Eventhandler: Aktiviert die OEE Trendgrafik. Die Wertebelegung wird durch die Methode getTrendData() vorgenommen
SeriesCollection getTrendData(OEE_ANALYSE_STRUCT[] trendstruct1)	Erzeugt eine SeriesCollection für das Trenddiagramm.

**Tabelle 13 : Methoden der Codebehind Klasse \_Default**

### **3.2.3.2.3. Web.Config**

ASP.NET Webseiten besitzen eine eigene Konfigurationsdatei. Hier ist darauf zu achten, dass in dieser Datei die Adresse des Web Service eingebunden wird. Normalerweise geschieht dies bei Erstellung des Projektes innerhalb des Visual Studio automatisch. Nachfolgend ist der Ausschnitt aus der Web.Config dargestellt, an dem der Web Service eingebunden ist.

```
<appSettings><add key="MES_CON_OS_WS.Service"  
value="http://localhost:2006/MES-CON-OS WS/Service.asmx"/></appSettings>
```

Nun ist die Adresse des Web Service bekannt. Sollte der Web Service einmal den Server wechseln, muss an dieser Stelle die Adresse des Web Service verändert werden, damit er weiter verwendet werden kann.



### 3.2.3.2.4. Screenshots und Funktionsbeschreibungen



Abbildung 15: MES-CON-OS OEE Startbildschirm

Bei dieser Abbildung handelt es sich um den Startbildschirm der Webseite von MES-CON-OS OEE. Der Benutzer hat die Möglichkeit, über die Kalender-Steuer-elemente ein Start- und ein Enddatum für den zu betrachtenden Zeitraum auszuwählen. Danach gibt es über die drei Button-Steuer-elemente die Möglichkeit, die OEE Events, die OEE Analyse oder den OEE Trend zu betrachten.

Stoptime	Starttime	Duration	Status ID	Status	Group	Groupname	Charge	Product ID	Productname	Total	Good
26.06.2006 11:00:00	26.06.2006 10:56:57	00:03:02	50	_Produceert	P	Production time [P]	407997	10202	Milli 40*200 (70%)	34,00	16,00
26.06.2006 10:56:57	26.06.2006 10:49:50	00:07:07	152	Storing Inpakker	UTB	Unexpected technical Breakdowns	407997	10202	Milli 40*200 (70%)	0,00	0,00
26.06.2006 10:47:52	26.06.2006 10:42:10	00:05:41	164	Wegbrengen volle pallet	USF	Unexpected supply failures	407997	10202	Milli 40*200 (70%)	0,00	0,00
26.06.2006 10:42:10	26.06.2006 10:41:55	00:00:14	50	_Produceert	P	Production time [P]	407997	10202	Milli 40*200 (70%)	0,00	0,00
26.06.2006 10:41:55	26.06.2006 10:28:51	00:13:04	118	Ombouw inpakker	RPS	Routine Production stoppages	407997	10202	Milli 40*200 (70%)	0,00	0,00
26.06.2006 10:28:51	26.06.2006 10:17:52	00:10:59	21	Einde Productie	AU	Available unused time	407996	10206	Milli 16*200 (70%)	0,00	0,00
26.06.2006 10:17:52	26.06.2006 10:17:21	00:00:30	115	Inpakker - leegmaken machine	USF	Unexpected supply failures	407996	10206	Milli 16*200 (70%)	0,00	0,00
26.06.2006 10:17:21	26.06.2006 10:11:32	00:05:49	50	_Produceert	P	Production time [P]	407996	10206	Milli 16*200 (70%)	69,60	67,20
26.06.2006 10:11:32	26.06.2006 10:10:54	00:00:37	147	Storing Wikkeltoevoer	UTB	Unexpected technical Breakdowns	407996	10206	Milli 16*200 (70%)	0,20	0,00
26.06.2006 10:10:54	26.06.2006 10:00:00	00:10:54	50	_Produceert	P	Production time [P]	407996	10206	Milli 16*200 (70%)	132,20	131,20

Abbildung 16: MES-CON-OS OEE Eventabfrage

Bei ausgewähltem Zeitraum und mit einem Klick auf den Button „OEE Events“ gelangt der Benutzer auf die tabellarische Darstellung aller aufgezeichneten OEE Events. Dargestellt sind die dazugehörigen Zeitstempel, die Dauer des Vorgangs, der zugehörige OEE Status inklusive Klartextbeschreibung sowie Chargen, Produkt und Zählerwerte. Aufgrund der hohen Datenmenge ist die Tabelle mit einem Navigationselement ausgestattet, mit der es möglich ist, einzelne Tabellenabschnitte anzuwählen.

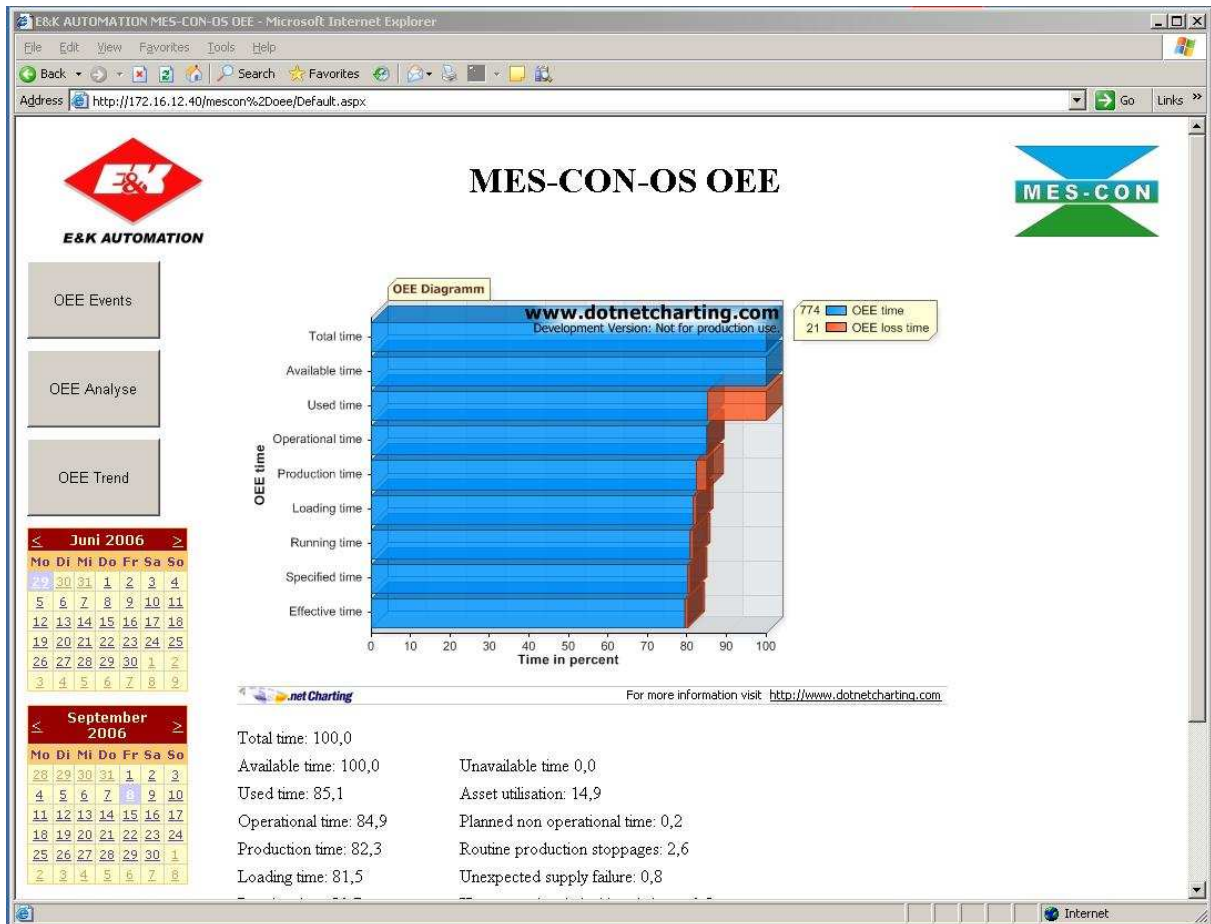


Abbildung 17: MES-CON-OS OEE Analyse

Mit der Auswahl „OEE Analyse“ gelangt der Benutzer bei ausgewähltem Zeitraum auf den Analysebereich. Hier werden anhand der in der Datenbank vorhandenen OEE Events alle wichtige Anlagenwerte fertig berechnet dargestellt, unter anderem sämtliche Zeiten (Stillstandszeiten, Rüstzeiten), Produktivitäts- und Qualitätswerte sowie der OEE-Wert. Zusätzlich werden alle Zeiten in einem Balkendiagramm dargestellt.

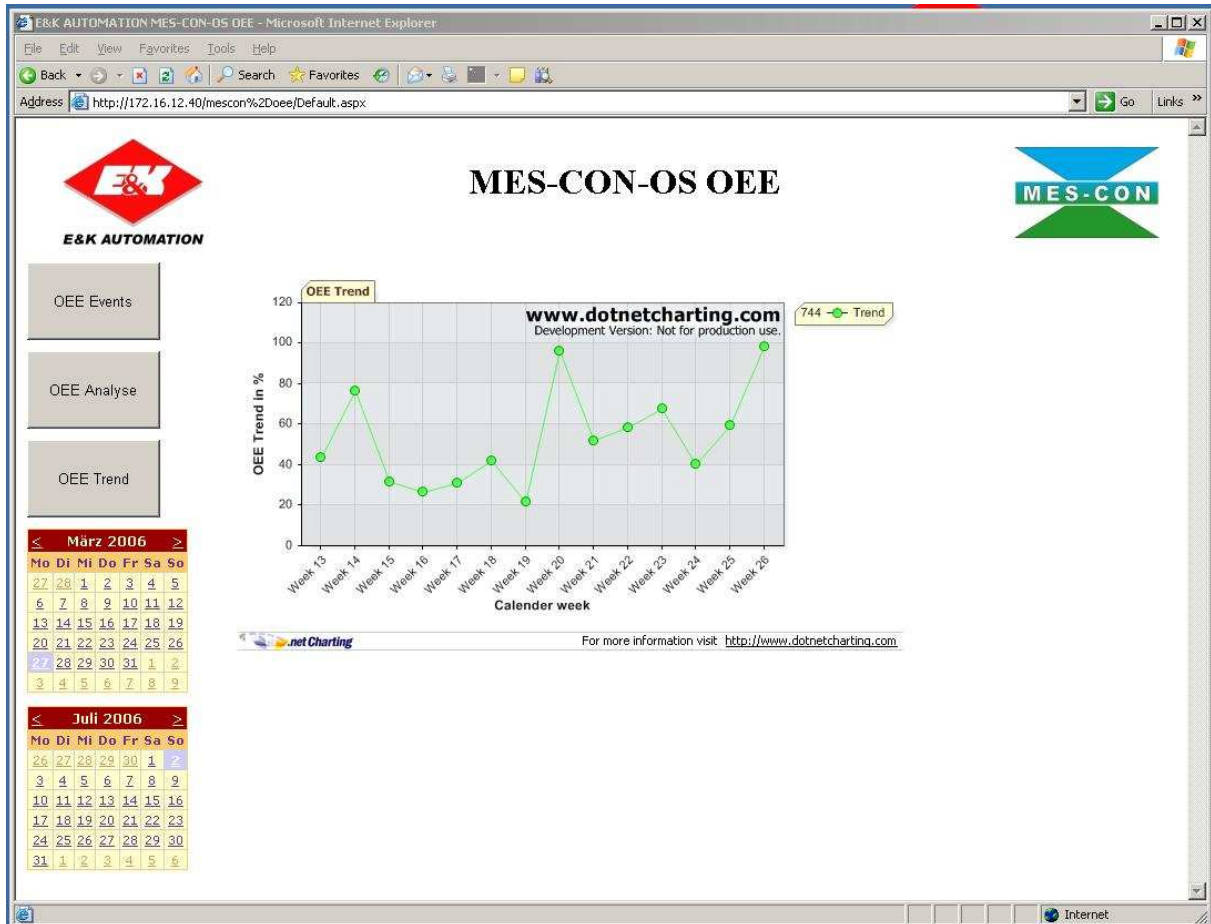


Abbildung 18: MES-CON-OS OEE Trendbetrachtung

Im Bereich „OEE-Trend“ werden grafisch die wochenweise kumulierten OEE-Werte dargestellt. Für den ausgewählten Bereich wird für jede Woche der OEE-Wert berechnet und in das Diagramm eingefügt. Somit lässt sich anhand dieser Grafik der Trend des OEE-Wertes der Anlage / Maschine erkennen.

## **4. Vergleich MES-CON-OS und Wonderware**

Ein Vergleich des neuen „Datenaustausch- und Analysetool“ mit anderen auf dem Markt befindlichen Programmen gestaltet sich als äußerst schwierig. Für diese Art von Analyse gibt es kein Standardprogramm, mit dem alle Funktionen abgedeckt sind.

Am ehesten lässt sich das neue Softwarekonzept mit Produkten der Firma Wonderware vergleichen. Hierbei sei jedoch angemerkt, dass diese Programme untereinander nicht kompatibel sind.

Bei dem nachfolgenden Vergleich geht es lediglich um die Software- und Entwicklungskosten. Eine Ausarbeitung der Hardware war für diese Arbeit nicht erforderlich und wird somit nicht weiter betrachtet.

Für eine Verbindung zwischen einer Steuerung und einem Server bietet die Firma Wonderware lediglich den I/O Server. Mit ihm ist auch der Zugriff auf eine Datenbank möglich. Die Kosten für eine Lizenz liegt bei 750,00€. Somit entspricht dies in etwa der Funktionalität des Datenaustauschtools MES-CON-OS COM.

Die Firma Wonderware bietet derzeit keinen direkten Vergleich zum Analysetool (MES-CON-OS WS / OEE). Der Industrial Application Server (IAS) lässt sich mit dem Modul Downtime-Analyst(DT) kombinieren. Dabei muss eine Anzahl von sog. Down Time Definitions (DTD) berücksichtigt werden, also eine Anzahl von möglichen Fehlermeldungen. Eine Lizenz für diese Programm kostet 18150,00€.

Die Geschäftsleitung hat für die neu erstellten MES-CON-OS Tools kosten von 8100,00€ angesetzt. Diese beinhalten sowohl die Lizenzkosten für die Programme MES-CON-OS COM / WS / OEE als auch eine SQL Lizenz.

Es ergibt sich eine Kostenübersicht für die Software wie folgt:

<u>Software</u>	<u>E&amp;K (€)</u>	<u>Wonderware (€)</u>
1. Datenaustausch	1600,00	750,00
2. Datenanalyse	6500,00	18150,00
<b>Summe:</b>	<b>8100,00</b>	<b>18900,00</b>

**Tabelle 14 : Vergleich – Softwarekosten (E&K / Wonderware)**



Somit beläuft sich die Differenz beider Softwareprodukte auf 10800€.

Gegenzustellen sind die Entwicklungskosten der neuen Softwaretools:

Der Entwicklungsaufwand dieser Diplomarbeit wird mit 600 Stunden zu je 20€ angesetzt. Noch zu leistende Entwicklungsarbeit wird mit 200 Stunden zu je 20€ angesetzt. Der Entwicklungsaufwand vor dieser Diplomarbeit wird global mit 50000€ geschätzt.

<u>Software</u>	<u>E&amp;K (€)</u>
1. Entwicklungskosten Diplomarbeit (600h*20€)	12000,00
2. Zusätzliche Entwicklungskosten (200h*20€)	4000,00
3. Bereits geleistete Entwicklungsarbeit	50000,00
<b>Summe:</b>	<b><u>66000,00</u></b>

**Tabelle 15 : Entwicklungskosten**

Bei einer Preisdifferenz von 10800€ und einer Entwicklungssumme von 66000€ ergibt sich:  $66000€ / 10800€ = 6,11$

Demnach müssten 7 Installation bestehend aus den neuen Komponenten MES-CON-OS COM / WS / OEE verkauft werden.

Allerdings ist diese Rechnung stark idealisiert, da der Bereich OEE für diese Diplomarbeit eine Führungsgröße darstellte. Der größte Nutzen liegt in der Verwendung der neuen Webtechnologien. Hierdurch entstehen vor allem geringere Kosten im Zuge von Updates. Durch die Webtechnologien sind alle nötigen Funktionen zentral programmiert. Somit können an dieser Stelle Änderungen vorgenommen werden. Ein Verteilen von Updates auf Clients ist somit nicht nötig. Dies bedeutet eine enorme Kostenersparnis, die zu diesem Zeitpunkt nicht geschätzt werden kann.

## **5. Zusammenfassung / Ausblick / Bewertung**

Abschließend lässt sich sagen, dass die Abwicklung dieser Diplomarbeit nicht reibungslos aber erfolgreich verlief.

Die Anwendung neuer Technologien und Verfahren birgt immer gewisse Risiken. Mögliche Fehlerquellen können zwar geschätzt werden, in der Realität sind diese Schätzungen aber meist nicht mehr korrekt. Mögliche eingeplante Probleme zeigen sich nicht als Probleme auf, an anderen Stellen entwickeln sich leichte Problemstellungen zu unüberwindbaren Hindernissen, die im Rahmen einer Diplomarbeit nicht komplett zu erörtern sind.

Die Zusammenarbeit mit dem Unternehmen E&K Automation gestaltete sich sehr angenehm. Auch wenn auf technischer Seite kaum Ansprechpartner und Hilfestellungen zur Verfügung standen, so stand auf persönlicher und fachübergreifender Ebene jede Tür offen.

Die Funktionalität der Programme ist, wie sie im Pflichtenheft gefordert war, erfüllt:

- MES-CON-OS COM läuft als Windowsdienst im Hintergrund des Betriebssystems. Ein Konfigurationstool hilft bei der Erstellung der Verbindungen. Die maximale Verbindungsanzahl ist durch die physikalische Begrenzung der Ports gegeben. Sämtliche Funktionalitäten sind auf den Rechnern der CONLOG-AG getestet worden. Die Telegramme aus der Abfüllanlage der CONLOG-AG werden in die zugehörige Datenbank geschrieben.
- Die OEE-Daten der Firma Friesland Foods werden durch die neue Webtechnologie bestehend aus MES-CON-OS WS und OEE so ausgewertet, wie sie auf dem alten MS Access - Frontend zu sehen sind. Jedoch wird für die Darstellung nur noch der MS Internet Explorer benötigt. Es werden die Bereiche OEE Event / Analyse / Trend abgedeckt. Beide Softwaretools befinden sich ebenfalls zu Testzwecken auf den Rechnern der CONLOG-AG, jedoch spielt die eigentliche Funktionalität der CONLOG-AG keine Rolle, es wird nur die vorhandene Datenbank für die Daten der Firma Friesland Foods sowie der IIS benötigt.

Eine gewünschte zusätzliche Funktion ließ sich im Rahmen dieser Arbeit nicht umsetzen. Optional erwünscht war die Möglichkeit, MES-CON-OS COM nicht nur passiv auf Verbindungseingänge warten zu lassen, sondern diese auch aktiv herzustellen. Nach erfolgreicher Verbindung und im Zuge der Kommunikation mit dem Verbindungsteilnehmer stieg jedoch die CPU-Auslastung auf 100%, so dass der Computer ausgelastet war. Es ist nicht zu erkennen, ob es sich um ein programmiertechnisches Problem oder um einen möglichen Fehler in der Implementierung der Netzwerkfunktionalität im .NET Framework handelt. Eine weitere Untersuchung dieses Problems konnte im Rahmen dieser Ausarbeitung nicht mehr stattfinden.

Das Datenaustauschtool MES-CON-OS COM befindet sich mit minimalen Anpassungen bereits im produktiven Einsatz in einem Lagersystem der Firma Antolin. Dort sorgt es dafür, dass Daten eines Barcodelesegerätes auf MS Windows CE Basis per WLAN in eine SQL-Datenbank geschrieben werden.

In Gesprächen mit der Geschäftsleitung des Unternehmens E&K Automation stellte sich heraus, dass diese Diplomarbeit einen hohen Stellenwert für das Unternehmen hat. Die Verwendung von Webtechnologien bedeutet eine Verbesserung für die Produkte des Unternehmens und somit auch eine bessere Wettbewerbsfähigkeit gegenüber anderen Unternehmen.

Die Grundsteinlegung für die Vereinheitlichung der Programmierumgebungen und für die Einführung der neuen Webtechnologien ist vollzogen. Nun liegt es in der Verantwortung des Unternehmens, diese Technologien fortzuführen.

Weitere Anwendungsmöglichkeiten der neuen Technologien:

- Auslagerung von Datenbankabfragen und Programmlogik anderer Unternehmenssoftware in Web Services
- Umsetzung von GUI Oberflächen auf ASP.NET Technologie

Somit können viele Softwaretools angefangen bei der Lagerverwaltung über Reportingprogramme für Fahrerlose Transportsysteme auf eine moderne Umgebung umgesetzt werden. Die Möglichkeiten durch die neue Technologie sind sehr vielfältig.



## **6. Quellen**

### **6.1. Literaturverzeichnis**

- E&K Firmenpräsentation, Stand 07.2006
- E&K Funktionsbeschreibungen zu MES-CON PLC (FI 05-07-01)
- Pflichtenheft - Diplomarbeit Stefan Boehning
- Jeffrey Richter (2005) *Microsoft .NET Framework-Programmierung in C#* Microsoft Press
- Makofske/Donahoo/Calvert (2004) *TCP/IP Sockets in C#* Morgan Kaufmann
- Charles Petzold (2005) *Windows Forms-Programmierung mit Visual C# 2005* Microsoft Press
- Dusan Petkovic (2001) *Microsoft SQL Server 2000* Addison-Wesley
- John Sharp (2005) *Visual C# 2005 Schritt für Schritt* Microsoft Press

## **6.2. Tabellenverzeichnis**

Tabelle 1 : Abkürzungen.....	11
Tabelle 2 : Struktur der XML-Konfigurationsdatei .....	29
Tabelle 3: Aufbau Telegrammkopf.....	30
Tabelle 4 : Felder der Klasse cls_Config .....	34
Tabelle 5 : Methoden der Klasse frm_MesconComController .....	35
Tabelle 6 : Methoden der Klasse „MesconComServer“ .....	39
Tabelle 7 : Methoden der Klasse „cls_Communication“ .....	40
Tabelle 8 : Methoden der Klasse „cls_Telegrammserver“ .....	41
Tabelle 9 : Methoden der Klasse „cls_Communication“ .....	41
Tabelle 10 : Ausschnitt aus der Struktur der Tabelle T_OEE_MES_CON_TELE.....	48
Tabelle 11 : Steuerelemente der Webseite „Default.aspx“ .....	52
Tabelle 12 : Steuerelemente der Webseite „Default.aspx“ .....	54
Tabelle 13 : Methoden der Codebehind Klasse _Default .....	55
Tabelle 14 : Vergleich – Softwarekosten (E&K / Wonderware).....	61
Tabelle 15 : Entwicklungskosten .....	62

### **6.3. Abbildungsverzeichnis**

Abbildung 1: Zusammenspiel der MES-CON Komponenten .....	9
Abbildung 2: Legende.....	10
Abbildung 3: Simatic Pyramide – Vertikale Integration .....	12
Abbildung 4: Balkendiagramm aus der Beispielsammlung von dotnetcharting2005.	26
Abbildung 5: Ausschnitt zum Bereich Datenaustausch .....	28
Abbildung 6: Klassendiagramm cls_Telegramm.....	31
Abbildung 7: Klassendiagramm Konfigurationstool für MES-CON-OS COM.....	33
Abbildung 8: Screenshot des Konfigurationstools .....	36
Abbildung 9: Klassendiagramm MES-CON-OS COM .....	38
Abbildung 10: Ausschnitt zum Bereich Datenanalyse .....	43
Abbildung 11: MES-CON-OS OEE (alt) - Events.....	44
Abbildung 12: MES-CON-OS OEE (alt) - Analyse.....	45
Abbildung 13: MES-CON-OS OEE (alt) - Trend .....	46
Abbildung 14: Klassendiagramm MES-CON-OS WS .....	51
Abbildung 15: MES-CON-OS OEE Startbildschirm .....	57
Abbildung 16: MES-CON-OS OEE Eventabfrage.....	58
Abbildung 17: MES-CON-OS OEE Analyse .....	59
Abbildung 18: MES-CON-OS OEE Trendbetrachtung.....	60



## **7. Erklärung zur Diplomarbeit**

Name : Boehning  
Vorname : Stefan  
Matr.-Nr. : 1155324  
Studiengang : Ingenieur Informatik

An den Prüfungsausschuss  
des Fachbereichs Automatisierungstechnik  
der Universität Lüneburg  
Volgershall 1

21339 Lüneburg

### **Erklärung zur Diplomarbeit**

Ich versichere, dass ich diese Diplomarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Schwarzenbek, den 11. September 2006

---

Stefan Boehning

## **8. Anhang**

Der komplette Anhang der Diplomarbeit befindet sich auf der beiliegenden CD.

### **8.1. Ablageort der Diplomarbeit**

<b><u>Bestandteil</u></b>	<b><u>Verzeichnis auf der CD</u></b>
Diplomarbeit Stefan Boehning.doc	[CD]:\Diplomarbeit\
Diplomarbeit Stefan Boehning.pdf	[CD]:\Diplomarbeit\
Pflichtenheft.doc	[CD]:\Diplomarbeit\

### **8.2. Ablageort der Visual Studio - Projekte**

<b><u>Bestandteil</u></b>	<b><u>Verzeichnis auf der CD</u></b>
Projekt MES-CON-OS COM	[CD]:\MES-CON-OS COM\
Projekt MES-CON-OS WS	[CD]:\MES-CON-OS WS\
Projekt MES-CON-OS OEE	[CD]:\MES-CON-OS OEE\