



D I P L O M A R B E I T

Einbindung von automatisiert extrahierten Webdaten in Internet Portalen

Am Beispiel des SAP Enterprise Portals
und der Lixto - Technologie

Ausgeführt am

Institut für Informationssysteme (184/2)
Abteilung für Datenbanken und Artificial Intelligence

der Technischen Universität Wien

unter der Anleitung von

Univ. Prof. Dipl.-Ing. Dr. Georg Gottlob
und

Univ. Ass. Dipl.-Ing. Dr. Marcus Herzog

als verantwortlich mitwirkendem Universitätsassistenten

durch

Lukas Steiner
Liebhartstalstraße 33 C
1160 Wien

Danksagung

Diese Arbeit ist ganz speziell meiner Familie und meinen Eltern gewidmet, welche mich während meiner Ausbildung vorbehaltlos unterstützten und mir den Abschluss dieses Studiums überhaupt erst ermöglichten.

Mein besonderer Dank gilt auch Univ. Ass. Dipl.-Ing. Dr. Marcus Herzog für die tatkräftige Unterstützung bei der Ausarbeitung der Diplomarbeit und für viele wertvolle Hinweise, Anregungen und fachliche Diskussionen.

Weiters möchte ich mich bei den Mitarbeitern des EC3 für die gute Zusammenarbeit, sowie für rasche Hilfestellungen während der Projektrealisierung recht herzlich bedanken.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Wien, Oktober 2008

Kurzfassung

In der heutigen Zeit wird es für große Unternehmen, die eine Vielzahl an verschiedenen Arbeitsprozessen zu koordinieren haben, immer wichtiger in jedem Segment ihres Aufgabenbereiches eine möglichst effiziente Softwarelösung zu haben. Um jedoch nur wenige Ressourcen bei der Benutzung dieser Programme zu verbrauchen, werden dringend neue Technologien benötigt, die den Anwendern eine Erleichterung und Hilfestellung bei der Nutzung bieten sollen.

Diese Diplomarbeit zeigt einen realisierbaren Lösungsansatz auf, welcher nicht nur die technische Integration und Kommunikation von verschiedenen Unternehmensanwendungen berücksichtigt, sondern auch für die Benutzung dieser Applikationen einen möglichst anwenderorientierten und ressourcenschonenden Weg mit einbezieht.

Im Rahmen eines Forschungsprojektes, wurden die Voraussetzungen für eine Systemlandschaft geschaffen, welche die Verarbeitung und Darstellung von automatisiert extrahierten Informationen in einem für industrielle Zwecke geeignetem Szenario simuliert. In dieser Forschungsumgebung wurden unterschiedliche Technologien zur Systemintegration und -transformation evaluiert und anschließend in dem für die Automobilindustrie sehr wichtigen Bereich der Qualitätssicherung umgesetzt, wobei auch der Schwerpunkt auf eine systemunabhängige Darstellung gelegt wurde.

Das Projekt wurde im Forschungsverein **Electronic Commerce Competence Center – EC3** unter dem Namen „*in2ex - Case Study Integration mit SAP Portal*“ publiziert und schenkt den Technologien der Firmen **SAP** und **Lixto** besondere Berücksichtigung.



Abstract

Today big businesses have a lot of different workflows to coordinate, due to that the need of an effective software solution for each and every task is becoming more and more important. To prevent wasting human resources by many different applications, new technologies are needed to increase the usability and to force the integration aspect.

This diploma thesis points out an automotive scenario solution that not only considers the commercial aspects but also regards the technical and conceptual facets.

In the context of a research project the conditions for a system landscape were created, which simulated the processing and representation of automated extracted information in a scenario suitable for industrial purposes. In this research environment different technologies for system integration and -transformation have been evaluated and applied in the field of quality assurance, which constitutes a core business in the automobile industry. The emphasis was put on a system-independent representation.

The project was published in the research association Electronic Commerce Competence Center – EC3 under the name „in2ex - Case Study Integration mit SAP Portal“ and considers the technologies of the companies SAP and Lixto in particular.



Inhaltsverzeichnis

Danksagung	2
Eidesstattliche Erklärung	3
Kurzfassung	4
Abstract	5
Inhaltsverzeichnis	6
Abbildungsverzeichnis	9
Tabellenverzeichnis	11
1 Einleitung	12
2 Projektbeschreibung	14
2.1 Allgemeine Projektbeschreibung	14
2.2 Zielsetzung	16
2.2.1 Qualitätsprozess ohne Adaptierung.....	19
2.2.1.1 Navigation und Benutzerparameter	20
2.2.1.2 Vergleich und Informationsübernahme	20
2.2.1.3 Informationsverteilung und Rückmeldungskontrolle	21
2.2.2 Qualitätsprozess mit Lixto Software.....	23
2.2.2.1 Navigation und Benutzerparameter	23
2.2.2.2 Vergleich und Informationsübernahme	23
2.2.2.3 Informationsverteilung und Rückmeldungskontrolle	24
2.2.3 Qualitätsprozess mit Lixto Software im SAP Enterprise Portal	26
2.2.3.1 Navigation und Benutzerparameter	26
2.2.3.2 Vergleich und Informationsübernahme	27
2.2.3.3 Informationsverteilung und Rückmeldungskontrolle	27
2.2.4 Ergebnis der Aufwandsanalyse	29
2.2.5 Ausblick	30
2.3 Anwendungsarchitektur	32
3 Lixto Suite	36
3.1 Lixto Visual Wrapper.....	36
3.1.1 Visual Wrapper Systemarchitektur.....	37
3.1.2 Konzept des Visual Wrappers	38
3.1.2.1 Patterns	39

3.1.2.2	Filters	40
3.1.2.3	Conditions	41
3.1.3	VW Benutzeroberflächen (GUI)	41
3.2	Lixto Transformation Server	43
3.2.1	Konzepte des Transformation Servers	44
3.2.1.1	Role Model (Benutzerverwaltung)	45
3.2.1.2	Pipes (Workflow)	45
3.2.1.3	Components	46
3.2.2	Technologie	50
3.2.3	TS Benutzeroberflächen (GUI)	52
3.2.3.1	Transformation Server Administration Console	52
3.2.3.2	Transformation Server Workbench	53
3.2.3.3	MyLixto	55
4	SAP Enterprise Portal	58
4.1	Portaldefinition und deren Einsatzgebiete	58
4.2	Vorteile beim Einsatz eines Portals	60
4.2.1	Einstiegspunkt über Single-Sign-On Konzepte	61
4.2.2	Individuelle rollenbasierte Benutzeroberfläche	61
4.2.3	Einfacher und geringer Wartungsaufwand	61
4.2.4	Sicherer und standardisierter Portalzugriff	62
4.3	Integrationsmöglichkeiten	62
4.3.1	Enterprise Portal Standard Komponenten	62
4.3.2	SAP Branchenlösungen und Business Packages	63
4.3.3	Knowledge Management	63
4.3.4	SAP xApps	64
4.4	Enterprise Portal Content-Konzept	64
4.5	Enterprise Portal Oberfläche	66
5	Umsetzung und Realisierung	68
5.1	Anwendungsserver und deren Hardware-Ressourcen	68
5.2	Lixto Visual Wrapper Templates	69
5.3	Lixto Transformation Server Konfiguration	72
5.4	SAP Enterprise Portal Konfiguration	76
5.5	Integration des SAP R/3 Systems	81
5.5.1	Konfiguration der SAP R/3 Einbindung	81
5.5.2	iView Anwendungsfälle	82
5.5.2.1	Daten aus R/3 lesen	83

5.5.2.2	Qualitätsmeldungsliste anzeigen	83
5.5.2.3	Qualitätsmeldung anzeigen	84
5.5.2.4	Daten vom TS ins R/3 schreiben	85
5.5.3	Entwicklungsdokumentation	87
5.5.4	Mapping der Qualitätsmeldungen	89
Zusammenfassung.....		91
Literatur- und Internetquellen		93
Anhang.....		96
I	Marken.....	96
II	Source Code.....	97
II.a	iView Integration „PortalQualityNotificationList.java“	97
II.b	iView Integration „R3QualityNotificationList.java“	101
II.c	Zusatzfunktion Tooltip.....	109

Abbildungsverzeichnis

Abbildung 1: Darstellung im Quellsystem und in der Portalanwendung	15
Abbildung 2: Break-Even Punkt.....	17
Abbildung 3: Einsparungspotential	18
Abbildung 4: Qualitätsprozess ohne Adaptierung.....	22
Abbildung 5: Qualitätsprozess mit Lixto Software	25
Abbildung 6: Qualitätsprozess mit Lixto Software im SAP Enterprise Portal	28
Abbildung 7: Erweiterte Qualitätsprozessbetrachtung	31
Abbildung 8: Anwendungsarchitektur	32
Abbildung 9: Detailanwendungsfall Transformation Server	34
Abbildung 10: Detailanwendungsfall SAP Enterprise Portal.....	35
Abbildung 11: Visual Wrapper Systemarchitektur [lix08b]	38
Abbildung 12: Funktionsbausteine Wrapper Regel	39
Abbildung 13: Visual Wrapper Benutzeroberfläche	42
Abbildung 14: Visual Navigator Benutzeroberfläche [lix08h]	43
Abbildung 15: Transformation Server Input-Transaktion-Output Prozess	44
Abbildung 16: Beispiel einer Pipe.....	46
Abbildung 17: Komponentenstatus [lix08c]	49
Abbildung 18: Verbindungsmöglichkeiten der Pipe Komponenten	50
Abbildung 19: Transformation Server Architektur [lix08d]	51
Abbildung 20: Transformation Server Login	52
Abbildung 21: Transformation Server Console [lix08d]	53
Abbildung 22: Transformation Server „Workbench“	54
Abbildung 23: MyLixto „Control“	56
Abbildung 24: MyLixto „Activity“	56
Abbildung 25: MyLixto „Results“	57
Abbildung 26: „Desktop“ Konzept.....	59
Abbildung 27: „Webtop“ Konzept.....	60
Abbildung 28: Portal Content Directory	65
Abbildung 29: Enterprise Portal Oberfläche	67
Abbildung 30: Original Qualitätsmeldungsseite	70
Abbildung 31: Meldungsstruktur im Visual Wrapper	71

Abbildung 32: XML Datenstruktur.....	72
Abbildung 33: Pipe Definition	73
Abbildung 34: XSL Stylesheet	74
Abbildung 35: HTML Ausgabecode	75
Abbildung 36: Webansicht der HTML Ausgabedatei	75
Abbildung 37: Portal Informationsseite	78
Abbildung 38: Portal General Motors (nicht abonniert).....	78
Abbildung 39: Portal Lixto Admin	79
Abbildung 40: Portal Service Benutzerparameter.....	80
Abbildung 41: Portal General Motors (abonniert)	80
Abbildung 42: Portal R/3 Konfiguration	82
Abbildung 43: iView Anwendungsfalldiagramm.....	83
Abbildung 44: Qualitätsmeldungsliste	84
Abbildung 45: Qualitätsmeldung.....	85
Abbildung 46: iView Aktivitätsdiagramm.....	86
Abbildung 47: iView Klassendiagramm	88
Abbildung 48: iView Sequenzdiagramm.....	89

Tabellenverzeichnis

Tabelle 1: Einsparungspotential der Qualitätsprozessvarianten.....	29
Tabelle 2: Hardware-Ressourcen.....	69
Tabelle 3: Felddefinition Beispielqualitätsmeldung.....	71
Tabelle 4: Mapping-Tabelle der Qualitätsmeldungen.....	90

1 Einleitung

Um als Großunternehmen erfolgreich sein zu können, ist eine global definierte Strategie unabkömmlich. Jeder erfolgreiche Konzern hat eine genaue Vorstellung darüber, welche Kriterien in seiner Strategie einen hohen Stellenwert haben. Darunter fallen in den meisten Fällen der Grad der individuellen Spezialisierung, der ein Unternehmen von anderen abhebt und es einzigartig macht. Auch die optimale Ausnutzung der zur Verfügung stehenden Ressourcen ist eine Komponente, die in sehr vielen Unternehmensstrategien eine hohe Gewichtung einnimmt. Neben diesen zwei erwähnten Kriterien gibt es eine Vielzahl an weiteren möglichen Strategiebestandteilen, wie die Wahl des richtigen Standorts, der Einsatz von ausgewählten Marketinginstrumenten, die Pflege von Kundenbeziehungen und die Motivation von Mitarbeitern. Jedes Management muss die für sein Unternehmen wichtigsten Kriterien definieren, um diese anschließend nach eigenen Bedürfnissen zu prägen. Alle Komponenten einer Unternehmensstrategie müssen stetig kontrolliert und optimiert werden, um Ihren Erwartungen zu entsprechen.

Aus meiner Sicht sind nicht nur die einzelnen Kriterien der Strategie eines Unternehmens ausschlaggebend für Erfolg oder Misserfolg, sondern auch das effiziente Zusammenspiel dieser. Die Optimierung von einzelnen Bestandteilen kann direkte und indirekte Auswirkungen auf andere Komponenten der Strategie haben und somit positive Synergie-Effekte bewirken.

In dieser Diplomarbeit werden Ausschnitte der zwei Strategiekriterien „individuelle Spezialisierung“ und „Ressourcenoptimierung“ in einem industriebezogenen Anwendungsfall evaluiert. Es handelt sich dabei um den Bereich der Automobilindustrie und den Qualitätsdatenaustausch zwischen Hersteller- und Zulieferunternehmen, welche komplett unabhängige Konzerne sind. Anhand beider Kriterien wird in Form dieses Projekts ersichtlich, dass sich die Effizienz und damit auch der Unternehmenserfolg durch den Einsatz von übergreifenden Softwarelösungen steigern lassen. Erst durch das Zusammenspiel und die

indirekte Beziehung dieser Strategiekomponenten wird das maximale Steigerungspotential aufgezeigt.

Bei der Strategie der Spezialisierung wird davon ausgegangen, dass über besondere und einzigartige Lösungsansätze der Erfolg eines Unternehmens gesichert werden kann. Auch wenn es nicht immer direkt mit dem Erzeugnis zusammenhängen muss, ist dabei in fast allen Branchen der Einsatz von heterogenen Anwendungen nicht zu verhindern, insbesondere nicht bei konzernübergreifenden Kooperationen. In unserem Projekt ist es möglich, die Basisfunktionalitäten unterschiedlichen Softwarearchitekturen durch den gezielten Einsatz einer Integrationsplattform zu bündeln. Eine globale Plattform hat den Vorteil, dass sie unabhängig vom Szenario erweiterbar ist und zentral verwaltet wird.

In diesem Evaluierungsszenario wird die Rolle der zentralen Integrationsplattform von der Lixto Suite übernommen.

In den meisten Unternehmen kann davon ausgegangen werden, dass Ressourcen einen konstanten Aufwand erzeugen und damit ein fixes Budget binden. Im konkreten Forschungsprojekt wird das Arbeitszeitpotential von Mitarbeitern als Ressource definiert. Um dieses optimal nutzen zu können, müssen Tätigkeitsprozesse und Arbeitsabläufe mit zu hoher Komplexität und Fehleranfälligkeit vermieden werden. Dies kann durch die Einführung einer globalen Portalanwendung realisiert werden. Prozesszuständigkeiten können gefiltert für die verantwortlichen Mitarbeiter aufbereitet werden. Kanalisierte Informationen, die über eine Portalsystemlandschaft verteilt werden, können die Komplexität und die Fehleranzahl auf Benutzerebene minimieren.

In unserem Projekt wird das Enterprise Portal von SAP die Rolle der zentralen Portalanwendung übernehmen.

2 Projektbeschreibung

Im Rahmen eines Forschungsprojektes wurde eine Systemlandschaft erstellt, die die Verarbeitung und Darstellung von automatisiert extrahierten Informationen in einem für industrielle Zwecke geeigneten Szenario simuliert.

Die Projektarbeit wurde im Rahmen des **Electronic Commerce Competence Center – EC3**¹ evaluiert, wodurch es möglich wurde, die unterschiedlichen Technologien der Firmen **SAP** und **Lixto**, welche beide Mitglieder des EC3 sind, so erfolgreich gemeinsam einzusetzen. Teile des Projekts wurden im Forschungsverein unter dem Namen „*in2ex - Case Study Integration mit SAP Portal*“ publiziert.

2.1 Allgemeine Projektbeschreibung

Der Schwerpunkt des Projektes lag darin, den für die Automobilindustrie sehr wichtigen Bereich der Qualitätssicherung in verschiedene Teilprozesse zu gliedern, anschließend diese zu adaptieren und mit unterschiedlichen Applikationen neu aufzubereiten. Die Automotive Branche arbeitet im Bereich der Qualitätssicherung verstärkt über den Austausch von Qualitätsmeldungen; und genau bei diesen Kommunikationsverfahren sind die Vorteile der im Projekt eingesetzten Software besonders ersichtlich.

Das Projektszenario betrachtet die enge Zusammenarbeit und Kommunikation zwischen Automobilherstellern (OEM) und deren Zulieferunternehmen. Der Meldungs-austausch betrifft auf der einen Seite die Definition und Norm von Einzelteilen, welche vom OEM beim Lieferanten bestellt werden. Auf der anderen Seite ist die Rückmeldung über die Qualität der ausgelieferten Teile, wie Stückfehler oder Lieferschwierigkeiten für den Lieferanten und den OEM sehr von Bedeutung, um vereinbarte Standards der Qualitätssicherung zu halten bzw. zu

¹ Forschungsverein Electronic Commerce Competence Center – EC3, Donau-City-Straße 1, 1220 Wien

verbessern.

Der sich momentan im Einsatz befindliche Workflow setzt auf dem Konzept der manuellen Informationssammlung auf. Werden Informationen benötigt bzw. könnten sich vorhandene Informationen geändert haben, so werden diese durch dafür zuständige Mitarbeiter auf den neuesten Stand gebracht. In einem weiteren Schritt müssen die veränderten Daten an alle beteiligten Personen im betroffenen Unternehmen verteilt und deren Empfang überprüft werden. Mögliche Rückmeldungen müssen ebenfalls koordiniert verarbeitet und protokolliert werden. Die dadurch entstehenden Arbeitsschritte, wie zum Beispiel das Sammeln von nicht veränderten Informationen, die Eingabe von Identifikationsinformationen oder die Verteilung der Qualitätsdaten können beim herkömmlichen Arbeitsprozess nicht vermieden werden.

Erst der Einsatz und die Unterstützung von zusätzlicher Software, wie von Lixto oder SAP hergestellt, kann ersichtliche Vorteile bringen. Ziel ist es, heterogene Informationen von unterschiedlichen Quellsystemen so aufzubereiten, dass Mitarbeiterressourcen optimal genutzt werden können. Die folgende Abbildung (Abbildung 1) zeigt die Möglichkeit, Qualitätsmeldungen für den Anwender strukturiert aufbereitet darzustellen. Auf der linken Seite sind die Originaldaten im Quellsystem zu sehen und auf der rechten Seite sind diese bereits extrahiert und in die Portalanwendung integriert.

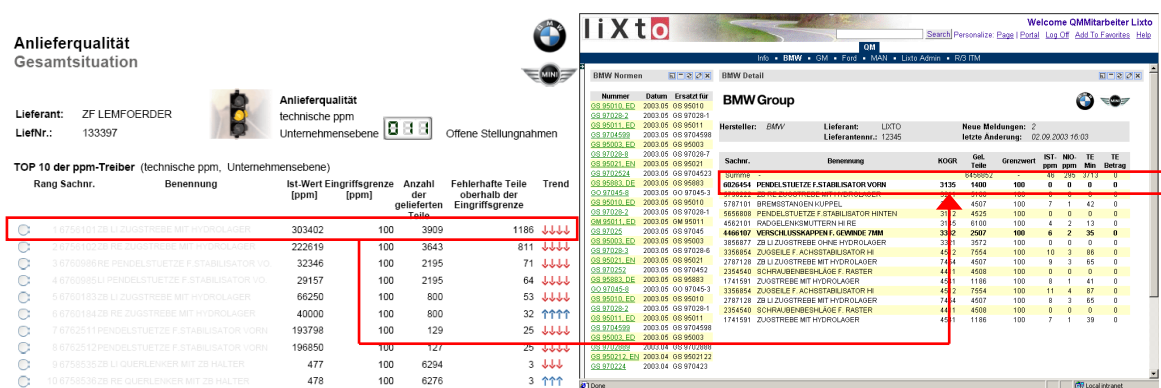


Abbildung 1: Darstellung im Quellsystem und in der Portalanwendung

Eine genaue Analyse und Abschätzung des Einsparungspotenzials ist im

nächsten Kapitel (2.2 Zielsetzung) detailliert aufgelistet.

Auf Grund der Vielzahl an unterschiedlichen Ansatzpunkten für die Projektarbeit hat man sich in dieser Phase dazu entschlossen, nur die Prozesse ausgehend vom Autobilieferanten und seine Kommunikationsstrukturen zu den Automobilherstellern zu evaluieren.

2.2 Zielsetzung

Das Projekt zeigt die Erleichterungen und Verbesserungen in der Automobilindustrie, im speziellen Bereich der Qualitätsmeldungen, welche durch den Einsatz und die Integration der neuen Systeme ersichtlich werden, auf.

Um Situationen im Umgang mit Qualitätsmeldungen zu finden, welche Verbesserungspotential beinhalten, wurde zuerst der gesamte Arbeitsprozess analysiert. Anschließend wurden Teile des Workflows verglichen, wobei einerseits die gebräuchlichen Technologie und andererseits Produkte der Firmen SAP und Lixto eingesetzt wurden. Beim Vergleich der Teilprozesse wird von bestehenden Systemlandschaften und bereits etablierten Arbeitsprozessen ausgegangen.

Um wirtschaftlich eine Aussage treffen zu können, ab wann sich der Einsatz von zusätzlichen Softwarekomponenten rentiert, kann die klassische Gegenüberstellung von Kosten und Erlös [ks96] als Basis herangezogen werden.

Grundlage für die Abbildung 2: Break-Even Punkt sind folgende Funktionen:

- **Kostenfunktion $K(x)$**

$$K(x) = k_v * x + K_f$$

- **Erlösfunktion $E(x)$**

$$E(x) = p * x$$

In den Funktionen entsprechen die Variablen x der Absatzmenge, k_v den variablen Kosten, K_f den Fixkosten und p dem Einzelpreis.

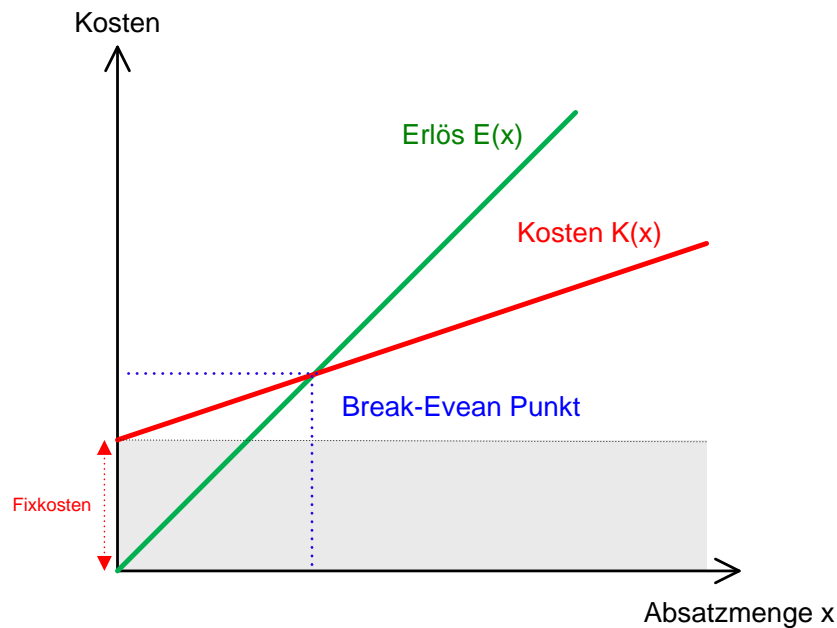


Abbildung 2: Break-Even Punkt

Als Gewinnschwelle bzw. Break-Even-Punkt² gilt, dass die jeweiligen Absatzmengen x der Kosten- und Erlösfunktion gleich groß sein müssen.

$$\begin{aligned}
 K(x) &= E(x) \\
 k_v \cdot x + K_f &= p \cdot x \\
 x &= K_f / (p - k_v)
 \end{aligned}$$

Die beschriebenen Funktionen können jetzt für das geplante Einsatzszenario der Qualitätsmeldungen übernommen und geringfügig adaptiert werden. x kann als die Anzahl an Meldungsinteraktionen, p als Einsparungspotential und k_v als Parameter für die Komplexität der Informationsstruktur interpretiert werden. Die Fixkosten K_f bestehen aus den initialen Anschaffungskosten für die Implementierung der neuen Softwarelösung.

Ziel ist es, eine aussagekräftige Angabe für das Einsparungspotential p zu bekommen. Um dies zu erreichen, müssen die Funktionen wie folgt umgeformt

² Der Break-Even-Punkt („Return of Investment“) oder die Gewinnschwelle ist in der Wirtschaftswissenschaft der Punkt, an dem Erlös und Kosten einer Produktion gleich sind und damit weder Verlust noch Gewinn erwirtschaftet werden.

werden:

$$p = K_f / x + k_v$$

Daraus kann die Annahme getroffen werden, dass die Anzahl an Interaktionen x das Einsparungspotential p indirekt proportional beeinflusst, da die anteiligen Fixkosten mit zunehmender Anzahl an Meldungen sinken.

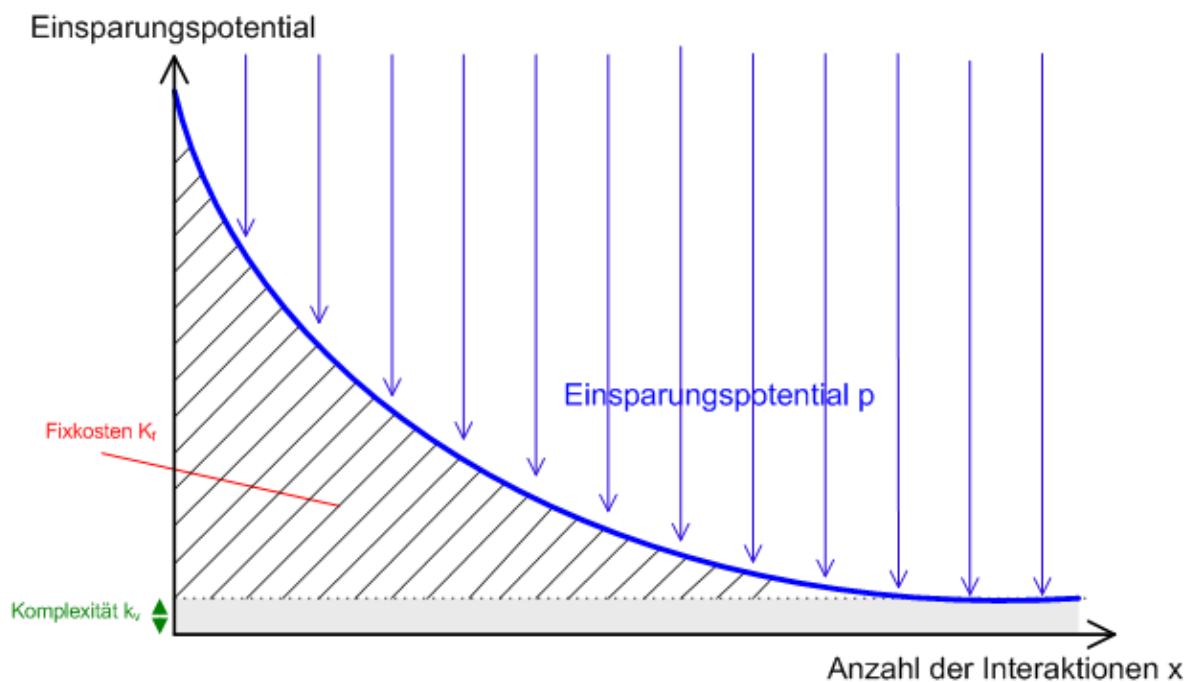


Abbildung 3: Einsparungspotential

Die Grafik aus Abbildung 3 zeigt, je mehr Anforderungen x bestehen, neue Informationen zu erfassen und zu bearbeiten, umso geringer ist der Kostenaufwand je Interaktion. Dies ist dargestellt durch den Bereich der Fixkosten K_f .

Der Parameter für die Komplexität der Informationsstruktur k_v in Abbildung 3 ist ausschlaggebend für die minimalen Kosten je Interaktion. Dieser stellt eine fixe Schranke dar, welche nicht unterschritten werden kann und unabhängig von der Qualitätsmeldungsanzahl mit der neuen Softwaretechnologie ist.

Ziel der Analyse war, durch den Einsatz der evaluierten Technologien die

Teilprozesse mit dem größten Einsparungspotential zu finden, denn je größer das Sparpotential, desto schneller wird der „Return of Investment“ erreicht.

Der Aufwand für Systemkonfiguration, Schulungs- und Lernprozesse wird in den jeweiligen Detailanalysen nicht berücksichtigt, jedoch sollten diese Mehrleistungen für eine weiterführende Analyse nicht komplett vernachlässigt werden. Es wird immer - auch bei den Qualitätsprozessen mit zusätzlich eingesetzter Software - von einem voll integrierten und funktionstüchtigen Umgebungsszenario ausgegangen.

In nachstehendem Kapitel wird eine detaillierte Ablaufbeschreibung des Qualitätsprozesses aus der Sicht des Lieferanten in den unterschiedlichen im Projekt umgesetzten Varianten dargestellt. Dabei werden folgende Varianten unterschieden:

- ***Qualitätsprozess ohne Adaptierung***
- ***Qualitätsprozess mit Lixto Software***
- ***Qualitätsprozess mit Lixto Software im SAP Enterprise Portal***

Jede dieser Varianten wird bei der Analyse in drei Teilprozesse unterteilt, um die jeweiligen Abläufe noch besser beschreiben und die zugehörigen Aufwände genauer abschätzen zu können. Die Unterteilung fällt folgendermaßen aus:

- ***Navigation und Benutzerparameter***
- ***Vergleich und Informationsübernahme***
- ***Informationsverteilung und Rückmeldungskontrolle***

Anschließend werden diese neuen Workflowunterteilungen (Varianten und Teilprozesse) im Kapitel 2.2.4 Ergebnis der Aufwandsanalyse miteinander verglichen und gegenübergestellt.

2.2.1 Qualitätsprozess ohne Adaptierung

In diesem Abschnitt wird der „Qualitätsprozess ohne Adaptierung“, das heißt ohne Veränderungen und ohne den Einsatz von weiterer Software, beschrieben.

Ausgegangen wird von einem Mitarbeiter, der die Aufgabe hat, neue Qualitätsinformationen von drei Automobilherstellern einzuholen und alle dazugehörigen Schritte zu bearbeiten. In der Realität werden Informationen von weit mehr Herstellern gesammelt, und auch die Aufteilung der Arbeit ist in den meisten Fällen auf mehrere Personen bzw. Abteilungen gestreut.

2.2.1.1 Navigation und Benutzerparameter

In fast allen Fällen müssen die Meldungen von unterschiedlichen Supplier Portalen erfasst werden. Als erstes muss der Mitarbeiter zur Informationsquelle des Herstellers navigieren, welche in den wenigsten Fällen Gemeinsamkeiten aufweisen. Aus Sicherheitsgründen muss jeder Meldungsbereich durch Benutzername und Passwort geschützt sein, wodurch auch ein direktes navigieren (wobei die Qualitätsmeldungsseite sofort angesprungen werden kann) nicht möglich ist. Der Benutzer hat auf herkömmliche Art und Weise keine Möglichkeit, sich einen der oben beschriebenen Arbeitsschritte zu erleichtern.

In unserem Analyseverfahren wird davon ausgegangen, dass ein Mitarbeiter für den oben beschriebenen Prozess 3 Minuten pro Herstellerseite braucht. Also ergibt sich für die Navigation zu allen Herstellerquellen ein kumulierter Aufwand von 0,15 Stunden.

2.2.1.2 Vergleich und Informationsübernahme

Ist der Mitarbeiter erfolgreich angemeldet und hat die gewünschten Informationen am Bildschirm, beginnt der Abschnitt des zweiten Teilprozesses. Die Meldungen müssen mit den bereits vorhandenen Daten verglichen werden, um anschließend nur die neu verfügbaren Einzelmeldungen zu übernehmen. Der Vorgang des Vergleichens kann in manchen Fällen relativ komplex sein, da nicht alle Herstellerseiten Gruppier- bzw. Sortierfunktionen unterstützen und damit ein „unscharfer Suchvorgang“³ der einzige Lösungsansatz ist.

³ „Unscharfe Suche“ oder „Fuzzy-Suche“ bezeichnet man den Vorgang, bei dem eine bestimmte Zeichenkette in einer längeren Zeichenkette bzw. einem Text gesucht wird. Typisch für die „unscharfe Suchmethode“ ist, dass nicht die exakte Zeichenfolge als Suchkriterium zugrunde gelegt werden muss, sondern auch ähnliche Zeichenketten gefunden werden sollen.

Wurden nun die Duplikateinträge vom Benutzer gefiltert, müssen die verbleibenden aktuellen Qualitätsmeldungen in das System des Lieferanten übernommen werden. So wie in den meisten Datenbank basierenden Systemen üblich, gibt es vorgegebene Felddefinitionen zur Neuerfassung von Meldungen (so wie z.B. im SAP R/3 Qualitätsmeldungs-Modul). Schwierigkeiten treten erst beim Zusammenführen der Informationen aus den unterschiedlichen Herstellerquellen mit unterschiedlich hinterlegten Werten und den meist statisch bereits vorgegeben Feldern der Lieferantensysteme auf.

Auch die Möglichkeit, dass normalerweise mehrere Personen Daten vergleichen und übernehmen, erleichtert in keinem Fall den Aufwand für diesen Teilprozess. Von anderen Mitarbeitern bereits übernommene Daten sind im Einzelfall wesentlich schwieriger zu überprüfen und falls ein individuelles Mapping⁴ der Qualitätskriterien erforderlich ist, birgt dies eine große Fehlerquelle für Dateninkonsistenzen.

Bei unserer Analyse wird davon ausgegangen, dass ein Mitarbeiter für oben beschriebenen Teilprozess „Vergleich und Informationsübernahme“ 10 Minuten pro Herstellerseite braucht. Also ergibt sich für oben beschriebene Arbeitsschritte bei allen Herstellern ein kumulierter Aufwand von 0,5 Stunden.

2.2.1.3 Informationsverteilung und Rückmeldungskontrolle

Nachdem alle relevanten Informationen der Hersteller korrekt übernommen werden konnten, müssen anschließend die neuen Daten intern, im besten Fall bereits automatisiert, über das Lieferantensystem weiterverteilt werden. Steht dem Mitarbeiter keine unterstützende Lieferantensystemlösung zur Verfügung, müssen die Informationen an individuell selektierte Benutzerlisten versandt werden.

Der Aufwand zur Kontrolle, ob alle relevanten Personen die neuen Qualitätsmeldungen auch wirklich erhalten haben bzw. ob während der

⁴ Unter dem Begriff Mapping versteht man eine Liste von Korrespondenzen, die äquivalente Bestandteile zweier heterogener Schemata miteinander in Beziehung setzt. Aus einem Mapping sollten sich Transformationsregeln ableiten lassen, mit denen sich die Daten aus dem einen Schema möglichst vollständig in das andere Schema überführen lassen.

Datenübernahme keine Fehler unterlaufen sind, wird in diesem Teilprozess der „Informationsverteilung und Rückmeldungskontrolle“ berücksichtigt.

Die Evaluierung dieses Teilprozesses hat ergeben, dass ein Mitarbeiter 5 Minuten pro Herstellerseite braucht, um alle Informationen zu verteilen und eventuelle Fehler auszubessern. Also ergibt sich bei allen Herstellern für oben beschriebene Arbeitsschritte ein kumulierter Aufwand von 0,25 Stunden.

Die nachfolgende Grafik (Abbildung 4) zeigt, dass der Gesamtaufwand für den „Qualitätsprozess ohne Adaptierung“ bei 54 Minuten liegt. Der Zeitbedarf des Mitarbeiters ist für die Selektion und die Übernahme der Qualitätsinformationen vom Hersteller- zum Lieferantensystem mit 30 Minuten am größten. Die sehr wenig individuelle Anpassung benötigenden Arbeitsprozesse „Navigation und Benutzerdateneingabe“ beziehungsweise „Informationsverteilung und Rückmeldungskontrolle“ verursachen einen Aufwand von 24 Minuten.

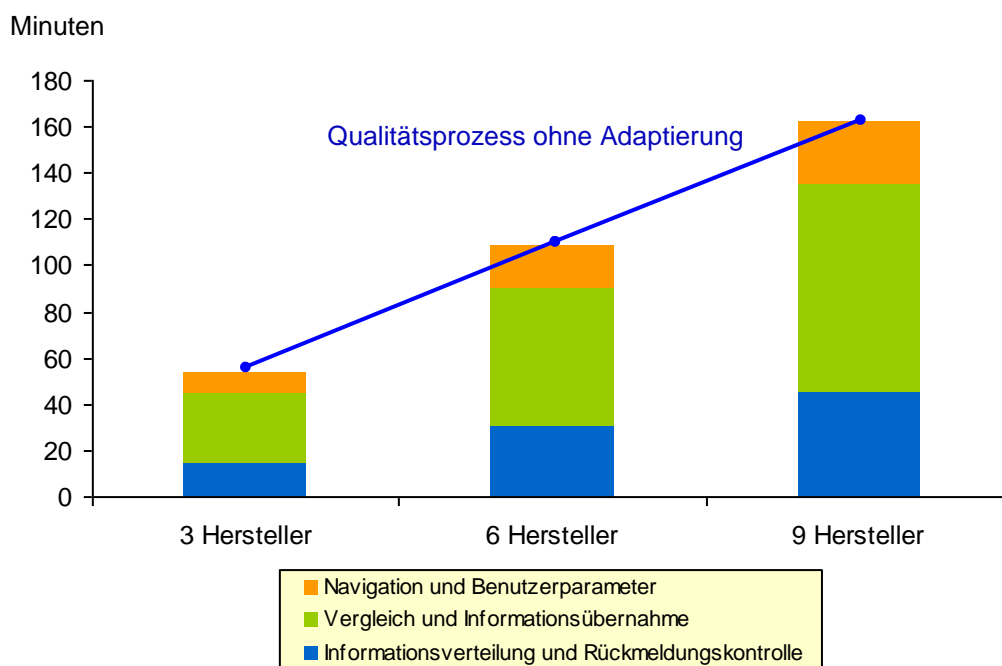


Abbildung 4: Qualitätsprozess ohne Adaptierung

Wird der gesamte „Qualitätsprozess ohne Adaptierung“ für mehr als drei Hersteller betrachtet, ergibt sich eine lineare Aufwandskurve (*blau*). Im gleichen Verhältnis wie sich die Anzahl der Hersteller verändert, so verändert sich auch der benötigte

Aufwand für den betrachteten Qualitätsmeldungsprozess. Als Beispiel werden in der Grafik zuerst drei, dann sechs und neun Hersteller evaluiert. Hier wächst der Zeitbedarf von 54 Minuten bei drei Herstellern, auf 108 bei sechs und 162 Minuten bei neun Automobilherstellern.

2.2.2 Qualitätsprozess mit Lixto Software

In diesem Abschnitt wird der „Qualitätsprozess mit Lixto Software“, das heißt unter der Verwendung von Lixto Software Komponenten und den daraus resultierenden Änderungen des Prozessverlaufs, beschrieben.

Voraussetzung für die Evaluierung dieser Variante waren die gleichen Anforderungen wie im „Qualitätsprozess ohne Adaptierung“. Ein Mitarbeiter hat die Aufgabe, neue Qualitätsinformationen von drei Automobilherstellern einzuholen und alle dazugehörigen Schritte zu bearbeiten. Dabei werden Teile der Einzelprozesse durch den Einsatz von Lixto Technologien adaptiert.

2.2.2.1 Navigation und Benutzerparameter

Der Teilprozess der Erfassung von neuen Informationsquellen wird durch die HTML-Seiten Navigation und Benutzerparametereingabe eingeleitet. Beide Arbeitsschritte können durch den Einsatz der Lixto Software Suite wesentlich erleichtert werden.

Die vordefinierte Navigationsfunktion des Transformation Servers kommuniziert in regelmäßigen Abständen vollkommen automatisiert mit den einzelnen Herstellerportalen, wobei die benutzerspezifischen Login-Parameter ebenfalls hinterlegt sein müssen.

Das Ergebnis unseres Analyseverfahrens war, dass ein Mitarbeiter nach der einmaligen Konfiguration der Benutzerparameter für oben beschriebenen Prozess im alltäglichen Routinebetrieb keinen zeitlichen Aufwand mehr hat. Alle Aufgaben werden vollkommen automatisiert vom Lixto Transformation Server übernommen und durchgeführt.

2.2.2.2 Vergleich und Informationsübernahme

Durch den Einsatz der Lixto Software können die neuen, noch nicht vom Transformation Server extrahierten Qualitätsmeldungen, dem Benutzer ganz selektiv dargestellt werden. Die Darstellung ist kundenspezifisch und kann E-Mail oder Webseiten basiert erfolgen. Damit kann der Arbeitsschritt des Vergleichens für die Mitarbeiter wesentlich vereinfacht werden, und die aktuellen Qualitätsmeldungen müssen lediglich in das System des Lieferanten übernommen werden.

Bei der Datenübernahme auftretende Probleme und Schwierigkeiten können durch den reinen Einsatz der Lixto Suite nicht vermieden werden. Individuelle Mapping-Konflikte bei mehreren Herstellerquellen und die Probleme bei einer Vielzahl an unterschiedlichen Benutzern bleiben weiter bestehen.

Ein Mitarbeiter braucht durchschnittlich für den Teilprozess „Vergleich und Informationsübernahme“ 5 Minuten pro Herstellerseite. Daraus ergibt sich für oben beschriebene Arbeitsschritte bei allen Herstellern ein kumulierter Aufwand von 0,25 Stunden.

2.2.2.3 Informationsverteilung und Rückmeldungskontrolle

Alle relevanten Herstellerinformationen konnten korrekt in das Lieferantensystem übernommen werden. Auch hier kann die Lixto Suite keinen sofort ersichtlichen Ansatz bieten, den Teilprozess für den Mitarbeiter zu vereinfachen. Die zeitlichen Aufwände sind komplett vom Funktionsumfang des Lieferantensystems abhängig.

Hat der Mitarbeiter keine unterstützende Lieferantensystemlösung zur Verfügung, müssen die Informationen an individuell selektierte Benutzerlisten versandt werden. Bei der manuellen Verteilung der Informationen können bereits vorhandene Funktionalitäten des Lixto Transformation Servers genutzt werden. Sobald relevante Daten extrahiert wurden, können diese nicht nur in komplexer Form an den betroffenen Hauptbenutzern übermittelt werden (siehe 2.2.2.1 Navigation und Benutzerparameter), sondern auch in vereinfachten Formaten an weitere Informationsempfänger. Eine Aktivierung einer Übermittlungs- und Kontrollfunktion ist E-Mail-, Datei- und Datenbank basiert jederzeit möglich.

Der Vergleich dieses Teilprozesses hat ergeben, dass ein Mitarbeiter 3 Minuten pro Herstellerseite braucht, um alle Informationen zu verteilen und eventuelle Fehler auszubessern. Daraus ergibt sich für die oben beschriebenen Arbeitsschritte ein gesamter Aufwand von 0,15 Stunden für alle Hersteller.

Die nachfolgende Grafik (Abbildung 5: Qualitätsprozess mit Lixto Software) zeigt, dass der Gesamtaufwand für den „Qualitätsprozess mit Lixto Software“ bei 24 Minuten liegt. Der Zeitbedarf des Mitarbeiters ist für die Selektion und die Übernahme der Qualitätsinformationen vom Hersteller- zum Lieferantensystem mit 15 Minuten weiterhin am größten. Der Aufwand für den Arbeitsprozess „Navigation und Benutzerdateneingabe“ fällt nach einer einmaligen Konfiguration, welche in der Detailanalyse nicht berücksichtigt wurde, komplett weg. Die „Informationsverteilung und Rückmeldungskontrolle“ wird mit einem geringeren Aufwand von 9 Minuten abgeschätzt.

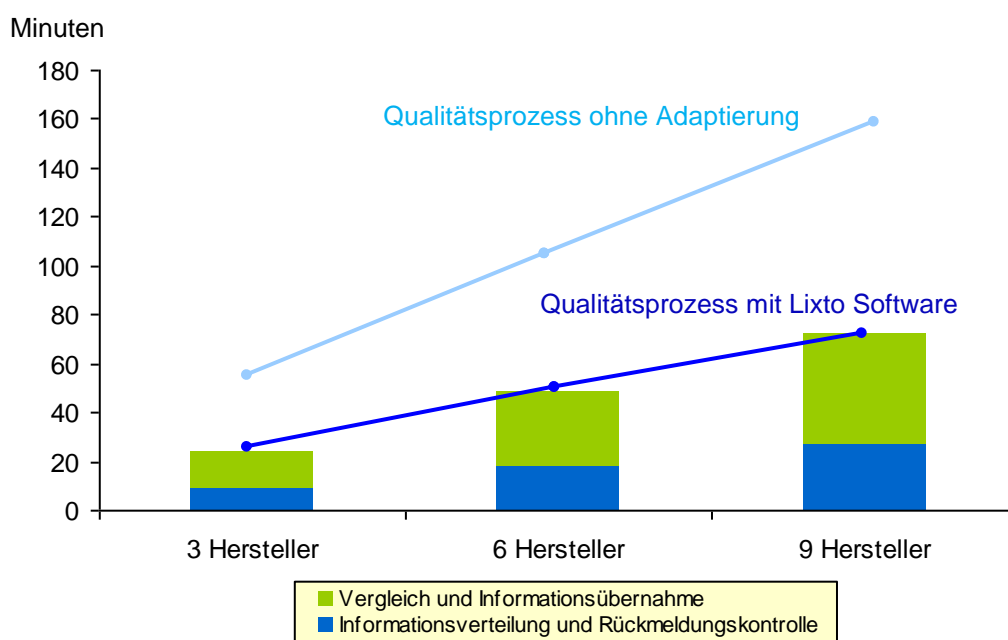


Abbildung 5: Qualitätsprozess mit Lixto Software

Wird der gesamte „Qualitätsprozess mit Lixto Software“ für mehr als drei Hersteller betrachtet, ergibt sich eine lineare Aufwandskurve (*blau*). Zu dieser ist anzumerken, dass die Aufwandskurve im beschriebenen Qualitätsprozess flacher ist, als im Prozess ohne Adaption (*hellblau*). Daraus lässt sich schließen, dass es durch den Einsatz der Lixto Software zu einem verringerten Gesamtaufwand

kommt, wobei sich dieser bei einer höheren Anzahl an abgefragten Herstellern stärker bemerkbar macht (siehe Abbildung 3: : Einsparungspotential). Als Beispiel werden in der Grafik zuerst drei, dann sechs und neun Hersteller evaluiert. Hier wächst der Zeitbedarf von 24 Minuten bei drei Herstellern, auf 48 bei sechs und 72 Minuten bei neun Automobilherstellern.

2.2.3 Qualitätsprozess mit Lixto Software im SAP Enterprise Portal

In diesem Abschnitt wird der Qualitätsprozess unter der Verwendung der Lixto Software Komponenten in Verbindung mit dem SAP Enterprise Portal beschrieben. Besonders zu beobachten waren dabei die Änderungen des Prozessverlaufs.

Auch hier wird von der gleichen Anforderung ausgegangen, wie Sie im „Qualitätsprozess ohne Adaptierung“ definiert wurde. Dabei werden fast alle Einzelprozesse durch den Einsatz von Lixto Technologien und die Einbindung in das SAP Enterprise Portal adaptiert.

2.2.3.1 Navigation und Benutzerparameter

Die beiden Arbeitsschritte der „Navigation und Benutzerparametereingabe“ können durch den Einsatz der Lixto Software Suite und des SAP Enterprise Portals extrem vereinfacht und erleichtert werden.

Die vordefinierte Navigationsfunktion des Transformation Servers kommuniziert in regelmäßigen Abständen vollkommen automatisiert mit den einzelnen Herstellerseiten. Die Eingabe der Login-Parameter erfolgt über die benutzerspezifische MyLixto Konfigurationsoberfläche, welche in das SAP Enterprise Portal integriert wurde. Damit kann auch für Verwaltungs- und Überwachungstätigkeiten der Vorteil des Single-Sign-On⁵ voll ausgenutzt werden.

Auch nach dem zusätzlichen Einsatz der SAP Software ist der Zeitaufwand des Mitarbeiters für den unter Punkt 2.2.2.1 beschriebenen Prozess unverändert

⁵ Unter dem Begriff Single-Sign-On versteht man die einmalige Anwender-Authentifizierung am System, wobei diese für alle nachfolgend benutzten Applikationen automatisch mit Hilfe von verschiedenen Technologien übernommen wird.

geblieben. Im alltäglichen Betrieb ist nach einer einmaligen Konfiguration kein zeitlicher Aufwand mehr vorhanden. Unsere Analysen konnten als Resultat des neuen Prozessablaufs zusätzlichen Komfort aufzeigen.

2.2.3.2 Vergleich und Informationsübernahme

Ergebnis dieses Teilprozesses sollte sein, dass alle relevanten Informationen in das Lieferantensystem übernommen wurden.

Durch den Einsatz der Lixto Software können die neuen Qualitätsmeldungen ganz selektiv extrahiert werden. Dargestellt werden die Daten in einer individuell konfigurierbaren Benutzeransicht des Enterprise Portals. So ist es für jeden Mitarbeiter leicht möglich, den geänderten Informationsbestand zu analysieren.

Zusätzliche Funktionalitäten des SAP Enterprise Portals ermöglichen es, die Meldungen auch weiter zu verarbeiten. Mapping-Vorgaben können dem Mitarbeiter automatisiert zur Verfügung gestellt werden, sodass eine Übernahme in das Lieferantensystem auf sehr einfachem Weg erfolgen kann. Im Idealfall können alle generierten Mappingvorschläge ohne Änderungsaufwand angenommen werden. Auch bei mehreren Mitarbeitern kann über diese Methode eine konsistente Datenübernahme gewährleistet werden.

Die Evaluierung hat ergeben, dass ein Mitarbeiter für den Teilprozess „Vergleich und Informationsübernahme“ 3 Minuten pro Herstellerseite braucht. Daraus würde sich für oben beschriebene Arbeitsschritte ein Aufwand von 0,15 Stunden auf Basis aller Hersteller ergeben.

2.2.3.3 Informationsverteilung und Rückmeldungskontrolle

Die Verteilung der Meldungen an betroffene Personen funktioniert ebenfalls vollkommen automatisiert über das SAP Enterprise Portal.

Mitarbeiter können die Qualitätsmeldungen über individuelle Benutzer- bzw. Benutzergruppenansichten im Portal betrachten. Ein eingebundenes Rechtekonzept kontrolliert die Datenübernahme bzw. die Anzeige von Detailinformationen.

Zusätzlich ist die Visualisierung und Einbindung des primären Lieferantensystems im Portal möglich. In diesem Anwendungsfall kann das SAP R/3 Qualitätsmeldungsmodul zur Weiterbearbeitung direkt integriert werden.

Die Analyse des Teilprozesses hat ergeben, dass pro Hersteller der Aufwand auf 2 Minuten reduziert werden kann. Der Aufwand zur Kontrolle, ob während der Datenübernahme Fehler unterlaufen sind, ist auch durch den Einsatz von Zusatzsoftware nur sehr schwer zu reduzieren. Es ergibt sich für die Arbeitsschritte des Qualitätsprozesses mit Lixto Software im SAP Enterprise Portal ein kumulierter Aufwand von 0,1 Stunden.

Der Gesamtaufwand für den Qualitätsprozess liegt bei ca. 15 Minuten. Wenn man die Einzelzeitaufwände betrachtet, könnte durch den Einsatz der SAP Software der Teilprozess „Vergleich und Informationsübernahme“ nochmals deutlich von 15 auf 9 Minuten reduziert werden. Der gesamte Zeitbedarf für die „Informationsverteilung und Rückmeldungskontrolle“ wurde ebenfalls auf 6 Minuten reduziert.

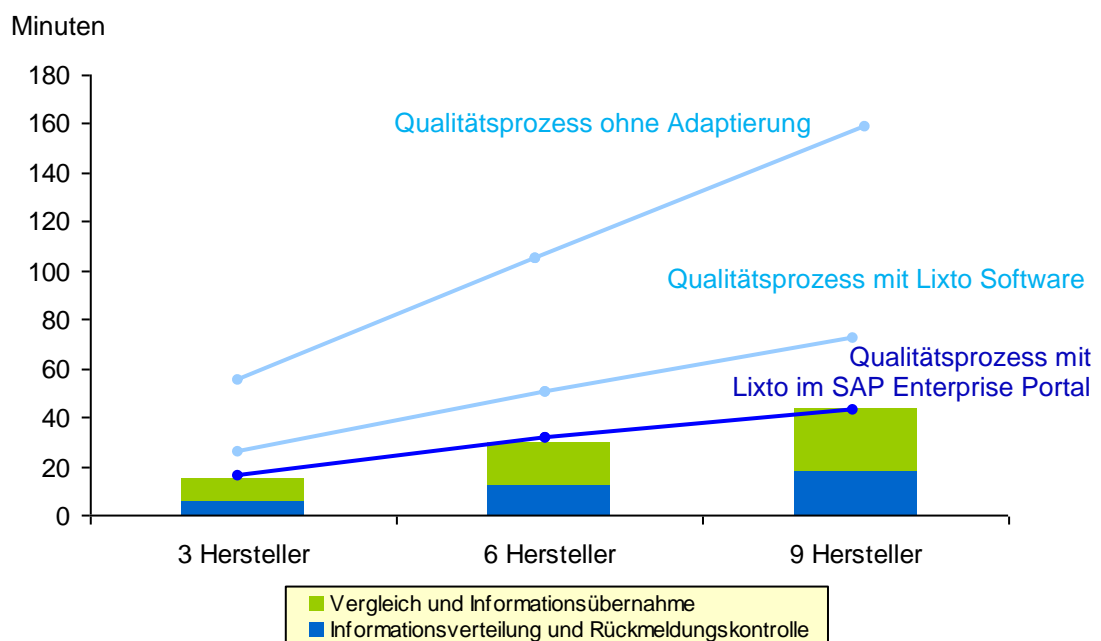


Abbildung 6: Qualitätsprozess mit Lixto Software im SAP Enterprise Portal

Wird der gesamte „Qualitätsprozess mit Lixto Software im SAP Enterprise Portal“

für mehr als drei Hersteller betrachtet, ergibt sich eine neue Aufwandskurve (*blau*). Diese ist im beschriebenen Qualitätsprozess flacher, als in den anderen Vergleichsprozessen (*hellblau*). Als Beispiel werden in der Grafik (Abbildung 6: Qualitätsprozess mit Lixto Software im SAP Enterprise Portal) zuerst drei, dann sechs und neun Hersteller evaluiert. Hier wächst der Zeitbedarf von 15 Minuten bei drei Herstellern, auf 30 bei sechs und 45 Minuten bei neun Automobilherstellern.

2.2.4 Ergebnis der Aufwandsanalyse

Die Analyse der Evaluierung bezieht sich auf die Ergebnisse bei der Verarbeitung von drei Herstellerseiten.

Der Vergleich des Qualitätsprozesses macht deutlich, dass allein durch den zusätzlichen Einsatz der Lixto Software der Aufwand auf 44,4 % des Gesamtaufwands reduziert werden kann. Durch die weitere Einbindung des SAP Enterprise Portals ist es möglich, den Gesamtaufwand auf 27,7 % zu senken. Davon kann man ableiten, dass rund 72,3 % des Zeitbedarfs für den betrachteten Arbeitsprozess im Umgang mit Qualitätsmeldungen eingespart werden kann.

Die nachfolgende Tabelle (Tabelle 1: Einsparungspotential der Qualitätsprozessvarianten) zeigt das prozentuelle Einsparungspotential der drei Qualitätsprozessvarianten im Bezug auf drei Herstellerquellen.

Zeitaufwand in Minuten	Qualitätsprozess		
	ohne Adaptierung	mit Lixto Software	mit Lixto Software im Enterprise Portal
Navigation und Benutzerparameter	9	0	0
Vergleich und Informationsübernahme	30	15	9
Informationsverteilung und Rückmeldungskontrolle	15	9	6
	54	24	15
Einsparungspotential	0 %	55,6 %	72,3 %

Tabelle 1: Einsparungspotential der Qualitätsprozessvarianten

Bei den Evaluierungsstudien sind folgende Vorteile ersichtlich geworden, welche Unternehmen durch den Einsatz der Lixto Suite und des SAP Enterprise Portals erzielen können. Diese können auch in anderen Bereichen Synergieeffekte hervorrufen:

- Zeitersparnisse bei alltäglichen anderen Tätigkeiten der Mitarbeiter
- Zentrale Verwaltung des Rechtekonzepts über verschiedene Arbeitsprozesse
- Single-Sign-On
- Soft- und Hardwareressourcenteilung für zusätzliche Anwendungen im Unternehmen

2.2.5 Ausblick

Der unter Punkt 2.2.1 „Qualitätsprozess ohne Adaptierung“ beschriebene IST-Prozess befindet sich momentan bei einigen Unternehmen der Automobilindustrie im Einsatz. Das gesamte Einsparungspotential wird erst dann ersichtlich, wenn man die evaluierten Aufwände der einzelnen Prozessschritte mit Zahlen reeller Anwenderdaten auswerten könnte.

In nachfolgendem Absatz soll das Einsparungspotential und die Vorteile im Großkundenbetrieb aufgezeigt werden. Dabei dürfen zeitliche und monetäre Mehraufwände nicht vernachlässigt werden.

Folgende Näherung soll zeigen, welche Ausmaße die Einführung weiterer Softwarelösungen für Zulieferbetriebe haben kann. Wir wollen ein Unternehmen betrachten, in dem 20.000 Mitarbeiter an das SAP Enterprise Portal angeschlossen werden. Damit sollen die unterschiedlichsten Einsatzgebiete abgedeckt werden. Geplant ist, dass der Bereich der Qualitätsmeldungskontrolle ebenfalls umgestellt und in das Portal integriert werden soll. Durch die weitere Einbindung der Lixto Suite wird die Interaktion mit allen Herstellerinformationssystemen ermöglicht. Die folgende Grafik (Abbildung 7: Erweiterte Qualitätsprozessbetrachtung) soll zeigen, wie die Aufwandskurven im Detail verlaufen würden und welches erweiterte Einsparungspotential sich daraus

ergeben würde.

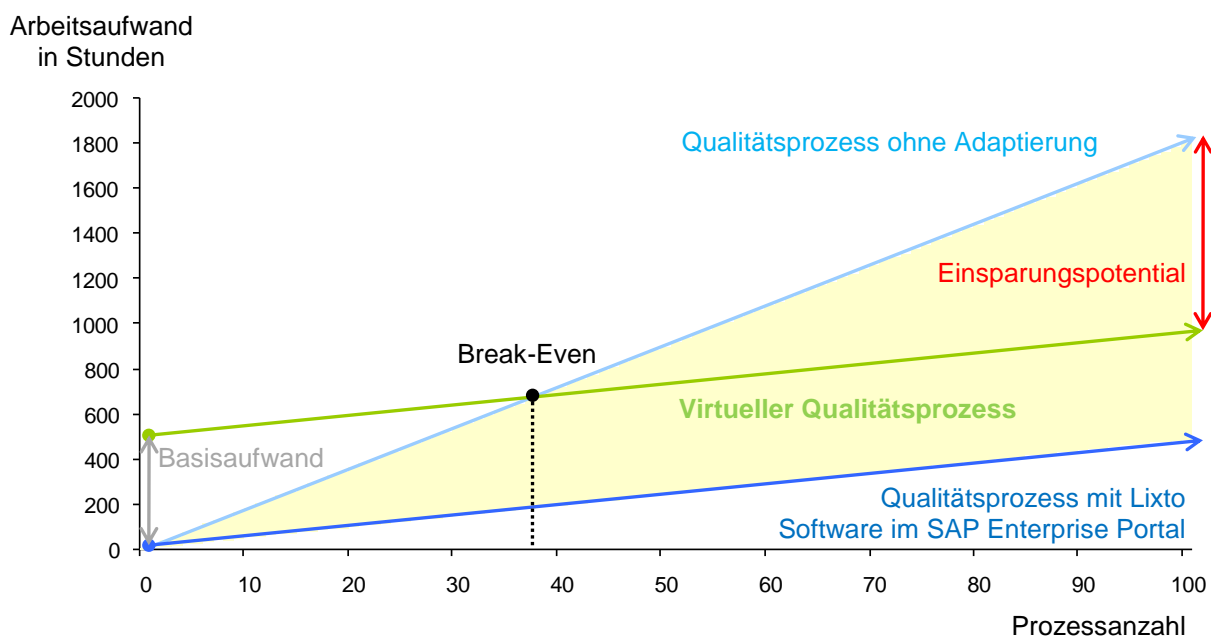


Abbildung 7: Erweiterte Qualitätsprozessbetrachtung

Der gelbe Kegel zeigt die Differenz der Aufwände zwischen dem „Qualitätsprozess ohne Adaptierung“ (*hellblau*) und dem „Qualitätsprozess mit Lixto Software im SAP Enterprise Portal“ (*dunkelblau*) unter der Annahme, 100 Einzelprozesse von 20 Quellen bearbeiten zu müssen. Die Anzahl der Arbeitsstunden, welche dabei eingespart werden könnten, beträgt 1315 (Einsparungspotential).

Um den Aufwand der Implementierung nicht ganz zu vernachlässigen, ist in dieser Grafik ein „virtueller Qualitätsprozess“ (*grün*) dargestellt. Dieser soll zeigen, dass sich bei einem Basisaufwand von 60 Personentagen (480 Stunden) der Einsatz der zusätzlichen Software ebenfalls bereits nach ca. 37 Prozessen rentiert hat. Bei dieser Anzahl ist der Break-Even Punkt erreicht. Ungewisse Faktoren wie Hardwareaufwände und Zusatzanforderungen sind weiterhin nicht berücksichtigt.

Das Evaluierungsergebnis zeigt, dass sich eine Adaptierung der Geschäftsprozesse im Bezug auf die Qualitätsmeldungsabwicklung für große Unternehmen die als Automobillieferanten tätig sind, auf jeden Fall rentieren würde.

2.3 Anwendungsarchitektur

Die detaillierte Beschreibung des Anwendungsfalls hilft, den Einsatz der neuen Systeme und deren Zusammenhang zu verstehen.

Die Grafik der Anwendungsarchitektur stellt den Informationsfluss der einzelnen Systeme für unser Szenario dar.

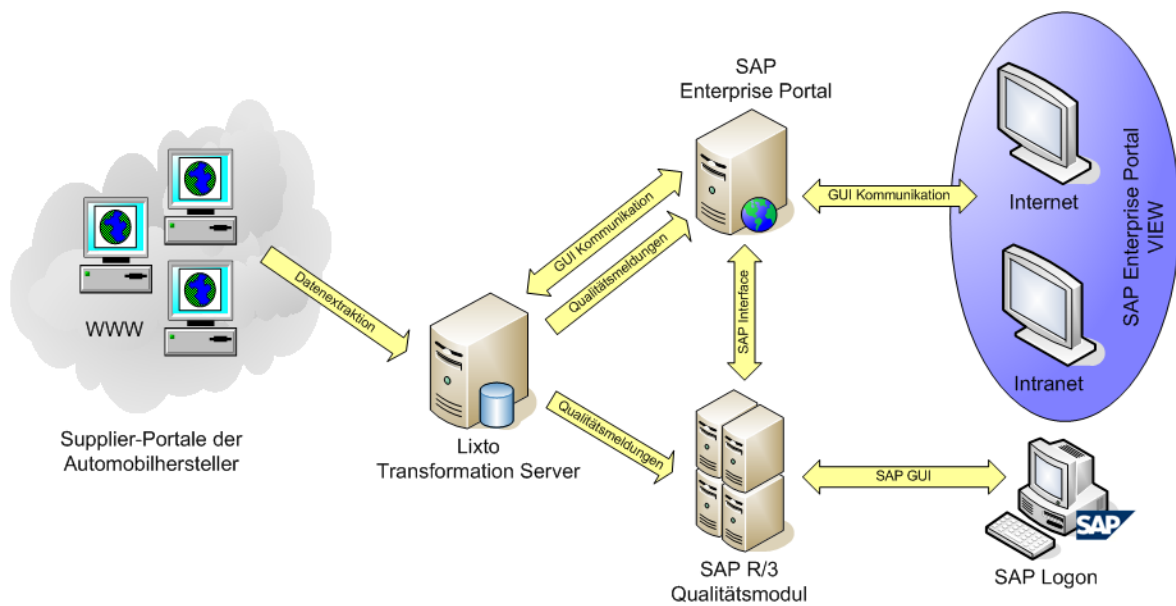


Abbildung 8: Anwendungsarchitektur

Die in dieser Abbildung (Abbildung 8: Anwendungsarchitektur) dargestellte Trennung der einzelnen Server zeigt, dass jedes dieser Systeme auch als Stand-alone Lösung bzw. Anbindungspunkt für andere Services voll einsatzfähig ist. Damit ist auch eine Skalierung von gleichen Aufgaben auf mehreren Systemen je nach Auslastung gewährleistet.

Auf dem Lixto Transformation Server werden die Informationen, hauptsächlich Qualitätsmeldungen der einzelnen Automobilhersteller, in periodischen Abständen gesammelt und anschließend für die Weiterverarbeitung im SAP Enterprise Portal bzw. SAP R/3 Qualitätsmodul aufbereitet.

Die Extraktion der gesicherten Meldungen von den verschiedensten Supplier-Portalen der einzelnen Automobilhersteller funktioniert durch den Einsatz der

speziellen Lixto Technologie vollkommen automatisiert. Dargestellt wird der Extraktionsprozess in der Grafik (Abbildung 9: Detailanwendungsfall Transformation Server) durch die Symbole eines Schlosses und einer Zeitstempeltabelle.

Das Schloss soll die individuelle Benutzerparameterverwaltung darstellen, wobei diese für jede Datenquelle eines Automobilherstellers unterschiedlich konfiguriert werden kann und diverse Authentifizierungsverfahren unterstützt. Die Tabelle soll die zeitgesteuerten Lixto Abläufe darstellen, welche ebenfalls für jeden Anwender bzw. Hersteller in Bezug auf die Frequenz und Startzeiten unterschiedlich zu konfigurieren sind.

In der Abbildung 9 sind auch zwei Firewall⁶-Symbole verwendet. Diese sollen einerseits zeigen, dass die örtliche Gebundenheit der Server vollkommen unabhängig voneinander ist und diese auch als Einzellösung eingesetzt werden können bzw. auch schon sind. Eine durchschnittlich dimensionierte Netzwerkanbindung für den gemeinsamen Servereinsatz ist vollkommen ausreichend. Andererseits sollen die Symbole aber auch auf integrierte Sicherheitskonzepte hinweisen, da die extrahierten und anschließenden weiterverarbeiteten Daten meist sehr sensibel sind und keinesfalls Unbefugten zur Verfügung stehen dürfen.

In einem ersten Schritt wird über komplexe Mechanismen und vom Benutzer individuell vordefinierte Authentifizierungsmerkmale zur Quelldatenseite (überwiegend HTML) navigiert (1). Durch das jeweilig vordefinierte Lixto Visual Wrapper Template (2) ist es möglich, die Inhalte der Quelldatenseite in strukturierte XML-Daten zu extrahieren (3). Bei der Aufbereitung müssen die Formate und die Reihenfolgen der Daten geändert (4), sowie programmspezifische Syntax hinzugefügt werden (5). Dies geschieht in den meisten Fällen über XSLT-Prozesse, welche über Scripts gesteuert werden und in speziellen Stylesheets hinterlegt sind. Die konvertierten Daten werden an vordefinierten Speicherorten abgelegt (6), um dann als Webservercontent

⁶ Eine Firewall ist in der Computertechnik eine Netzwerk-Sicherheitskomponente, die Netzwerkverkehr anhand eines definierten Regelwerks erlaubt oder verbietet.

dargestellt werden zu können (7).

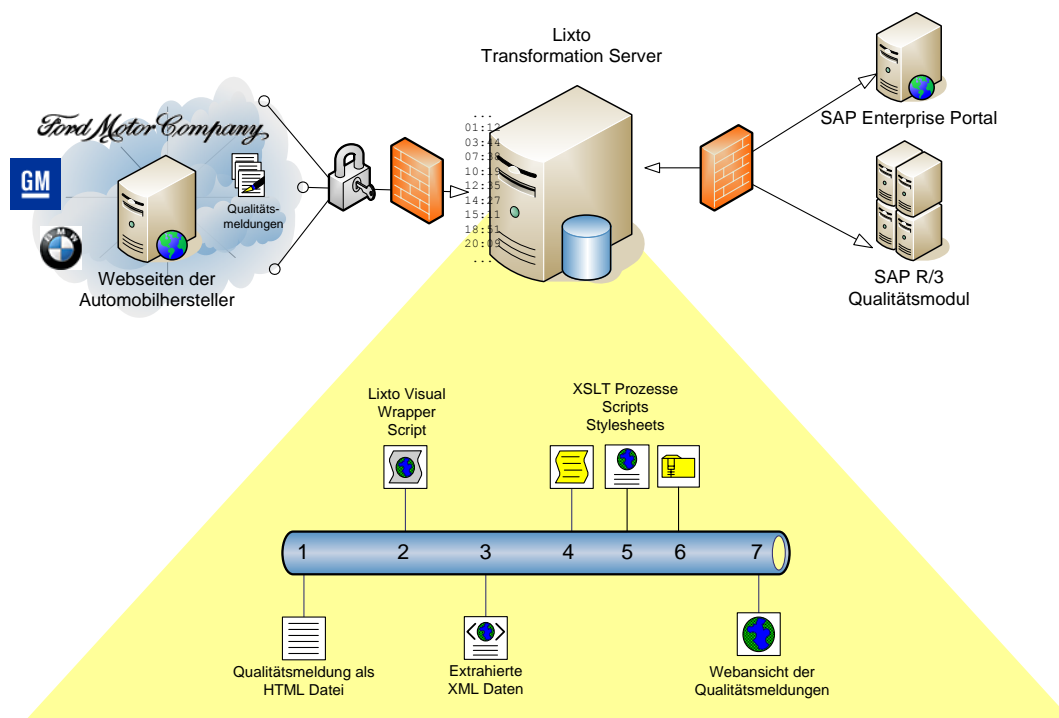


Abbildung 9: Detailanwendungsfall Transformation Server

Die Abbildung 10 beschreibt den Detailbereich des SAP Enterprise Portals. Dieses integriert die vom Lixto Transformation Server gewonnenen Daten in eine für den Benutzer gewohnte Arbeitsoberfläche und ermöglicht auch die Darstellung dieser. Die Berechtigungen für die Anzeige und Adaption von benutzerspezifischen Informationen werden auch vom zentralen Teil des Systems, dem Enterprise Portal, überprüft, wobei die Verwaltung dieser übergreifenden Daten meist in Basissystemen der Organisation stattfindet (LDAP).

Alle direkt als Output vom Transformation Server angezeigten Informationen können im Portal vom Benutzer selektiert und weiterbearbeitet werden, bevor der Speichervorgang im SAP R/3 gestartet wird. Auf diese gespeicherten Datensätze und jegliche anderen Module des eingebundenen SAP R/3 Systems kann natürlich auch über das Enterprise Portal zugegriffen werden.

Weiters ist es auch möglich, individuelle Einstellungen des Transformation Servers für jeden Benutzer über das SAP Enterprise Portal zu konfigurieren.

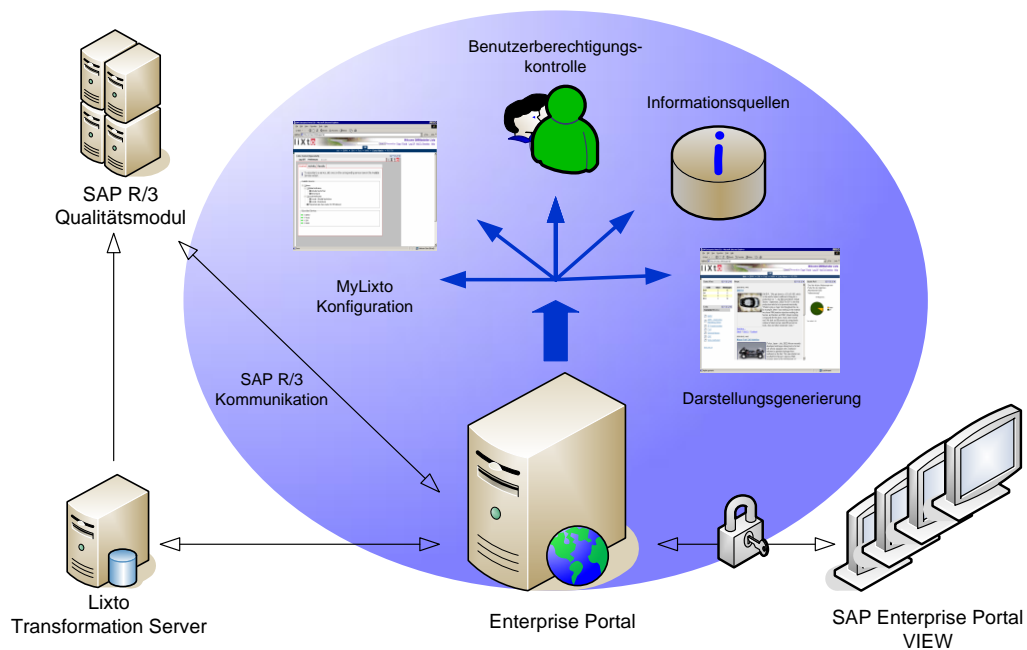


Abbildung 10: Detailanwendungsfall SAP Enterprise Portal

Der Zugriff für Benutzer auf das SAP Enterprise Portal funktioniert browserbasiert über das HTML-Format. Damit besteht auch die Möglichkeit, es in nahezu jede Systemlandschaft einzugliedern.

Die Hauptaufgabe des SAP R/3 Systems ist die Einbindung und Speicherung der extrahierten Daten in schon bestehende mit SAP modellierte Prozesse. Diese meist im SAP R/3 Qualitätsmodul abgelegten Informationen können entweder direkt im R/3 System bearbeitet werden oder es kann auf sie über die Funktion von iViews⁷ aus dem Enterprise Portal heraus zugegriffen werden.

⁷ iViews sind kleine SAP Anwendungen, die es ermöglichen interne und externe Informationsquellen in einem SAP Enterprise Portal dazustellen. Diese können auf Dokumentdateien, E-Mails, Webseiten und weitere Daten in Unternehmensanwendungen zugreifen.

3 Lixto Suite

Bei der Evaluierung der Systemsoftware dieses Projekts haben wir uns für die Lixto Suite entschieden. Die Lixto Suite ist ein Produkt der Lixto Software GmbH und mit ihr soll der automatisierte Extraktionsprozess abgebildet und realisiert werden.

Die Lixto Suite ermöglicht gerade Großunternehmen unterschiedlichste B2B Prozesse miteinander zu kombinieren. Ebenfalls können webbasierte E-Business Prozesse effektiv und automatisiert realisiert werden, ohne auf die Infrastruktur anderer Webservices zurückgreifen zu müssen und ohne die gewohnten IT-Umgebung anpassen zu müssen.

Die Lixto Suite besteht aus zwei unabhängigen Softwarelösungen, dem *Lixto Visual Wrapper* (VW) und dem *Lixto Transformation Server* (TS). Das Programm Visual Wrapper ermöglicht das Erstellen von individuellen Extraktionsformalissen. Der Transformation Server übernimmt die eigentliche Anwendung der Formalissen auf den unterschiedlichen Ressourcen und die Weiterverarbeitung der gewonnenen Daten. Im Kapitel 3.1 und 3.2 werden die zwei Komponenten der Lixto Suite im Detail beschrieben, wobei die im Projektzusammenhang stehenden Teilbereiche verstärkt erläutert wurden.

Erst durch den gemeinsamen Einsatz dieser Programme wird es möglich, Daten von Webressourcen einfach und automatisiert zu extrahieren, die Informationen zentralisiert zu sammeln und sie in strukturierten Formaten an beliebige Anwendungen weiterzuleiten.

3.1 Lixto Visual Wrapper

Der Visual Wrapper ist ein wesentlicher Bestandteil der Lixto Suite und ermöglicht die Erstellung von intelligenten Extraktionsvorlagen.

Beim Lixto Visual Wrapper handelt es sich um eine visuelle und interaktive Anwendung, die es Entwicklern ermöglicht, unstrukturierte Informationen von Webseiten in das strukturierte XML Format umzuwandeln. Dabei werden speziell für die optimale Extraktion entwickelte Techniken verwendet. Zum Beispiel können die Extraktionselemente in disjunktiver und konjunktiver Beziehung zueinander stehen. Während der Extraktion können automatisch unterschiedliche Seitenaufrufe veranlasst werden oder es können rekursive Wrapper zum Einsatz kommen. Bereits definierte und abgespeicherte Templates können auch Inhalte von sich ändernden Basisdokumenten extrahieren. Als Eingangsdaten können folgende Dokumentformate verwendet werden:

- **HTML Webseiten**
- **PDF Dokumente**
- **XLS**
- **CSV Textdateien**

Dokumente vom Typ PDF, XLS und CSV werden zur Definition des Wrappers in das HTML Format umgewandelt.

3.1.1 Visual Wrapper Systemarchitektur

Die Lixto Visual Wrapper Anwendung besteht aus den Komponenten des Wrapper Designers und des Wrapper Executors. Im Lixto Wrapper Designer werden die HTML Extraktionsvorgaben und XML Umwandlungsdefinitionen erstellt und können als Wrapper Projekt abgespeichert werden. Der Lixto Wrapper Executor wendet die vordefinierten Templates bei unterschiedlichen Inputobjekten an und erstellt den gewünschten XML Output. Dieser Vorgang entspricht auch der Grundfunktionalität des Transformation Servers ohne der Verwendung von Ausführungs- und Parametrisierungsautomatismen.

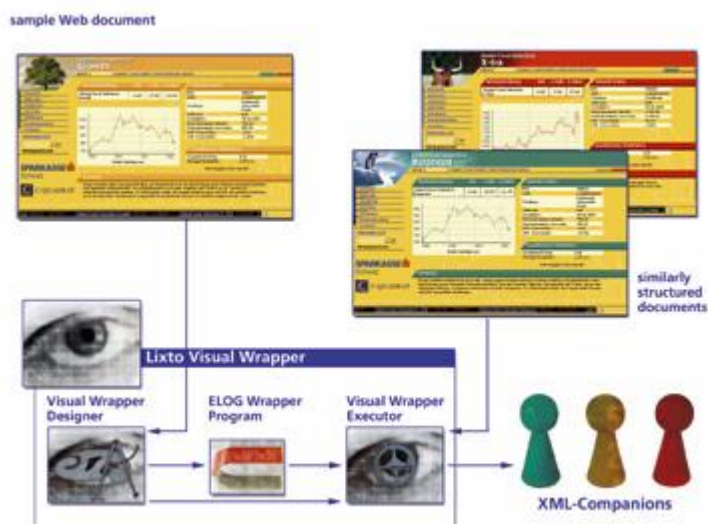


Abbildung 11: Visual Wrapper Systemarchitektur [lix08b]

Die interne Kommunikation zwischen den einzelnen Visual Wrapper Komponenten basiert auf der effizienten Programmiersprache ELOG. Diese wurde gemeinsam von der Lixto Software GmbH und der Technischen Universität Wien entwickelt. Dabei handelt es sich um eine „datalog“ Extraktionssprache mit strukturiert definierter Semantik, welche flexibel, sehr intuitiv und einfach zu erweitern ist. Alle Funktionsbausteine und Elemente von ELOG wurden anwendungstechnisch mit dem Lixto Visual Wrapper über eine visuelle Benutzeroberfläche umgesetzt.

3.1.2 Konzept des Visual Wrappers

Um heterogene Informationen möglichst flexibel aus Webseiten auslesen zu können, gibt es im Konzept des Visual Wrappers drei Funktionsbausteine, mit denen die unterschiedlichsten Arten von Extraktionsregeln realisierbar sind. Der Strukturaufbau ist in Abbildung 12 dargestellt. Jede Regel besteht aus den folgenden Elementen:

- **Patterns** (3.1.2.1)
- **Filters** (3.1.2.2)
- **Conditions** (3.1.2.3)

Diese sind hierarchisch strukturiert und die „Pattern“ Objekte stehen in disjunktiver Beziehung, dies entspricht einer ODER Verknüpfung (keine, eine oder mehrere

Bedingungen können zutreffen) mit den „Filter“ Objekten. Diese stehen mit „Condition“ Objekten in konjunktiver Beziehung, dies entspricht einer UND Verknüpfung (alle Bedingungen müssen zutreffen) zueinander.

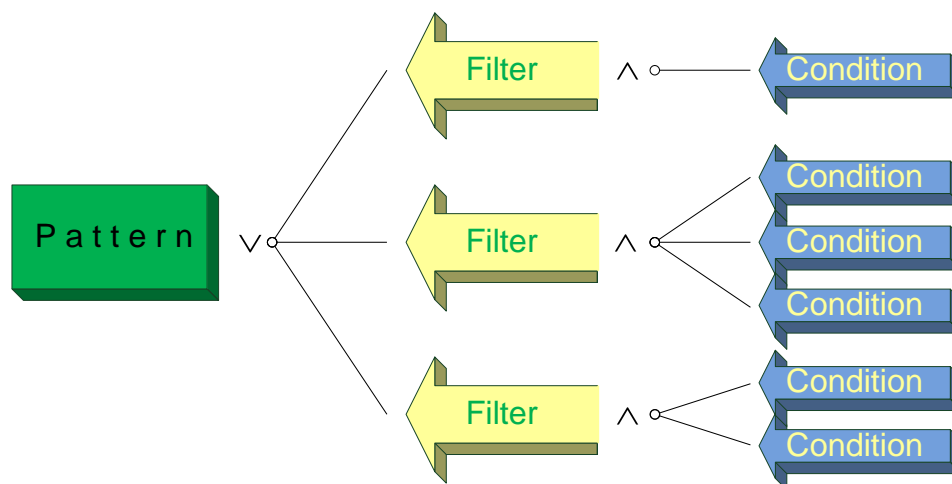


Abbildung 12: Funktionsbausteine Wrapper Regel

Eine logische Beschränkung der Anzahl je Element pro Ebene gibt es nicht, jedoch sollte man diese aufgrund der Übersichtlichkeit überschaubar halten.

3.1.2.1 Patterns

Als Pattern wird der eigentliche Regelsatz bezeichnet, welcher den Extraktionsinhalt einschränkt und die hierarchische Struktur definiert. Dieser kann aus mehreren Filtern bestehen, welche disjunktiv interpretiert werden.

Jeder Pattern wird beim Ergebnis in Form des XML Outputs als eigener Knoten dargestellt und benötigt ein übergeordnetes Element („parent“). Die oberste Ebene wird vom „root Pattern“ abgebildet, welcher automatisch mit einem neuen Wrapper Projekt angelegt wird. Dadurch ist es möglich, alle Elemente in Form einer Baumstruktur darzustellen, was der Basis von XML Dokumenten entspricht.

Beim Lixto Visual Wrapper können folgende drei Pattern Kategorien unterschieden werden:

- **Tree Pattern**
- **String Pattern**

- **Document Pattern**

Über Tree Patterns können strukturierte Elemente, wie Tabellen oder Listen und deren Inhalte extrahiert werden. Dies ist die am häufigsten eingesetzte Art von Patterns.

String Patterns werden verwendet, um auf HTML Seiten dargestellte (z.B. Emailadressen) und nicht dargestellte (z.B. der Name einer Bilddatei) Textbausteine zu extrahieren.

Mit Document Patterns können ganze Webseiten extrahiert werden, die in den meisten Fällen zur Navigation auf weiterverknüpfte Elemente verwendet werden. Ein Beispiel für einen Document Pattern wäre der immer verwendete „root Pattern“.

3.1.2.2 Filters

Ein oder mehrere Filter definieren einen Regelsatz, wobei jeder dieser Filter eine alternative zu extrahierende Datenmenge beschreibt. Filter sind abhängig von der übergeordneten Kategorie des Patterns und übernehmen diese automatisch. Filter können daher nach folgenden Arten unterschieden werden:

- **Tree Filter**
- **String Filter**
- **Document Filter**

Um einen Tree Filter zu definieren, ist die Angabe von einer Beispielinstantz, eines Pfads und weitere Attribute erforderlich. Bei der Spezifizierung des Pfads kann die Auswahl zwischen den drei Modellen „Default“, „Manual“ und „General“ getroffen werden.

Dargestellte Elemente können durch die Verwendung von regulären Ausdrücken, syntaktischen und semantischen Konzepten über String Filter eingeschränkt werden. Bei nicht sichtbarem Kontext wird die Selektion über Attribute vorgenommen.

Document Filter ermöglichen den Inhalt mehrerer Webseiten zu konkatenieren und in Form einer einzigen XML Struktur darzustellen. Diese Art von Filter wird benötigt, um Detailinformationen von verknüpften Seiten auszuwählen beziehungsweise Daten aus langen Listen, welche über mehr als eine Seite dargestellt werden, vollständig zu selektieren.

3.1.2.3 Conditions

Ein Filter besteht aus einer oder mehreren Conditions. Die unterschiedlichen Arten von Conditions schränken die Regelsatzergebnisse final ein. Dabei können folgende Modelle verschiedener Bedingungen unterschieden werden:

- ***Contextual Tree Condition***
- ***Contextual String Condition***
- ***Internal Tree Condition***
- ***Internal String Condition***
- ***Pattern Reference Condition***
- ***Nth Child Existence Pattern Condition***
- ***Range Conditions***
- ***Nth Child Existence Pattern Condition***

Zusätzlich können noch benutzerdefinierte und weitaus komplexere Conditions individuell hinzugefügt werden.

3.1.3 **VW Benutzeroberflächen (GUI)**

Beim Start des Lixto Visual Wrappers wird das Wrapper Fenster angezeigt. Dieses besteht aus den vier Bereichen der Statusleiste, der Wrapper (Baum-) Struktur, des Editierbereichs und dem bearbeiteten Dokument (siehe Abbildung 13).

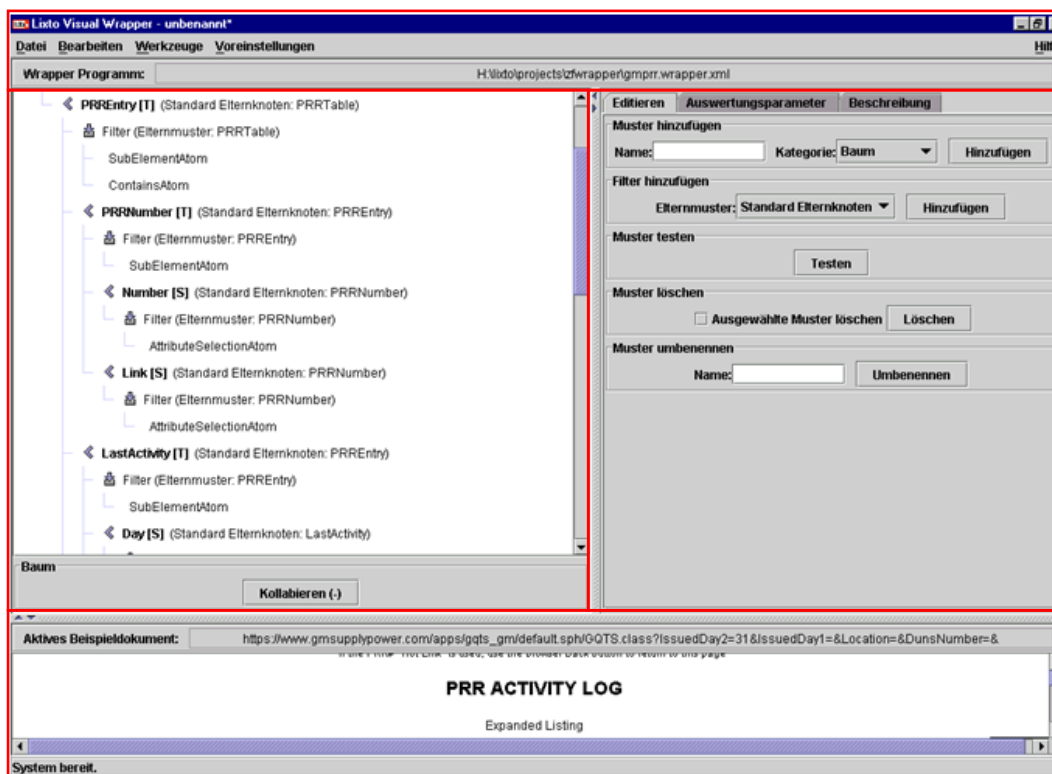


Abbildung 13: Visual Wrapper Benutzeroberfläche

Im Statusbereich, welcher sich ganz oben befindet, können allgemeine Aufgaben, wie Öffnen, Speichern und Schließen von Projekten erledigt werden. Zusatz- und Konfigurationsfenster können ebenfalls von hier aus gestartet werden. Das „XML Tool“ wäre eines dieser Zusatzanwendungen, welches eine Vorabdarstellung des XML Outputs ermöglicht.

Im mittleren linken Bereich wird die Struktur des Wrappers mit den einzelnen Pattern, Filter und Condition Bestandteilen dargestellt. Die Möglichkeit, diese zu modifizieren und bereits getroffene Definitionen auszulesen, bietet der rechte Bereich, in dem die angezeigten Informationen und Interaktionsmöglichkeiten von den selektierten Komponenten abhängig ist. Im Detail können Pattern Elemente hinzugefügt, umbenannt, getestet und gelöscht werden. Neue Filter können aus dieser Ansicht angelegt werden, wobei die erforderlichen Zusatzdefinitionen in Form eines Wizards eingegeben werden können. Bereits existierende Filter können zusätzlich editiert, umbenannt und gelöscht werden.

Das aktuell geöffnete Projektdokument wird in der Browseransicht ganz unten angezeigt und Ergebnisse von Extraktionstests werden ebenfalls in diesem

Bereich direkt auf der geladenen HTML Seite dargestellt. Mit Hilfe des „Document Managers“ können Online und Offline Dokumente eines Projektes hinzugefügt, aktualisiert und entfernt werden.

Eine weitere Anwendung der Lixto Suite, die gemeinsam mit dem Visual Wrapper zum Einsatz kommen kann, ist der Lixto Navigator. Mit ihm ist es möglich, Navigationssequenzen aufzuzeichnen und diese dann im Zuge der Wrapperausführung abspielen zu lassen. Die Aufzeichnung jeder Benutzerinteraktion ermöglicht es, den Inhalt von dynamisch strukturierten Webseiten automatisiert zu extrahieren.

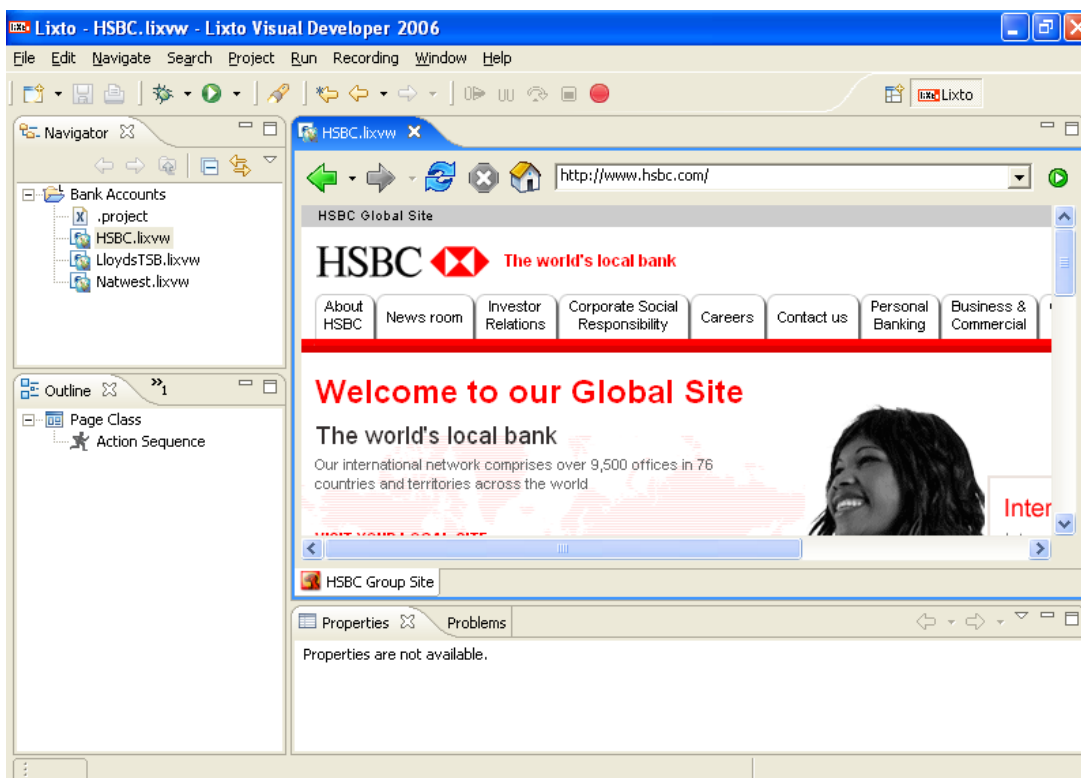


Abbildung 14: Visual Navigator Benutzeroberfläche [lix08h]

3.2 Lixto Transformation Server

Der Transformation Server ist das Kernstück der Lixto Suite und steuert zentral die einzelnen Aufgaben des Extraktionsprozesses.

Der Lixto Transformation Server nimmt die Eingangsdaten (Input) von verschiedenen Quellen, wie zum Beispiel Webseiten oder Datenbanken und wandelt sie in das standardisierte XML Format über einen XSLT - Prozess um. Anschließend können die zusammengestellten Daten transformiert (Transformation) und an ein beliebiges Ausgabeobjekt weitergeleitet werden (Output). Dieser Vorgang funktioniert vollautomatisiert ohne manuelle Adaption des XML Datenformats.

Die Teilprozesse Input, Transformation und Output werden etwas später in Kapitel 3.2.2 Technologie detaillierter beschrieben. Der Zusammenhang zwischen ihnen wird in der folgenden Abbildung dargestellt.

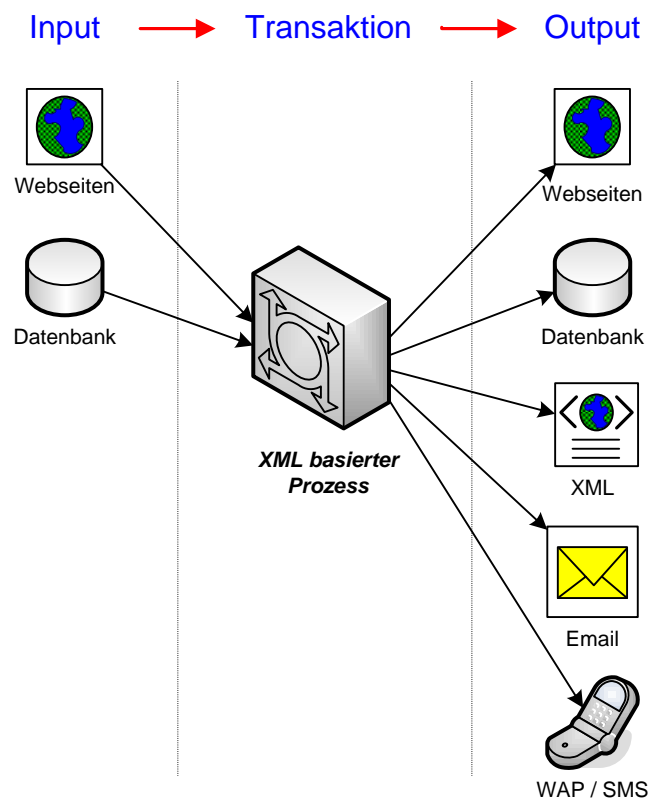


Abbildung 15: Transformation Server Input-Transaktion-Output Prozess

3.2.1 Konzepte des Transformation Servers

Um in allen Einsatzgebieten des Transformation Servers die jeweiligen Aufgabenstellungen möglichst optimal und benutzerfreundlich lösen zu können,

werden drei unterschiedliche Konzepte unterschieden:

- **Role Model** (3.2.1.1)
- **Pipes** (3.2.1.2)
- **Components** (3.2.1.3)

3.2.1.1 Role Model (Benutzerverwaltung)

Im Transformation Server sind drei standardisierte Rollen mit unterschiedlichen Rechten eingebaut, wobei jeder Benutzer einer dieser Rolle zugeordnet sein muss. Es kann zwischen den Rollen des System Administrators, des Service Designers und des Service Users unterschieden werden.

Als System Administrator des Transformation Servers hat man die Rechte und Aufgaben, die Systemeinstellungen zu konfigurieren, umgebungsabhängige Anwendungsparameter zu verwalten, weitere Benutzer anzulegen und Serverprozesse zu überwachen.

Service Designer können Workflows (Pipes) und die dazugehörigen Komponenten erstellen und verwalten. Als Besitzer einer Pipe haben Sie auch die Möglichkeit einzelne workflowspezifische Rechte, wie das Recht eine Pipe zu erweitern (Service Designer) oder zu personalisieren (Service User), an andere Benutzer weiterzugeben.

Service Users können an der Struktur der Pipes nichts ändern, jedoch haben Sie die Möglichkeit, Pipes über eine eigene Benutzeroberfläche (MyLixto) zu personalisieren und ihren gewünschten Anforderungen anzupassen.

Eine detaillierte Beschreibung der für das Rollenkonzept notwendigen Benutzeroberflächen und deren genaue Einsatzmöglichkeiten ist in Kapitel 3.2.3 TS Benutzeroberflächen (GUI) dokumentiert.

3.2.1.2 Pipes (Workflow)

Im Transformation Server versteht man unter einer "Pipe" eine grafische

Darstellung eines Informationsflusses. Dieser Workflow kann aus verschiedenen Komponenten (Source, Integrator, Transformer, Deliverer, ...) bestehen, welche in Kapitel 3.2.1.3 Components beschrieben werden. Erstellt, verwaltet und konfiguriert werden die einzelnen Komponenten von Benutzern, welche der Rolle Service Designers zugeordnet sind, wobei auch bestimmte Benutzer der Rolle Service Users ganze Pipes personalisiert nutzen können.

Die Aufgaben einer Pipe beinhalten die Informationsaggregation, -transformation und -verteilung basierend auf einem XML Datenstrom. Jede Aufgabe kann durch beliebig zeitgesteuerte und manuell gestartete Ereignisse ausgelöst werden.

Die Abbildung 16 zeigt die Gesamtstruktur eines Workflows und soll die Komplexität, mit der die einzelnen Komponenten zusammenspielen, deutlich machen.

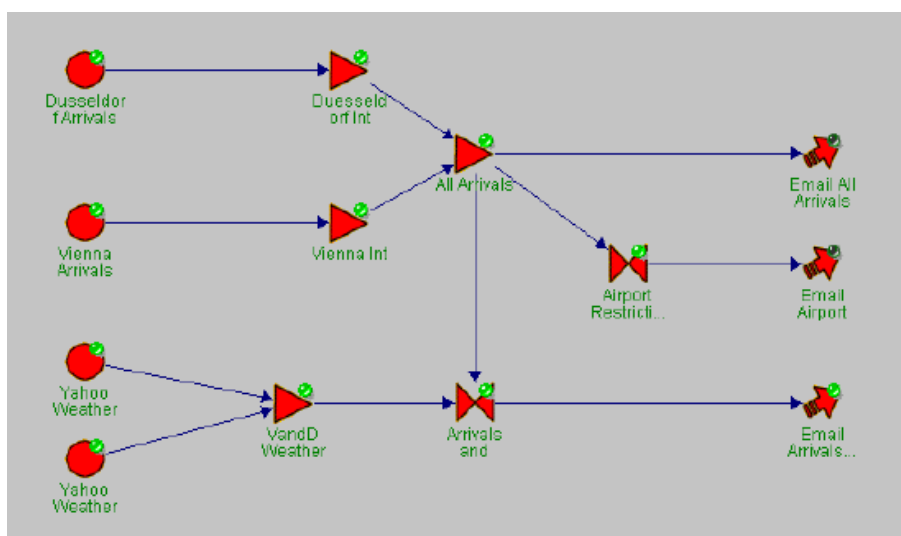


Abbildung 16: Beispiel einer Pipe

Eine detaillierte Beschreibung der Konfigurationsmöglichkeiten im Bezug auf das Forschungsprojekt wird in Kapitel 5.3 Lixto Transformation Server gegeben.

3.2.1.3 Components

Unter „Components“ bezeichnet man im Transformation Server die Bestandteile einer Pipe, wobei jeder der einzelnen Komponenten einen unterschiedlichen Prozess auf den XML basierenden Daten auslöst.

Nachfolgend werden die wichtigsten Komponenten, welche im Transformation Server genutzt werden können, beschrieben.



- Die Source Komponente

Über die Source Komponente werden die Informationsquellen einer Pipe definiert. Die Angaben verweisen in den meisten Fällen auf eine Internetressource (Web) und auf das dazu passende File mit Extraktionsparametern. Die spezifische Definition solcher Extraktionsdaten wird über den Lixto Visual Wrapper (siehe Kapitel 3.1 Lixto Visual Wrapper) erstellt. Die extrahierten Daten werden zur Weiterverarbeitung im Transformation Server in notwendige XML Struktur umgewandelt.

Weiters ist es über die Source Komponente möglich, Detailoptionen zur zeit- und ereignisgesteuerten Pipeaktivierung einzustellen und Informationen zur Fehleranalyse zu gewinnen.



- Die Integrator Komponente

Die Integrator Komponente ermöglicht es, eine Vielzahl an verschiedenen XML Source-Elementen zu kombinieren und die unterschiedlichen Inputdaten in ein strukturiertes Ausgabeformat umzuwandeln. Bei den ausgegebenen Daten handelt es sich ebenfalls um eine neu generierte XML Struktur.

Bei der Zusammenführung der einzelnen Elemente besteht die Möglichkeit, mit semantischen und synthetischen Prozessen zu arbeiten.



- Die Transformer Komponente

Die Transformer Komponente hat bei der Struktur einer Pipe die Aufgabe, Inputdaten über bestimmte Kriterien in Beziehung zu setzen (Join) und die generierten Ausgabewerte zu filtern.

Dabei ist es auch möglich, mehrere Kriterien gleichzeitig und unterschiedliche

Filtertypen auf den Eingangsdatenstrom anzuwenden.



- Die Iterator Komponente

Mit der Iterator Komponente können XML basierte Prozesse aus verschiedenen Pipes wiederholt ausgeführt werden und der Output anschließend über bestimmte Parameter bzw. Datenwerte mit anderen Prozessen in Beziehung gesetzt werden.



- Die XSLT Komponente

Über die XSLT Komponente können individuell gestaltete Stylesheets in den Workflow integriert werden. Im Gegensatz zu den sonstigen Komponenten kommen keine automatisch generierten Stylesheets zum Einsatz, sondern es werden extern definierte Formatierungen und Funktionen angewandt. Somit steht dem Anwender der komplette Funktionsumfang von XSLT Prozessen zur Verfügung.



- Die Differ Komponente

Mit der Differ Komponente ist es möglich, bereits extrahierte Informationen vom selben Datenstrom miteinander zu vergleichen und anschließend anhand bestimmter Kriterien einen neuen Output zu definieren.

In den meisten Fällen werden die Daten so gefiltert, dass bereits als Output gespeicherte Daten nicht noch einmal zur Verarbeitung weitergegeben werden und somit jeder Datensatz nur einmal abgelegt wird.



- Die Deliverer Komponente

Die Deliverer Komponente stellt die Verbindung zwischen den zuvor beschriebenen internen Lixto Bestandteilen (Source, Transformer, ...) und externen End-User Anwendungen oder Diensten dar. Dabei ist es möglich,

unterschiedlichste Technologien, wie Email, SMS, HTML, XML und WML zu nutzen.

Email und SMS Nachrichten werden direkt über vorkonfigurierte benutzerspezifische Dienste versendet und Daten in anderen unterstützten Formaten können an beliebigen Stellen im File System gespeichert werden. Im Detail können für die Deliverer Komponente noch bestimmte Ausgabeformate (Stylesheets) definiert, sowie Ausführungsoptionen (Datenausgabe zu einem bestimmten Zeitpunkt, Datenausgabe sobald sich die Inputdaten geändert haben, Datenausgabe nach einem bestimmten Zeitintervall, ...) gewählt werden.

Über die Source und Deliverer Komponente ist es auch möglich, auf ein externes Datenbankschema zuzugreifen. Eine zusätzlich konfigurierbare Middleware mit dem Namen XML-DBMS ermöglicht Lese und Schreib Prozesse zwischen XML Dokumenten und relationalen Tabellen der Datenbank. In den meisten Fällen handelt es sich bei der Datenbank um eine MySQL Datenbank, da diese bei Internetanwendungen sehr weit verbreitet ist.

Jede Komponente besitzt einen aktuellen Konfigurationsstatus. Dieser ist anhand des grünen bzw. roten Signals im Komponentensymbol, wie in Abbildung 17: Komponentenstatus ersichtlich. Das grüne Signal steht für eine korrekt konfigurierte und das rote für eine fehlerhafte Komponente. Bei der Erstellung einer Pipe durch mehrere Komponenten ist darauf zu achten, dass jeder einzelne Bestandteil des Workflows einen gültigen Status besitzt.

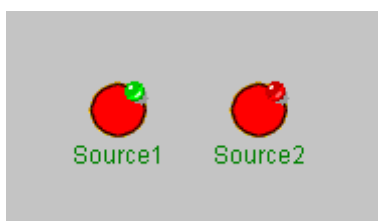


Abbildung 17: Komponentenstatus [lix08c]

Die einzelnen Komponenten können wie in Abbildung 18: Verbindungsmöglichkeiten der Pipe Komponenten definiert miteinander verbunden werden, wobei zwei verbundene Elemente einen funktionierenden XML

Datenstrom definieren.

		T O						
		Source	Integrator	Transformer	Iterator	XSLT	Differ	Deliverere
F R O M	Source	✓	✓	✓	✓	✓	✓	✓
	Integrator	✗	✓	✓	✓	✓	✓	✓
	Transformer	✗	✓	✓	✓	✓	✓	✓
	Iterator	✗	✓	✓	✓	✓	✓	✓
	XSLT	✗	✓	✓	✓	✓	✓	✓
	Differ	✗	✓	✓	✓	✓	✓	✓
	Deliverere	✗	✗	✗	✗	✗	✗	✗

Abbildung 18: Verbindungsmöglichkeiten der Pipe Komponenten

Die meisten im Transformation Server verwendeten Komponenten können nicht nur über statische Werte, sondern über benutzerspezifische Parameter konfiguriert werden. Diese Variante der Parametrisierung bietet sehr viele individuelle Möglichkeiten, eine Pipe zu nutzen und sie den speziellen Benutzerbedürfnissen anzupassen.

Benutzer, die nur als Service User mit dem System arbeiten, können ihre persönlichen Einstellungen über die MyLixto Oberfläche konfigurieren.

3.2.2 Technologie

Der Lixto Transformation Server ist eine Client-Server Applikation. Die Client Komponente kann vom Anwender bzw. Entwickler über eine webbasierte Oberfläche konfiguriert werden und stellt damit eine sehr einfache und praktikable Methode dar, komplexe Workflows zu definieren. Die Server Komponente ist ein unabhängiger Prozess der ad-hoc und zeitgesteuerte Anfragen bearbeitet, die Datenformatumwandlung durchführt und gegebenenfalls die unterschiedlichen

Zielsysteme über „push“ oder „pull“ Methoden anspricht.

Die Systemarchitektur des Transformation Servers ist in den meisten Fällen so konzipiert, dass dieser als Einzelserverlösung in einer Java Runtime Umgebung betrieben werden kann. Die benötigten System-, Benutzer- und Prozesskonfigurationen werden unabhängig von der Systemlandschaft abgespeichert. Die Konfigurationsoberfläche der Transformation Server Clients (TS Clients) ist über einen integrierten Web-Server anzusprechen. Bevorzugt wird der Apache Tomcat, wobei aber auch jeder andere J2EE kompatible Web-Server zum Einsatz kommen kann.

Der Transformation Server kommuniziert über verschiedene Technologien mit anderen Anwendungen bzw. Servern. Standardisiert sind zum Beispiel Komponenten, mit denen ein Mailserver (SMTP), eine Datenbank und Dateistrukturen angesprochen werden können. Zusätzlich kann der Transformation Server unter Verwendung von Extensions auch mit anderen Anwendungen bidirektional kommunizieren. Abbildung 19 stellt die internen Kommunikationswege des Transformation Servers dar.

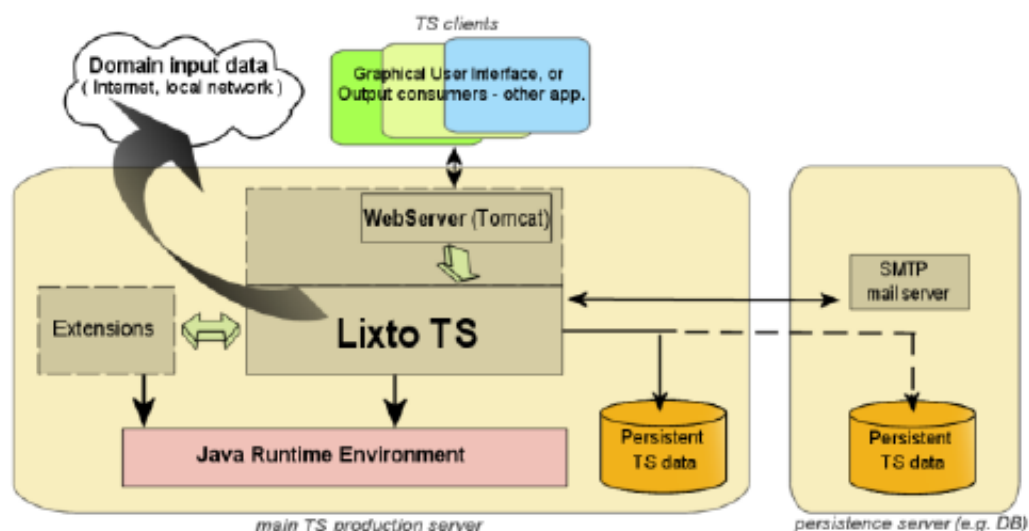


Abbildung 19: Transformation Server Architektur [lix08d]

3.2.3 TS Benutzeroberflächen (GUI)

Der Transformation Server gibt dem Benutzer die Möglichkeit, das System über drei verschiedene Oberflächen zu konfigurieren. Abhängig von den jeweiligen Benutzerrechten und dem zugeordnetem Rollenmodell, wird der Anwender automatisch mit der vordefinierten Oberflächenanwendung verbunden. Alle Anwendungen sind browserbasiert und ebenfalls vom Betriebssystem unabhängig.



Abbildung 20: Transformation Server Login

Es wird zwischen folgenden drei Rollen mit den dazugehörigen Benutzeroberflächen unterschieden:

- **System Administrator** → **TS Administration Console (3.2.3.1)**
- **Service Designer** → **TS Workbench (3.2.3.2)**
- **Service User** → **MyLixto (3.2.3.3)**

Die Trennung in unterschiedliche Rollen ermöglicht es auf Mitarbeiter- bzw. Benutzerebene Aufgaben und Qualifikation optimal zu nutzen und damit Ressourcen zu sparen.

Die Aufgabenbereiche der unterschiedlichen Rollen werden in Punkt 3.2.1.1 Role Model näher beschrieben.

3.2.3.1 Transformation Server Administration Console

Um den Anwendern des Transformation Servers eine benutzerfreundliche Serververwaltung und Wartung zu ermöglichen, ist im Lixto Softwarepaket eine Lösung integriert, die Transformation Server Administrations-Konsole (TS Administration Console).

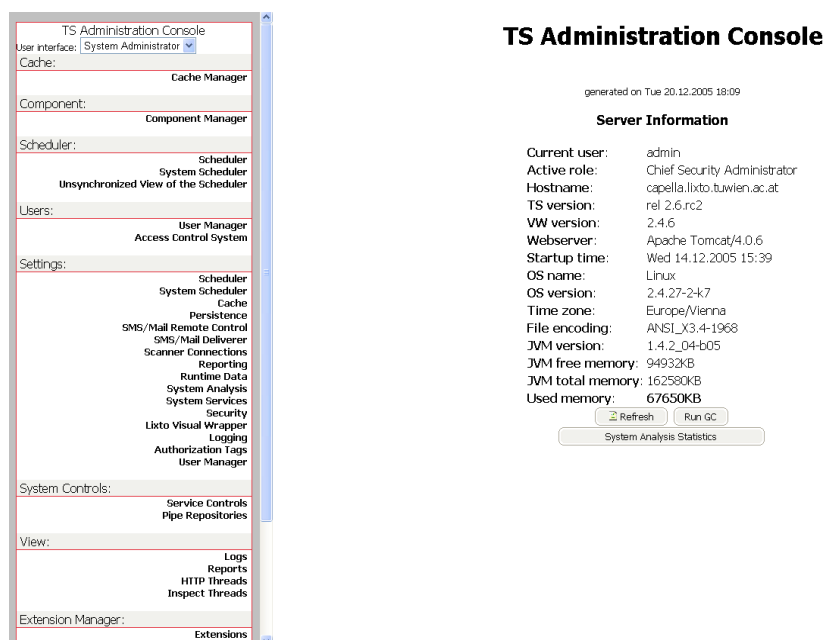


Abbildung 21: Transformation Server Console [lix08d]

Im Menü „Cache Manager“ können alle Elemente kontrolliert werden, die aktuell in den Speicher geladen wurden und über den Bereich „Component Manager“ können die bereits im Speicher befindlichen Pipe Komponenten überwacht werden. In beiden Menüpunkten lassen sich die Operationen „Speichern“ und „Entfernen“ auf die einzelnen Elemente anwenden. Alle im Hintergrund befindlichen Prozesse werden über geplante Jobs, bei denen die entsprechenden Ausführungsparameter hinterlegt sind, aktiviert. Der „Scheduler“ verwaltet diese zeitgesteuerten Jobs und stellt diese nach ihrem aktuellen Status dar. Die Sektion „Views“ bietet die Möglichkeit, Logdateien des Systems, Fortschrittsberichte und HTTP Anfragen zu überwachen. Im Bereich „Extensions“ lassen sich externe Module, wie ein MS Exchange Server oder ein Datenbank Connector einbinden, die über den Transformation Server angesteuert werden können. Unter „General“ lassen sich Systemparameter definieren und allgemeine Wartungsarbeiten durchführen.

3.2.3.2 Transformation Server Workbench

Meldet man sich am Transformation Server als Service Designer an, wird man zur persönlichen Workbench weitergeleitet.

Die Workbench Ansicht besteht aus drei Teilbereichen; dem Menü im oberen Bereich, dem Pipe Index links und dem Workspace rechts.

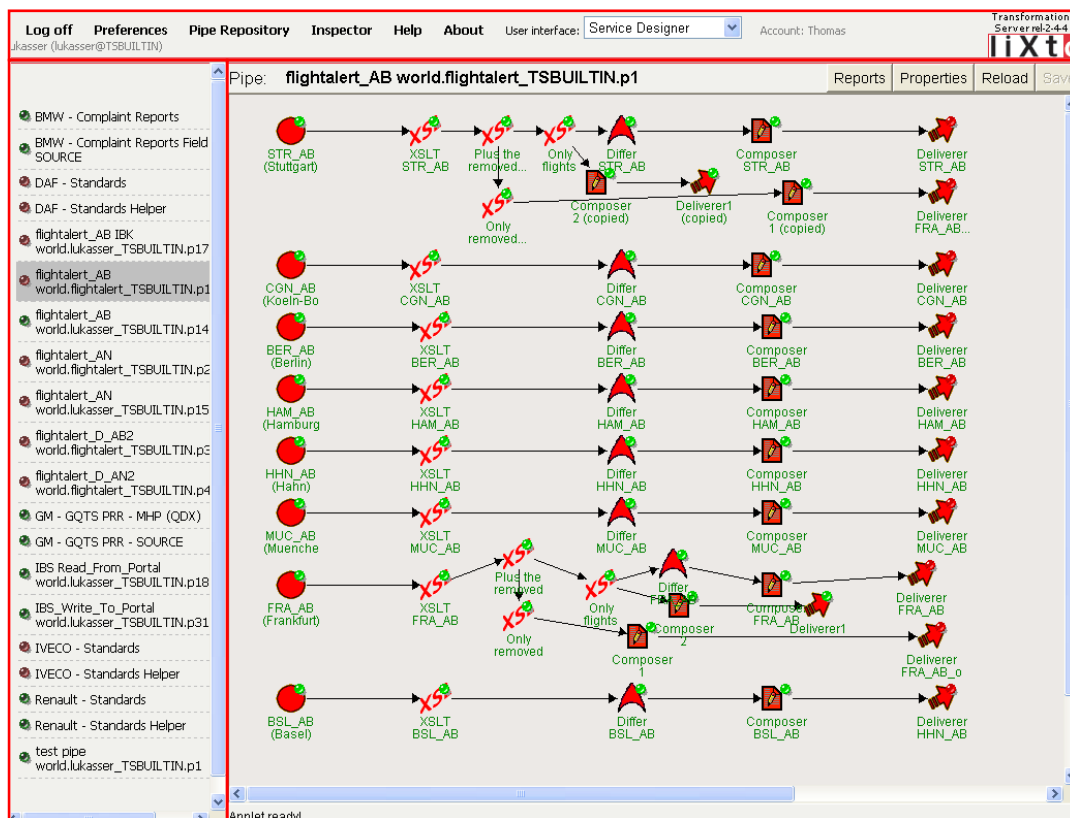


Abbildung 22: Transformation Server „Workbench“

Im Menü befinden sich die wichtigen Buttons „Log Off“, „Preferences“, „Pipe Repository“ und „Inspector“.

Im Bereich „Preferences“ kann der Anwender persönliche Einstellungen, wie Benutzername, Emailadresse, Formate und Layouteinstellungen definieren. Das „Pipe Repository“ beinhaltet alle verfügbaren Pipe Definitionen des Transformation Servers. Damit ist es möglich, Pipes zu erstellen, zu löschen, mit anderen Benutzern gemeinsam zu nutzen und zu kategorisieren. Auch eine Import und Export Funktionalität steht dem Anwender zur Verfügung. Unter „Inspector“ kann der aktuelle Status des Scheduler überprüft werden. Der „Log Off“ Button meldet den Benutzer vom Workbench ab.

Im Workbench Bereich Pipe Index werden alle dem Anwender zugeordneten Pipes angezeigt. Diese können aktiviert und deaktiviert bzw. ein- und ausgeblendet werden. Im Pipe Index markierte Pipes werden im Workspace angezeigt und stehen zur Bearbeitung bereit.

Der Workspace stellt die gerade aktive Pipe dar. Diese kann durch Hinzufügen, Modifizieren und Löschen von Komponenten adaptiert werden.

3.2.3.3 MyLixto

Die MyLixto Oberfläche ermöglicht einfachen Benutzern (Service-User) die Nutzung, Parametrisierung und Überwachung bereits vordefinierter Lixto Services.

Die in der Transformation Server Workbench von einem Service-Designer erstellten und einmalig konfigurierten Pipe-Komponenten können beliebig oft für unterschiedliche Quell- und Zielsysteme übernommen werden („Control“). Individuell benötigte Parameter müssen für jede neu eingebundene Umgebung adaptiert werden. Die Aktivität („Activity“) und Funktionalität („Results“) bereits im Einsatz befindlicher Dienste können möglichst zeitnahe in der MyLixto Oberfläche kontrolliert werden. Das MyLixto GUI kann in die drei bereits erwähnten Bereiche „Control“, „Activity“ und „Results“ unterteilt werden.

Auf der Registerkarte „Control“ sieht man alle Services, die einem Benutzer zur Verfügung stehen. Im unteren Bereich des Fensters sind die bereits abonnierten Services aufgelistet, wobei die jeweilige Farbe dem momentanen Aktivitätsstatus entspricht (Grün bedeutet ein Service ist aktiv und Rot bedeutet, dass ein Service deaktiviert wurde).

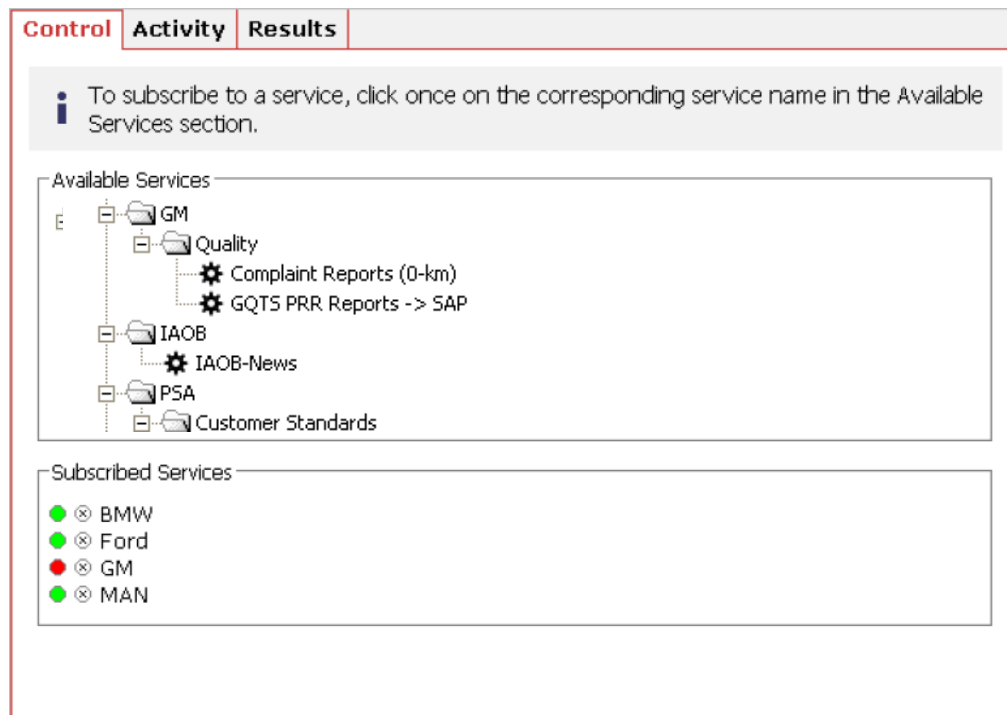


Abbildung 23: MyLixto „Control“

Auf der Registerkarte „Activity“ ist der aktuell Status, der nächste Ausführungszeitpunkt und der Status der bereits ausgeführten Aktivitäten jedes Services zu sehen. In diesem Beispiel ist das Service „Ford SIM“ ein Mal zu Beginn der Ausführungsperiode fehlgeschlagen.

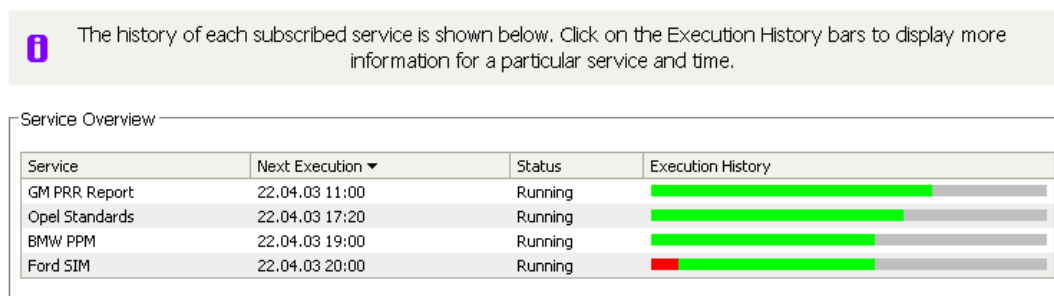


Abbildung 24: MyLixto „Activity“

Im Bereich „Result“ kann der Status der erzeugten Ergebnisse abgefragt werden. Jedes Ereignis wie eine Emailzustellung, eine HTML-Seitenerstellung oder das Verschicken einer SMS Nachricht ist mit einem Erfolgshinweis bzw. Datum und Uhrzeit protokolliert. Weiters ist es zusätzlich noch möglich, die detaillierten Inhalte der einzelnen Nachrichten (im XML-Format) abzufragen.

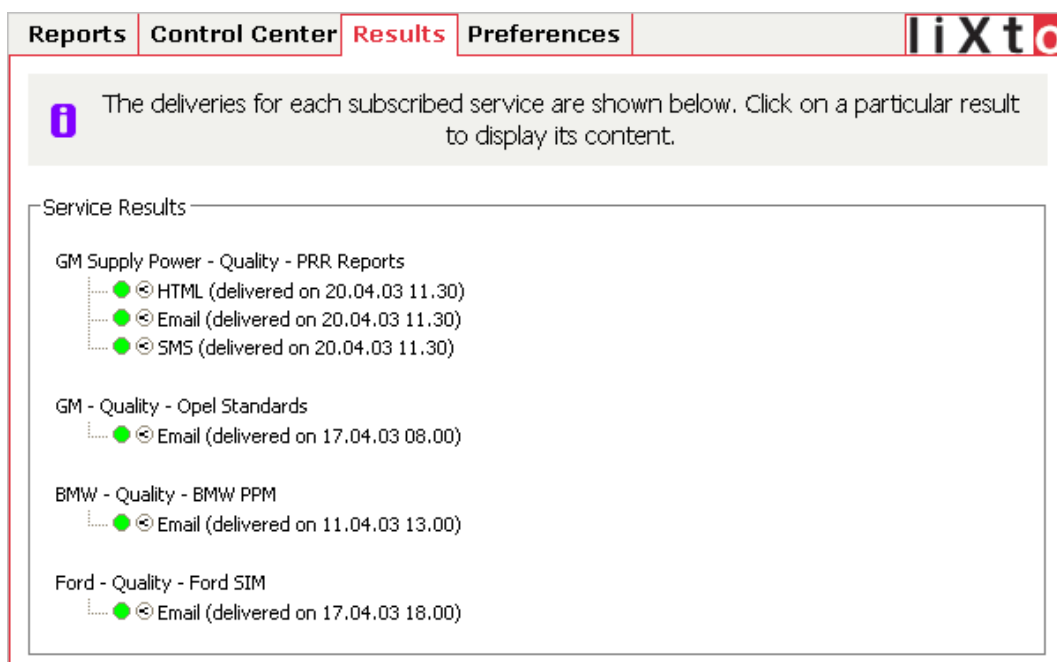


Abbildung 25: MyLixto „Results“

Im Projektumfeld wird von den Qualitätsmitarbeitern die Steuerung der Meldungsextraktion ausschließlich über die MyLixto Oberfläche durchgeführt.

4 SAP Enterprise Portal

Bei der Umsetzung dieses Softwareprojekts haben wir uns für das SAP Enterprise Portal entschieden. Das Enterprise Portal (EP) ist ein Produkt der SAP AG und ist Bestandteil der SAP NetWeaver Infrastruktur.

Das SAP Enterprise Portal ermöglicht, anwenderspezifische Informationen für eine große Anzahl an Personen aufzubereiten und eine interaktive Nutzung zu gewährleisten. Bei der Visualisierung und Weiterverarbeitung der Informationen, ist die ursprüngliche Herkunft für den Endbenutzer nicht relevant und ermöglicht somit einen vereinfachten Arbeitsprozess.

Das Enterprise Portal besteht aus mehreren Basiskomponenten, welche durch diverse Erweiterungen (z.B. Module für individuelle Branchenlösungen und unterschiedliche Konnektoren) ergänzt werden können. Im Zuge dieser Diplomarbeit werden nur eingesetzte Bestandteile der Basiskomponenten (Kapitel 4.3) und das Zusammenspiel mit der Softwarelösung Lixto Suite (Kapitel 5.4 und Kapitel 5.5) beschrieben.

4.1 Portaldefinition und deren Einsatzgebiete

Ursprünglich war der Funktionsumfang eines Internet-Browsers auf die Betrachtung reiner WEB Präsenzen fokussiert. Durch die Weiterentwicklung dieser betriebssystemunabhängigen Interaktionsmöglichkeit haben sich auch die Einsatzgebiete in und zwischen Unternehmen stark erweitert.

Bei Webportalen werden Technologien genutzt, die es ermöglichen, Applikationen in webbasierte Oberflächen zu integrieren. Somit können Portale als zentrale leistungsfähige Einstiegspunkte für die Abwicklung von Arbeitsprozessen und Verteilung von Informationen eingesetzt werden. Kriterien wie die Anwenderanzahl und örtliche Bindung stellen für den täglichen Betrieb nahezu keine Einschränkungen mehr dar.

Bis jetzt beinhaltete der tägliche Arbeitsprozess für den User erheblichen Interaktionsaufwand zwischen den einzelnen Anwendungen („Desktop“ Konzept).

In Abbildung 26 wird dargestellt, dass jede Anwendung durch ein eigenes Front-End angesteuert und bedient werden muss.

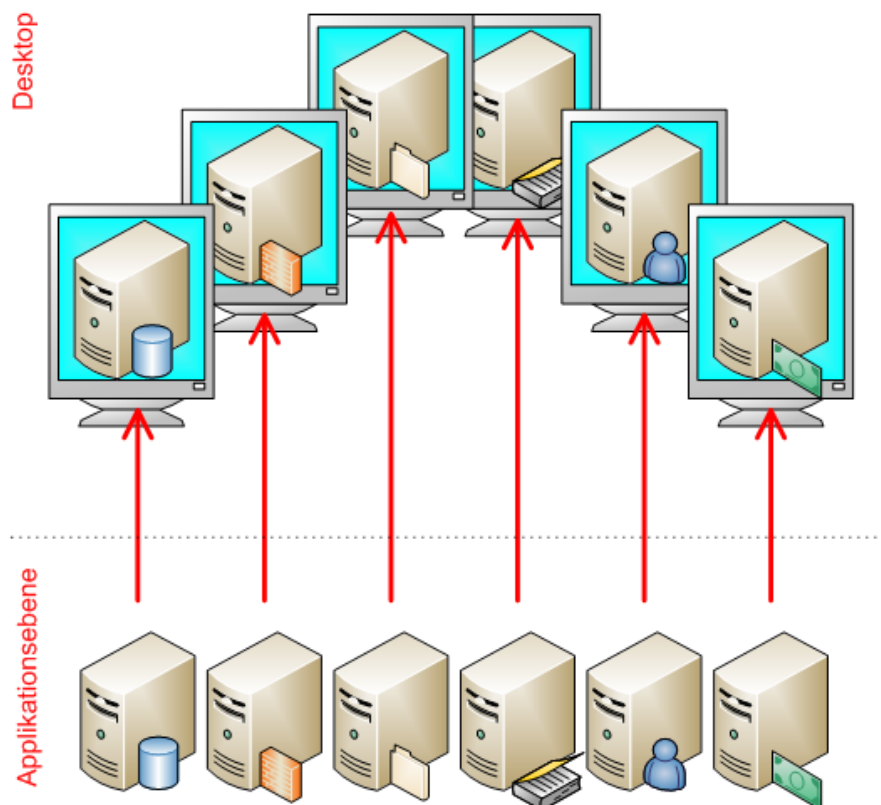


Abbildung 26: „Desktop“ Konzept

Bei Portalen wird der Interaktionsaufwand des Users auf eine zentralisierte Webanwendung („Webtop“ Konzept) [sap05] fokussiert, welche auch als „Web Desktop“ bezeichnet wird. Damit werden komplexe Prozesse, wie die Navigation und Suchvorgänge, aus dem Aufgabenbereich des Endbenutzers genommen und automatisiert durch neue Technologien von Portalen übernommen.

Die Abbildung 27: „Webtop“ Konzept zeigt die für den Anwender zentralisierten Interaktionswege einer „Webtop“ Lösung.

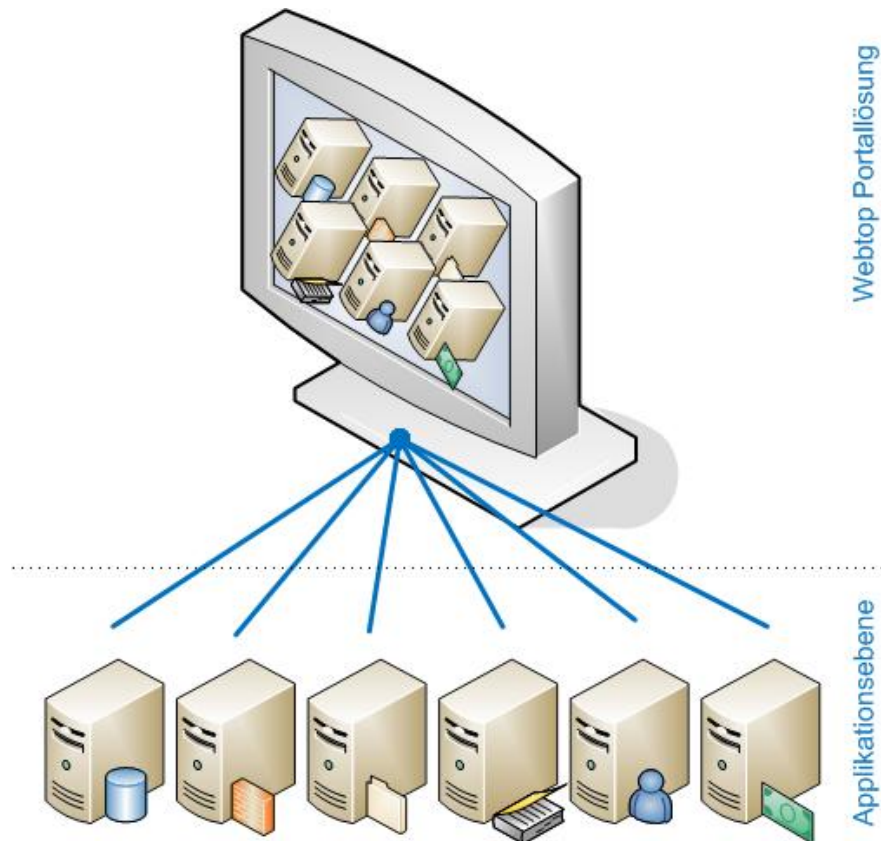


Abbildung 27: „Webtop“ Konzept

Mit Hilfe von Portalen kann für den Endbenutzer eine direkte Sicht auf den globalen Arbeitsbereich und die damit verbundenen Einzelprozesse geschaffen werden.

4.2 Vorteile beim Einsatz eines Portals

Der Einsatz eines Webportals [sap06] in einem großen Unternehmen, welches historisch bedingt auf sehr heterogenen Strukturen aufbaut, bringt organisatorisch und administrativ einige Vorteile. Die wichtigsten dieser Argumente können in folgenden Punkten zusammengefasst werden:

- **Einstiegspunkt über Single-Sign-On Konzepte (Kapitel 4.2.1)**
- **Individuelle rollenbasierte Benutzeroberfläche (Kapitel 4.2.2)**
- **Einfacher und geringer Wartungsaufwand (Kapitel 4.2.3)**
- **Sicherer und standardisierter Portalzugriff (Kapitel 4.2.4)**

4.2.1 Einstiegspunkt über Single-Sign-On Konzepte

Dem Anwender wird ein zentraler Zugriffspunkt über einen Webbrowser zur Verfügung gestellt. Die Portalauthentifizierung sollte nur ein Mal notwendig sein und kann mit einer beliebigen Technologie (User-Credentials, Fingerprint, Token, Security Cards,...) erfolgen.

Bei allen im Portal verwendeten Programmen können die Anmeldeinformationen direkt weitergereicht werden bzw. wird ein digital signiertes Anmeldeticket stattdessen weitergereicht, und es ist dadurch keine erneute Benutzeridentifikation notwendig.

4.2.2 Individuelle rollenbasierte Benutzeroberfläche

Jeder Benutzer arbeitet im Portal unter Verwendung eines bestimmten Profils, welchem die einzelnen Portalberechtigungen zugeordnet sind. Grundlage dafür ist ein spezielles Konzept auf der Basis von Rollen und Worksets. Im Rahmen dieses Rechtekonzepts, kann sich jeder Anwender seinen individuellen Webtop gestalten und diesen über sein Profil ablegen.

Im SAP Enterprise Portal ist damit eine Vielzahl an alternativen Konfigurationsmöglichkeiten nutzbar.

4.2.3 Einfacher und geringer Wartungsaufwand

Grundsätzlich muss aus Sicht des Endbenutzers nur eine Anwendung gepflegt werden, welche keine zusätzliche Clientsoftware benötigt. Das SAP Enterprise Portal ist ein offenes und flexibles System, mit dem jederzeit die Integration weiterer Komponenten möglich ist. Im Vergleich zu herkömmlichen Applikationen kann damit in großen Unternehmen der Wartungs-, Integrations- und Schulungsaufwand enorm gesenkt werden.

Die Systemparametrisierung erfolgt über spezielle Administrator-Rollen, welche ebenfalls über das Portal bedient werden. Damit sind rudimentäre Kontrollen und Überprüfungen der Funktionstüchtigkeit sehr einfach möglich.

4.2.4 Sicherer und standardisierter Portalzugriff

Das SAP Enterprise Portal verfügt über sehr komplexe Authentifizierungs- und Sicherheitsmechanismen. Diese sind für alle Benutzer bei jedem Zugriff auf das Portal gleich konfiguriert und erlauben keine Ausnahmen. Unabhängig von der Lokalität des Anwenders, können damit die hohen Sicherheitsanforderungen immer gewährleistet werden.

4.3 Integrationsmöglichkeiten

Relevant für den Erfolg eines Portals sind die mögliche Leistungsfähigkeit und der Individualisierungsgrad der Hintergrundprozesse. Je besser ein Portal eine Vielzahl an Applikationen miteinander verbinden kann, desto größer ist auch das entsprechende Einsatzgebiet. Für manche Szenarien ist es zusätzlich notwendig, weitere Integrationskomponenten, wie die Lixto Suite, einzubinden.

Beim SAP Enterprise Portal gibt es eine Vielzahl an Integrationsmöglichkeiten. Einige davon werden in den nachfolgenden Punkten aufgezählt und näher erläutert. Im Zuge dieses Diplomarbeitsprojekts kamen die Integrationskomponenten aus dem Kapitel 4.3.1, Kapitel 4.3.2 und Kapitel 4.3.3 zum Einsatz.

4.3.1 Enterprise Portal Standard Komponenten

Die einfachste Möglichkeit Informationen im Enterprise Portal darzustellen ist die Einbindung von iViews (siehe Kapitel 4.4 Enterprise Portal Content-Konzept). Über dieses können verschiedenste Informationsquellen angesteuert und integriert werden.

In einfachen Szenarien können iViews statische bzw. reine Webinhalte anzeigen, ohne zusätzliche Funktionalitäten zu nutzen. In komplexeren Umgebungen werden über iViews ganze Anwendungsinteraktionen gesteuert. Dann übernehmen iViews nicht nur die Darstellung in Form der Benutzeroberfläche, sondern sind die Schaltzentrale des Integrations- und Kommunikationsprozesses.

In unserem Projekt wurde die komplette Interaktion mit der Lixto Suite im Bezug auf die Datenextraktion und die Datenkonvertierung für das angebundene SAP R/3 System über iViews gelöst.

4.3.2 SAP Branchenlösungen und Business Packages

SAP bietet spezielle vorkonfigurierte Lösungen für einzelne Branchen an. Diese bieten betriebswirtschaftliche Funktionen und sind zu SAP Basissystemen, wie SAP R/3, vollständig kompatibel. Business Packages basieren auf den Standard Portalkomponenten, wie dem Rollenkonzept und einer Vielzahl von iViews, und bieten je nach Anwenderqualifikation unterschiedliche Sichten und Funktionsumfänge.

Im Enterprise Portal stehen diese als optionales Add-On zur Integration jederzeit zur Verfügung. Die folgende Liste ist ein kleiner Ausschnitt der SAP Business Packages:

- Business Package for Automotive Queries
- Business Package for Automotive Channel Management
- Business Package for Quality Inspector (mySAP ERP)
- Business Package for Supplier Collaboration
- Business Package for Key Account Management (Utilities)
- Business Package for High Tech Channel Management
- ...

Im Zuge des Projekts wurden die Branchenlösungen „Business Package for Automotive Queries“ und „Business Package for Automotive Channel Management“, evaluiert und zum Einsatz gebracht.

4.3.3 Knowledge Management

Das SAP Knowledge Management Modul (KM) bietet die Möglichkeit unstrukturierte Informationen aus verschiedenen Datenquellen zu integrieren. Die KM Plattform besteht aus zwei Subkomponenten:

- Content Management (CM)
- Search and Classification (TREX)

Mit dem SAP Modul Content Management ist es möglich, beliebige Dokumente mit ihren Eigenschaften strukturiert abzulegen. Auf diese Daten kann dann unter Verwendung der Portaloberfläche mit den Vorteilen einer Versionsverwaltung, gezielter Navigation und Berechtigungskontrolle zugegriffen werden.

TREX ermöglicht eine schnelle Suchfunktionalität im gesamten Enterprise Portal. Dies wird über angelegte und stetig aktualisierte Indizes realisiert. Somit können jegliche Art von Daten und Dokumenten unabhängig von ihrem Speicherort gefunden werden.

4.3.4 SAP xApps

Bei SAP xApps handelt es sich um eine Integrationsplattform, die es ermöglicht Daten, Webservices und weitere Komponenten unterschiedlicher Systeme miteinander zu verbinden.

Bei Großunternehmen können mit SAP xApps auch gesamte Geschäftsprozesse abgebildet und weiterentwickelt werden.

4.4 Enterprise Portal Content-Konzept

Das Portal bietet ein zentrales Content Directory (PCD), mit welchem es sehr einfach möglich ist, einer Vielzahl von Anwendern strukturiert und übersichtlich bestimmte Inhalte zuzuordnen und diese auch zu verwalten. In dieser Struktur gibt es anwenderbezogene und inhaltsbezogene Elemente. Im Detail werden folgende Komponenten unterschieden:

anwenderbezogen

- Gruppe
- Benutzer

inhaltsbezogen

- Rolle
- Workset
- Seite
- iView

Einzelne Anwender werden als Benutzer im System angelegt. Benutzer können zu Gruppen zusammengefasst werden.

Eine Gruppe steht in direktem Verhältnis zu einer oder mehreren Rollen, somit stehen alle Eigenschaften einer Rolle auch der zugeordneten Gruppe von Benutzern zur Verfügung. Einer Rolle können andere Rollen, Worksets, Seiten und iViews zugeordnet sein.

Unter einem Workset wird eine Sammlung von Aufgaben, Informationen und Services, die zu einem bestimmten Tätigkeitsbereich gehören, zusammengefasst. Diese können aus anderen Worksets, Seiten und iViews bestehen.

Eine Seite besteht aus Content, der in einem bestimmten Layout aufbereitet wurde. Dies können iViews oder andere Seiten sein.

iViews stellen die tiefste Ebene der inhaltsbezogenen Elemente dar. Unter einem iView versteht man ein Programm, das Daten aus verschiedensten Content-Quellen extrahiert und im Enterprise Portal darstellt.

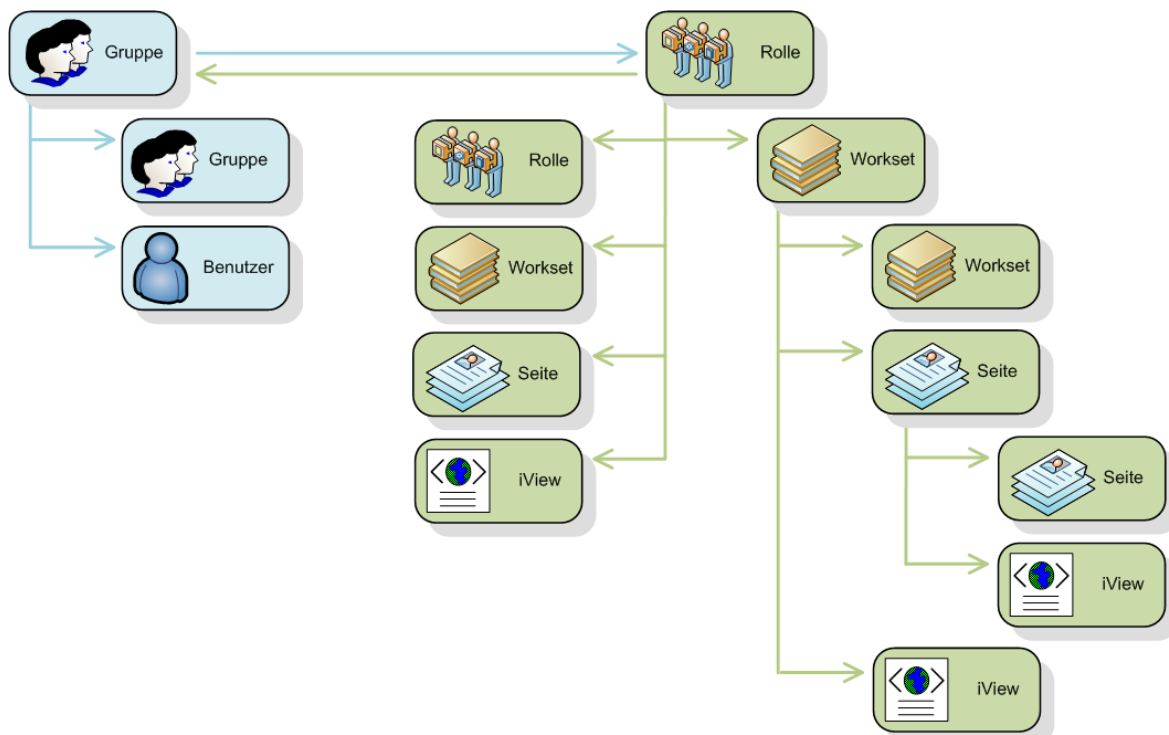


Abbildung 28: Portal Content Directory

Die Abbildung 28 stellt die Struktur der einzelnen Elemente des „Portal Content Directory“ dar und zeigt die Abhängigkeiten untereinander auf.

Die Content-Elemente, welche einer Ordnerstruktur eines Dateisystems sehr ähnlich sind, werden in Tabellen der Portaldatenbank abgelegt. Bei der Bearbeitung der gewünschten Struktur steht folgender Funktionsumfang zur Verfügung:

- Filter- und Suchmechanismen
- Anlegen von Beziehungen zwischen Objekten
- Links für die Wiederverwendung von Objektinstanzen
- Personalisierung
- Berechtigungen unter Verwendung von Zugriffskontrolllisten (ACLs)

4.5 Enterprise Portal Oberfläche

Der zentrale Einstiegspunkt für Portalanwender kann, abhängig von unterschiedlichsten Anforderungen, komplett individuell gestaltet werden. Einige Objekte und Bereiche sind in fast allen Enterprise Portalen vorhanden.

Die Abbildung 29 stellt folgende Objekte am Beispiel unseres Qualitätsmeldungsportals dar:

- Administrationsleiste
- Navigationsbereich
- Seitenbereich
- iViews

The screenshot displays the SAP Enterprise Portal interface. At the top, there is a header with the 'liXto' logo, the text 'Administrationsleiste', and a search bar. A navigation bar below the header contains links for 'Info', 'BMW', 'GM', 'Ford', 'MAN', 'Lixto Admin', and 'R/3 ITM'. The main content area is divided into three columns:

- Left Column (Status View):** A table showing the number of new messages for different OEMs.

OEM	Neue	Meldungen
BMW	2	15
GM	0	5
Ford	0	5
MAN	3	
- Middle Column (News):** Displays two news items. The first is about the GM EV1, featuring an image of the car and text describing its low drag coefficient and use of plastic. The second is about Nissan Fuel Cell Activities, featuring an image of a fuel cell vehicle and text about its development in Japan.
- Right Column (Quick Poll):** A poll asking 'Sind die letzten Änderungen am Portal für den täglichen Arbeitsbedarf eine Verbesserung?'. It shows 10 responses, with a pie chart indicating that the majority voted 'JA' (Yes).

Red annotations highlight the 'iView' components in the left and right columns, and the 'Seitenbereich' (side area) at the bottom right.

Abbildung 29: Enterprise Portal Oberfläche

In der Administrationsleiste kann der Anwender generelle Aufgabenbereiche des Portals, wie Suchfunktionen, Hilfemenüs und Personalisierung nutzen.

Der Navigationsbereich kann in mehreren Stufen (in Abbildung 29 ist er zweistufig) hierarchisch aufgebaut sein und bietet so die Möglichkeit, zur gewünschten Seite zu navigieren.

Der Seitenbereich ist vom Anwender entsprechend den Berechtigungen und zugeordneten Rollen individuell adaptierbar. Er kann aus einem bzw. mehreren iViews bestehen, welche die relevanten Informationen beinhalten.

5 Umsetzung und Realisierung

Der zur Realisierung des Anwendungsfalls in der Automobilindustrie notwendige Workflow wird in diesem Kapitel auf technischer Ebene beschrieben. Die benutzten Standardfunktionalitäten der Lixto Suite und des SAP Enterprise Portals und deren Konfiguration, sowie die zusätzlich im Zuge dieser Diplomarbeit entwickelten Softwarekomponenten sind hier dokumentiert.

Ziel dieses Abschnitts ist es, die Umsetzung der theoretischen Projektbeschreibungen aus Kapitel 2 so aufzubereiten, dass die Applikationszusammenhänge anhand eines praktischen Beispiels erläutert werden und ein anwendungsorientiertes Prozessverständnis geschaffen wird. Das Exempel beschreibt den gesamten Anwendungsfall und ist anhand der Qualitätsmeldungen von General Motors (GM), welche für Demonstrationszwecke anonymisiert wurden, nachzuvollziehen. Dabei wird auf folgende Einzelschritte näher eingegangen:

- ***Lixto Visual Wrapper Templates*** (Kapitel 5.2)
- ***Lixto Transformation Server*** (Kapitel 5.3)
- ***SAP Enterprise Portal Konfiguration*** (Kapitel 5.4)
- ***Integration des SAP R/3 Systems*** (Kapitel 5.5)

5.1 Anwendungsserver und deren Hardware-Ressourcen

Bei der Konzeption der Systemarchitektur für diesen Anwendungsfall wurde von einer gewachsenen Systemlandschaft, wie sie in fast allen Unternehmen vorliegt, ausgegangen. Systeme und Dienste, die bereits vorhanden sind werden weiterverwendet und zusätzliche Aufgaben wurden unabhängig erweitert. Damit ergibt sich technisch eine Trennung der Funktionalitäten auf Serverebene. Bei unserem Projekt unterscheiden wir folgende drei im Einsatz befindliche Server:

- ***Lixto Transformation Server***

- **SAP Enterprise Portal Server**
- **SAP R/3 (Qualitätsmodul) Server**

Die in der Tabelle 2: Hardware-Ressourcen dargestellte Tabelle zeigt die Hardware-Ressourcen der verwendeten Server, welche für unser Szenario vollkommen ausreichend waren. Bei zusätzlichen Systemanforderungen der einzelnen Server müssen diese je nach Bedarf individuell erweitert werden. Alle Server sollten über eine schnelle und ausreichend gesicherte Netzwerkverbindung miteinander verbunden sein. In unserem Fall wurde dies durch eine Kombination aus Standortgleichheit und einem verschlüsselten VPN Tunnel realisiert. Die Basiskonfiguration und weitere Betreuung der Server (Betriebssystem und Netzwerkkomponenten) wurde von den zuständigen Systemadministratoren vor Ort übernommen.

<i>Server</i>	<i>Lixto Transformation Server</i>	<i>SAP EP</i>	<i>SAP R/3</i>
<i>Ort</i>	Lixto Lab (TU)	EC3	ITM Institut (TU) via SAP Tunnel
<i>CPU</i>	Intel	Intel	Intel
<i>MHz</i>	2 x 1,2 GHz	2 x 800 MHz	2,4 GHz
<i>RAM</i>	2 GB	1 GB	2 GB
<i>Harddisk</i>	2 x 160 GB	4 x 20 GB (Raid)	80 GB
<i>IP</i>		62.116.97.30	128.130.106.141
<i>Hostname</i>	capella.dbai.tuwien.ac.at	sap.ec3.at	

Tabelle 2: Hardware-Ressourcen

5.2 Lixto Visual Wrapper Templates

Mit Hilfe des Lixto Visual Wrapper wurden Templates für unterschiedliche Automobilhersteller erstellt. Diese konnten im Transformation Server eingebunden werden, wobei die Entwicklung dieser Extraktionsvorlagen unabhängig vom Transformation Server ist und in einer beliebigen Umgebung erfolgen kann.

Im konkreten Fall wurden Templates erstellt mit denen Qualitätsmeldungen der Automotive Branche abgerufen werden können. Die unterschiedlichen Nachrichten werden von den Herstellern über gesicherte Webplattformen zur Verfügung gestellt bzw. in Einzelfällen manuell übermittelt. In unserem

ausgewählten Szenario werden Informationen von folgenden Automobilherstellern eingeholt: General Motors (GM), BMW, Ford Motor Company (Ford) und MAN Nutzfahrzeuge (MAN).

Auf den Originalwebseiten sind die Qualitätsmeldungen in den meisten Fällen reine HTML Webseiten mit Login Mechanismen und einfachen Such- und Auswahloptionen. Auch die angezeigten Strukturen sind in tabellarischer Form und maximal in zwei hierarchischen Ebenen (Übersichts- und Detailansicht) dargestellt. Die folgende Abbildung zeigt die originalen Qualitätsmeldungen der GM Webplattform.

Wenn der PRR "Hot Link" benutzt wurde, bitte die Zurück Taste des Browsers benutzen

PRR Aktivitäten

Erweiterte Liste

PRR Nummer	Lieferanten.Name	Antwort Status	Letzte Aktivität	PRR Typ	Primärer N/C	Sekundärer N/C	Teilnummer	Werks Code	DUNS	
20050807-145725	AUTOTEILE MUSTERSTÜCK CORP	NR	07-AUG-2005	Shipping	Schedule	Incomplete	15002494	OTFD	342913006	<input type="checkbox"/>
20030409-123250	Lieferant 1 SISTEMAS	NR	09-APR-2003	Kunden	Kommunikationsprobleme	GQTS OS9000 er ISO 14001 Information Not Updated		NFN OWNH	812514081	<input type="checkbox"/>
20030408-093818	Lieferant 2	IR CR PI	09-APR-2003	Qualität	Abweichung bei Zusammenbau	Teile lose oder demontiert	15047202	OXCPC	037714441	<input type="checkbox"/>
20030408-015911	Lieferant 3	NR	08-APR-2003	Qualität	Verschiedenes	Anderes	9181909	00RU	281604843	<input type="checkbox"/>
20030406-170755	Lieferant 12	NR	06-APR-2003	Qualität	Verschiedenes	Line Accumulations - Only Initial Response Required	55350966	00B0	332814130	<input type="checkbox"/>
20030404-024105	Lieferant 17	NR	04-APR-2003	Qualität	Verschiedenes	Line Accumulations - Only Initial Response Required	24436397	00RU	765808027	<input type="checkbox"/>
20030401-062752	Lieferant 17	NR	01-APR-2003	Qualität	Abweichung bei Zusammenbau	Geräusche / Quietschen / Rattern	5231220	00T4	765808027	<input type="checkbox"/>
20030321-123542	Lieferant 17	NR	21-MAR-2003	Qualität	Abweichung bei Zusammenbau	Geräusche / Quietschen / Rattern	5230750	00T4	765808027	<input type="checkbox"/>
20030318-095553	Lieferant 5	NR	18-MAR-2003	Qualität	Abweichung bei Zusammenbau	Falscher Zusammenbau	5530084	00T4	315596556	<input type="checkbox"/>
20030314-032820	Lieferant 5	NR	14-MAR-2003	Qualität	Abweichung bei Zusammenbau	Nicht gültig	5530076	00T4	315596556	<input type="checkbox"/>
20030225-094957	Lieferant 5	NR	25-FEB-2003	Kunden	Kommunikationsprobleme	Fehler bei Lieferung / überarbeiten der Lieferdaten	1	00RH	315596556	<input type="checkbox"/>
20030214-122651	Lieferant 2	IR NR	05-MAR-2003	Kunden	Rechtzeitigkeit / Endzeitpunkts Probleme	Fehler bei Lieferung der verlangten Daten	15195911	01AF	054975029	<input type="checkbox"/>
20021210-124733	Lieferant 4	NR	10-DEC-2002	Qualität	Abweichung bei Zusammenbau	Zusammenbau ausserhalb der Position	5530076	00T4	315596556	<input type="checkbox"/>
20021108-031322	Lieferant 8	NR	08-NOV-2002	Kunden	Rechtzeitigkeit / Endzeitpunkts Probleme	Fehler beim Bereitstellen der verlangten APDP Daten	12791464	00T4	317225266	<input type="checkbox"/>
20020625-125232	Lieferant 9	NR	25-JUN-2002	Anlieferung	Abweichungen bei der Daten	ASN - Nicht gesendet	88070115	SPS8	622547453	<input type="checkbox"/>
20020522-042410	Lieferant 6	NR	22-MAY-2002	Qualität	Abweichung bei Herstellung	Anderes	24460153	00RU	315596556	<input type="checkbox"/>
20020517-105841	Lieferant 2	NR	17-MAY-2002	Anlieferung	Abweichungen bei der Daten	ASN - Nicht gesendet	10423035	SPS8	037714441	<input type="checkbox"/>

Anzahl der ausgewählten Aufzeichnungen: 17

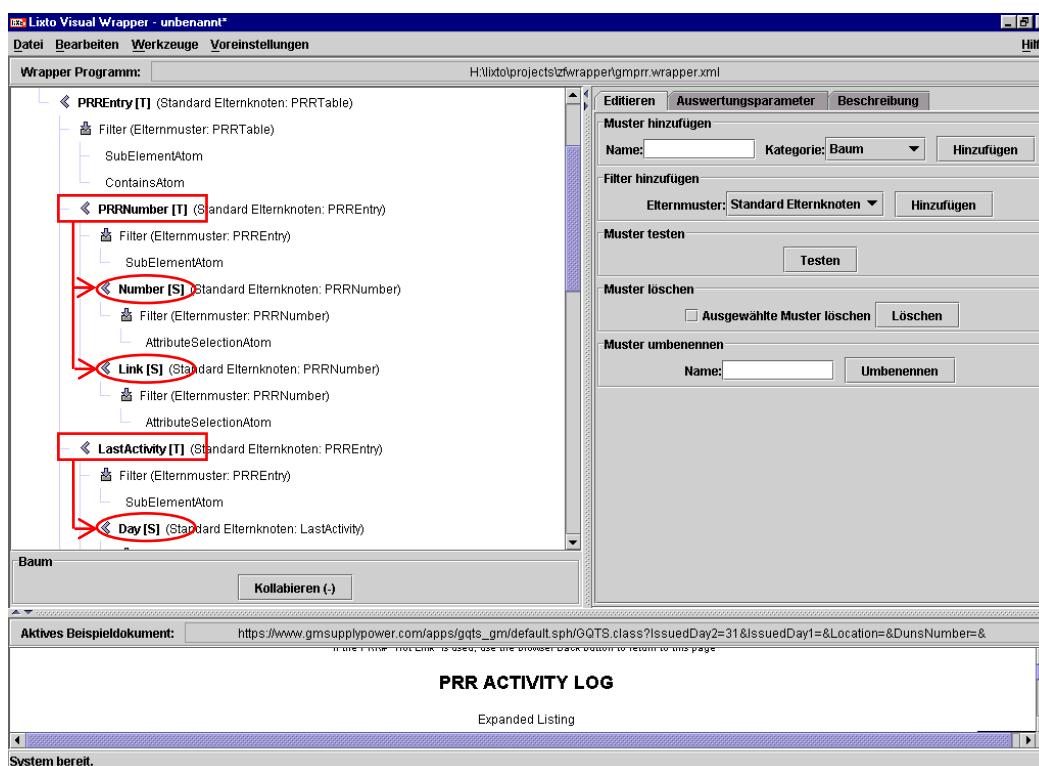
Abbildung 30: Original Qualitätsmeldungsseite

Anhand des ersten Beispieldatensatz in Abbildung 30 wird die weitere Datenaufbereitung in den unterschiedlichen Formaten erläutert. In diesem Fall besteht die Qualitätsmeldung aus folgenden Feldern:

Feld	Wert	Feld	Wert
PRR Nummer	20050807-145725	Lieferant	AUTOTEILE MUSTERSTÜCK CORP
letzte Aktivität	07-AUG-2005	Link	/apps/gqts_gm/default.sph/GQTS.class?
PRR Typ	Shipping	Primärer N/C	Schedule
Werks Code	OTFD	Sekundärer N/C	Incomplete
DUNS	342913006	Teilnummer	15002494

Tabelle 3: Felddefinition Beispielqualitätsmeldung

Im Lixto Visual Wrapper können die einzelnen Felder der Qualitätsmeldung durch bestimmte Identifikationskriterien abgegrenzt und in eine vordefinierte Struktur gebracht werden. Die Definition über Identifikationskriterien ermöglicht, dass Daten auch bei geringfügigen Änderungen der Quellseiten weiterhin extrahiert werden können.

**Abbildung 31:** Meldungsstruktur im Visual Wrapper

Die Abbildung zeigt eine Ansicht der Wrapper Oberfläche, in der die Felder der XML Struktur definiert werden. Bei unserem Beispieldatensatz wurden die Felder PRRNumber/Number (PRR Nummer), PRRNumber/Link (Link), LastActivity/Day (letzte Aktivität), LastActivity/Month (letzte Aktivität), LastActivity/Year (letzte Aktivität), Supplier (Lieferant), Type (PRR Typ), PrimaryNC (Primärer N/C), SecondaryNC (Sekundärer N/C), PartNr (Teilnummer), PlantCode (Werks Code) und DUNS (DUNS) definiert, welche den in Klammern stehenden Werten der original Qualitätsmeldungsseite entsprechen.

Das Ergebnis der Extraktionsvorlage muss pro Quellmeldung ein Datensatz auf

Ebene des PRREntry in der vordefinierten XML Struktur sein. Das XML Ergebnis wird bei der Erstellung nur rudimentären Prüfungen und Konvertierungsmechanismen unterzogen. In der nachstehenden Abbildung ist die extrahierte XML Beispielqualitätsmeldung dargestellt.

```
<rootPattern>
  <PRREntry>
    <PRRNumber>
      <Number>20030807-145725</Number>
      <Link>/apps/gqts_gm/default.sph/GQTS.class?</Link>
    </PRRNumber>
    <LastActivity>
      <Day>07</Day>
      <Month>AUG</Month>
      <Year>2005</Year>
    </LastActivity>
    <Supplier>AUTOTEILE MUSTERSTÜCK CORP</Supplier>
    <Type>Shipping</Type>
    <PrimaryNC>Schedule</PrimaryNC>
    <SecondaryNC>Incomplete</SecondaryNC>
    <PartNr>15002494</PartNr>
    <PlantCode>OTFD</PlantCode>
    <DUNS>342913006</DUNS>
  </PRREntry>
</rootPattern>
```

Abbildung 32: XML Datenstruktur

Durch das Einbinden der Lixto Visual Wrapper Templates können exakt die gleichen Ergebnisse im Transformation Server Prozess gewonnen werden. Dort finden auch erweiterte Kontrollen und Konvertierungen statt.

5.3 Lixto Transformation Server Konfiguration

Die installierte Komponente des Lixto Transformation Servers ist eine Serviceanwendung, welche nach einmaliger Konfiguration vordefinierte Prozesse automatisiert abarbeitet. Auf der Ebene eines Service Users findet die Anwendungskonfiguration und Überwachung über eine im SAP Enterprise Portal eingebundene Benutzeroberfläche (MyLixto) statt. Service Designer können ebenfalls über eine Webbenutzeroberfläche arbeiten, wobei diese in unserem Projekt als separater Zugang zum Transformation Server realisiert und nicht ins Enterprise Portal eingebunden wurde.

Auf Grund der Übersichtlichkeit und der Administrierbarkeit wurde für jeden Output Workflow eine eigene Pipe konfiguriert. Bei unserem Beispiel besteht dieser aus

einer Source Komponente, XSLT Komponenten und einer Deliverer Komponente.

Die nachfolgende Abbildung stellt die einzelnen Bestandteile unseres Beispielprozesses im Workspace dar. Auf der linken Seite im Pipe Index kann man erkennen, wie viele unterschiedliche Pipe Definitionen für die Realisierung des gesamten Anwendungsfalls notwendig sind.

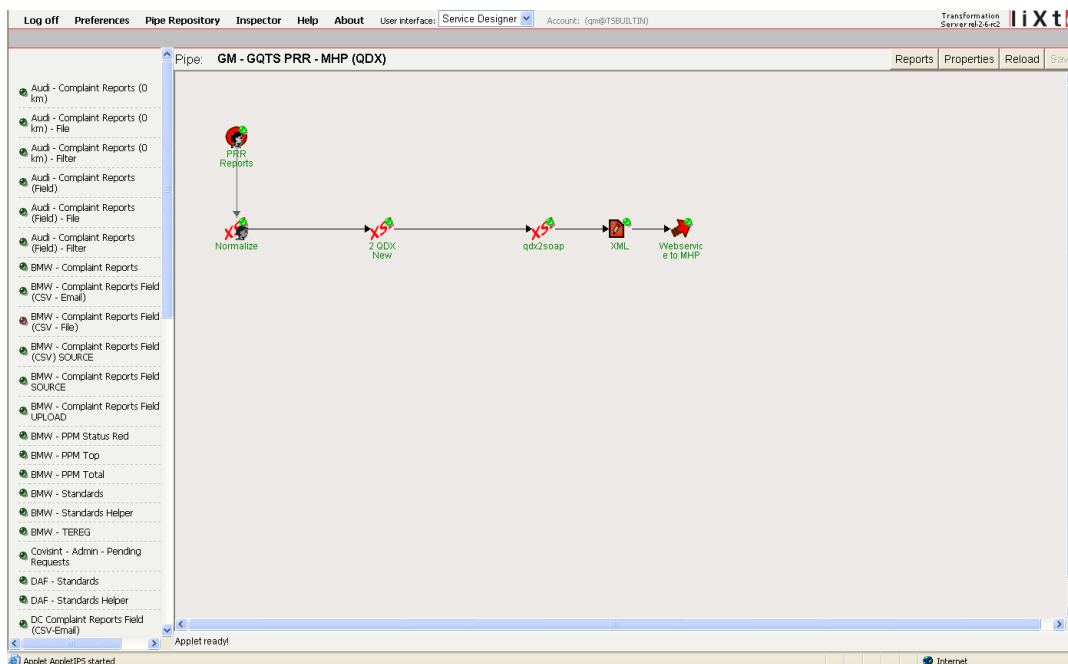


Abbildung 33: Pipe Definition

Auf Basis des eingebundenen Wrapper Templates wurde bei der Source Komponente zusätzlich noch definiert, dass es sich bei der Art der Datenquelle um Web handelt. Die Navigationssequenz, um zu den original Qualitätsmeldungen zu kommen, wurde unter der Berücksichtigung der Anmeldeparameter und diverser Suchoptionen bei den Extraktionseinstellungen definiert. Der Scheduler war so konfiguriert, dass stündlich ein Abgleich mit den Daten der Quellseite vorgenommen wurde. Bei Differenzen zur vorigen Abfrage wurde der Pipe Workflow angestoßen. Die XML Struktur der extrahierten Meldungen wurde unverändert aus der Wrapper Vorlage übernommen. Die Authentifizierung erfolgt über die Eingabe der Parameter Benutzername (USER) und Passwort (PASSWORD), welche direkt auf den angesteuerten Anmeldeseiten verwendet wurden. Somit kann eine sichere und vom Benutzer abhängige Informationsgewinnung garantiert werden. Die definierten Einstellungen können

unter „Inspect“ analysiert und überprüft werden.

Das Ergebnis der ursprünglichen Lixto Visual Wrapper Datei, welche in die oben beschriebene Source Komponente eingebunden wurde, sind Daten in der vordefinierten XML Struktur aus Abbildung 32.

Die weitere Verarbeitung der Qualitätsmeldungen passiert in den XSLT Komponenten, welche direkt mit der Source Komponente verbunden sind. In Abbildung 34: XSL Stylesheet ist exemplarisch ein Auszug aus einer XSL Stylesheet Datei zu sehen, welche die XML Daten in das gewünschte Ausgabeformat bringt und die Grundlage für den Aufbau der HTML Struktur darstellt. Dabei werden standardisierte Layout-, Sortier- und Gruppier-Algorithmen, sowie Zusatzfunktionalitäten zur Anzeige von erweiterten Informationen berücksichtigt.

```
<xsl:for-each select="rootPattern/PRREntry">
  <xsl:if test="position() mod 2 = 1">
    <tr class="1" tooltip="">
      <td>
        <xsl:element name="a">
          <xsl:attribute name="href">
            <xsl:value-of select="PRRNumber/Link" />
          </xsl:attribute>
          <xsl:value-of select="PRRNumber/Number" />
        </xsl:element>
      </td>
      <td align="left">
        <xsl:value-of select="Supplier" />
      </td>
      <td>
        <xsl:value-of select="LastActivity/Day" /> &#160;
        <xsl:value-of select="LastActivity/Month" /> &#160;
        <xsl:value-of select="LastActivity/Year" />
      </td>
      <td align="left">
        <xsl:value-of select="Type" />
      </td>
      <td align="left">
        <xsl:value-of select="PrimaryNC" />
      </td>
    </tr>
  </xsl:if>
</xsl:for-each>
```

Abbildung 34: XSL Stylesheet

Das Ergebnis der XSLT Komponenten sind HTML Webseiten, welche im SAP Enterprise Portal eingebunden sind. Die dargestellten Daten werden über die Deliverer Komponente in „real time“ Geschwindigkeit angepasst und sind somit immer am aktuellen Stand.


Bei der Deliverer Komponente wurde zusätzlich noch definiert, dass es sich bei der Art der Datenausgabe um HTML handelt. Zusätzlich müssen noch die Konfigurationen für den Speicherort, benötigte Authentifizierungen und Benutzerparameter definiert werden.

Die folgende Abbildung stellt einen Ausschnitt des automatisch vom Transformation Server generierten HTML Codes dar, welcher unserer Beispielqualitätsmeldung entspricht.

```
<tr class="1" tooltip="Tooltip">
  <td>
    <a target="_blank" href="/apps/gqts_gm/default.sph/GQTS.class?">20030807-145725</a>
  </td>
  <td align="left">AUTOTEILE MUSTERSTÜCK CORP</td>
  <td>IR</td>
  <td>07.08.2005</td>
  <td align="left">Shipping</td>
  <td align="left">Schedule</td>
</tr>
```

Abbildung 35: HTML Ausgabecode

In der Oberfläche des SAP Enterprise Portals können die generierten Qualitätsmeldungswebseiten durch die Berücksichtigung von Stylesheets an das individuelle Benutzerlayout angepasst werden. Die entsprechende Webseite des Automobilherstellers GM wird in nachstehender Bildschirmmaske abgebildet.

General Motors
[LOGIN](#)


Hersteller: GM	Lieferant: LIXTO	Neue Meldungen: 5
Lieferantennr.: 12345		letzte Änderung: 08.09.2003 11:25

PRR Nummer	Lieferantenname	Antwort	letzte Aktivität	PRR Typ	Primäre N/C
20030807-145725	AUTOTEILE MUSTERSTÜCK CORP	IR	07.08.2005	Shipping	Schedule
20030415-011	NACAM FRANCE SA	CP	27.05.2003	Quality	Coating Discrepancies
20030415-010	LIXTO GMBH LIEFERANT				Handling Damage
20030415-009	GMBH KG MUSTER				Machining
20030415-007	SYSTEM INC				Communication Issues
20030415-006	AUTO SALES CORP	IR	21.05.2003	Quality	Failure to Supply
20030415-003	MOBIL AG DEMO	NR	19.05.2003	Shipping	Electronic Communication Discrepancies
20030415-001	LIXTO GMBH	NR	24.05.2003	Quality	Timeliness / Deadline Issues

Abbildung 36: Webansicht der HTML Ausgabedatei

Die ursprünglich extrahierten Informationen der Qualitätsmeldung sind in zwei Darstellungsbereiche aufgeteilt worden; in den durch die rote Markierung gekennzeichneten primären Teil, welcher für den Anwender sofort ersichtlich ist und die Felder PRRNumber/Number (PRR Nummer), PRRNumber/Link (Link), LastActivity/Day (letzte Aktivität), LastActivity/Month (letzte Aktivität), LastActivity/Year (letzte Aktivität), Supplier (Lieferant), Type (PRR Typ) und PrimaryNC (Primärer N/C) beinhaltet. Durch die zusätzlich integrierte Funktion „Tooltip“ (siehe Anhang II.c Zusatzfunktion Tooltip) wird die Anzeige von detaillierteren Informationen der Qualitätsmeldungen in einem Hinweisfeld (schwarze Markierung) ermöglicht. Bei unserem Beispiel werden dort die vier Felder SecondaryNC (Sekundärer N/C), PartNr (Teilnummer), PlantCode (Werks Code) und DUNS (DUNS) definiert.

Die im SAP Enterprise Portal in Form von Webseiten dargestellten Informationen können vom Transformation Server auch parallel in das SAP R/3 Qualitätsmodul übernommen werden. Dieser Vorgang erfolgt nach Bestätigung der zu übernehmenden Daten durch den Anwender vollkommen automatisiert im Hintergrund. Die bereits übernommenen Daten können im SAP R/3 unternehmensspezifisch weiterverarbeitet und über einen weiteren Serverprozess im Portal abgerufen werden. Eine genaue Beschreibung dieses im Zuge der Diplomarbeit entwickelten Datenübernahme Workflows ist in Kapitel 5.5 Integration des SAP R/3 Systems zu finden.

5.4 SAP Enterprise Portal Konfiguration

In unserem Beispiel ist die zentrale Aufgabe des SAP Enterprise Portals, dem Anwender möglichst vielen Informationen automatisiert und auf sehr einfachem Weg zur Verfügung zu stellen.

Grundlage dafür ist die zentrale Anmeldung jedes Benutzers am Portal. Bei unserem Beispiel der Qualitätsmeldungen von General Motors entstehen durch die einmalige Identifikation des Anwenders (Single-Sign-On) folgende Vorteile im gesamten Ablauf des Anwendungsfalls:

Die bereitgestellten Darstellungen und damit die abrufbaren Informationen können aufgrund von Rechten und Abteilungszuordnungen unterschiedlich sein. In unserem Bereich der Qualitätsmeldungen wurden zwei Rollen angelegt, die als Vorlagen für die einzelnen Benutzerprofile von „Mitarbeitern“ und „Managern“ benutzt werden konnten. Die Rolle „Mitarbeiter“ erlaubt in unserem Fall die Ansicht und das Bearbeiten aller Daten, jedoch keine Änderungen der Portalstruktur. Benutzer mit der Rolle „Manager“ sehen nur eine Übersicht bzw. Zusammenfassung der Qualitätsauswertungen und können generelle Strukturen und allgemeine Informationen anpassen.

Die notwendigen Parameter, um Informationen von den Webseiten der Automobilhersteller abzurufen und die Verarbeitungsprozesse am Transformation Server zu überwachen, werden durch das Benutzerkonto automatisch zugeordnet. Die gleichen Parameter können auch bei der direkten LOGIN Verknüpfung (siehe Abbildung 36) zur Herstellerseite im Enterprise Portal verwendet werden und dadurch ist eine wiederholte Abfrage bzw. Eingabe von Benutzernamen und Passwort überflüssig.

Auch die Anzeige von Qualitätsmeldungen aus dem SAP R/3 System und die direkte Verknüpfung erfolgt über die Identifikation des Benutzers durch die Anmeldung am Portal.

Nach der erfolgreichen Anmeldung kommt jeder Anwender zu einer Informationsseite. Diese ist individuell zu gestalten und stellt in unserem Fall eine Zusammenfassung der Qualitätsmeldungen (Status View), eine Linksammlung (Links), einen News Bereich (News) und eine Unternehmensumfrage (Quick Poll) dar.

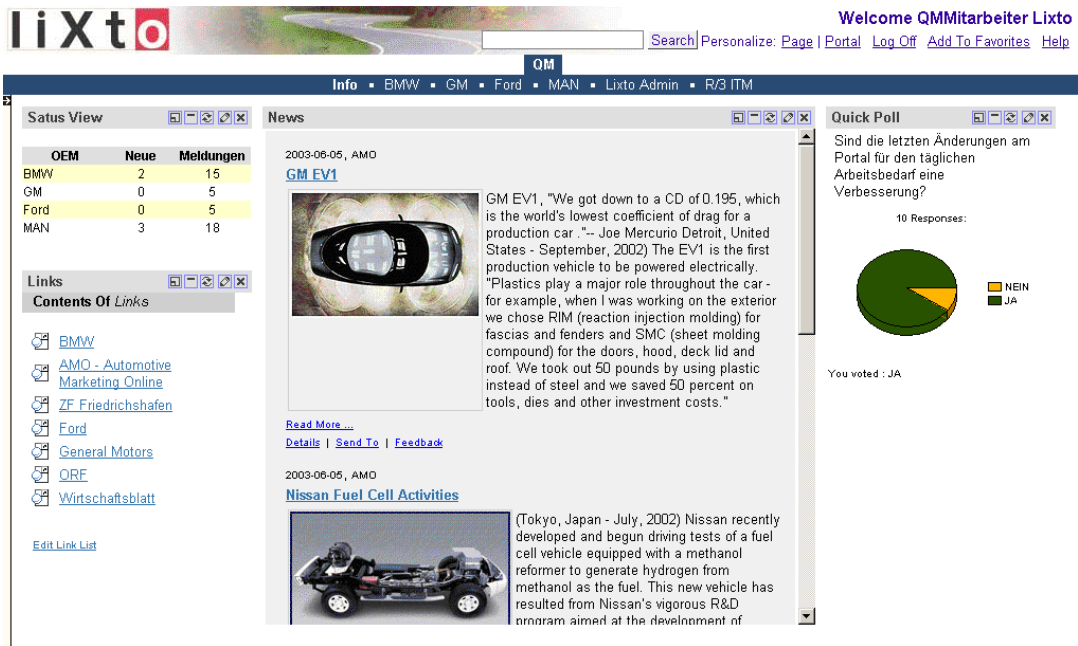


Abbildung 37: Portal Informationsseite

In der Titelleiste befinden sich die Verknüpfungen zu den Bereichen der Automobilhersteller General Motors (GM), BMW, Ford Motor Company (Ford) und MAN Nutzfahrzeuge (MAN). Zusätzlich kann die Administrationsseite der Lixto Komponenten (Lixto Admin) und der SAP R/3 Einbindung (R/3 ITM) angesteuert werden (siehe Kapitel 5.5.1).

In den nächsten Schritten soll der Workflow beschrieben werden, wie die bereits vordefinierten Qualitätsmeldungen aus unserem Beispiel von GM von einem Anwender abgerufen und abonniert werden können.

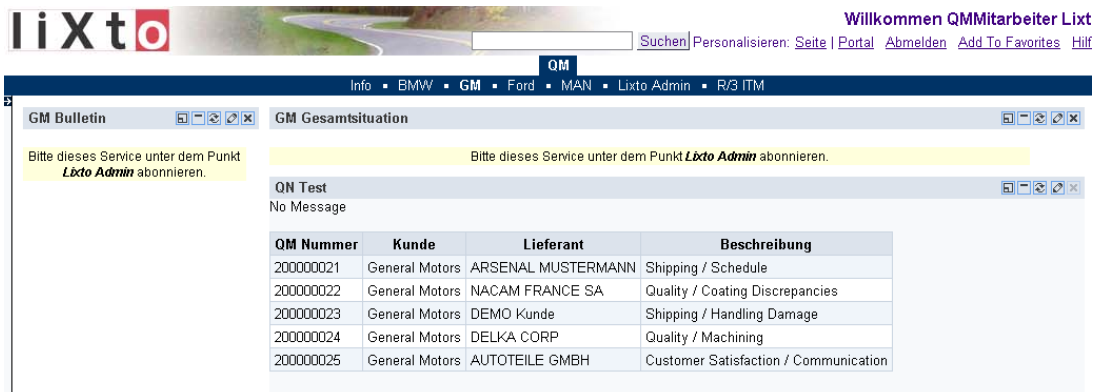


Abbildung 38: Portal General Motors (nicht abonniert)

Die Abbildung 38 zeigt die Meldungsansicht der General Motors Seite. Die Bereiche der „GM Gesamtsituation“ und der „GM Bulletin“ sind nicht mit Daten gefüllt, da die Services noch nicht abonniert wurden. Die untere Tabelle stellt interne Qualitätsmeldungsdaten aus der R/3 Systemlandschaft zur Verfügung.

Hinter dem Menüpunkt Lixto Admin wird eine Liste der aktuell verfügbaren Services angeführt. Services die grün markiert sind, wurden bereits abonniert und rot markierte Services können vom Benutzer hinzugefügt werden. Zusätzlich kann der aktuelle Status, der nächste Ausführungszeitpunkt und der Status beziehungsweise das Resultat der bereits ausgeführten Aktivitäten jedes Services kontrolliert werden.

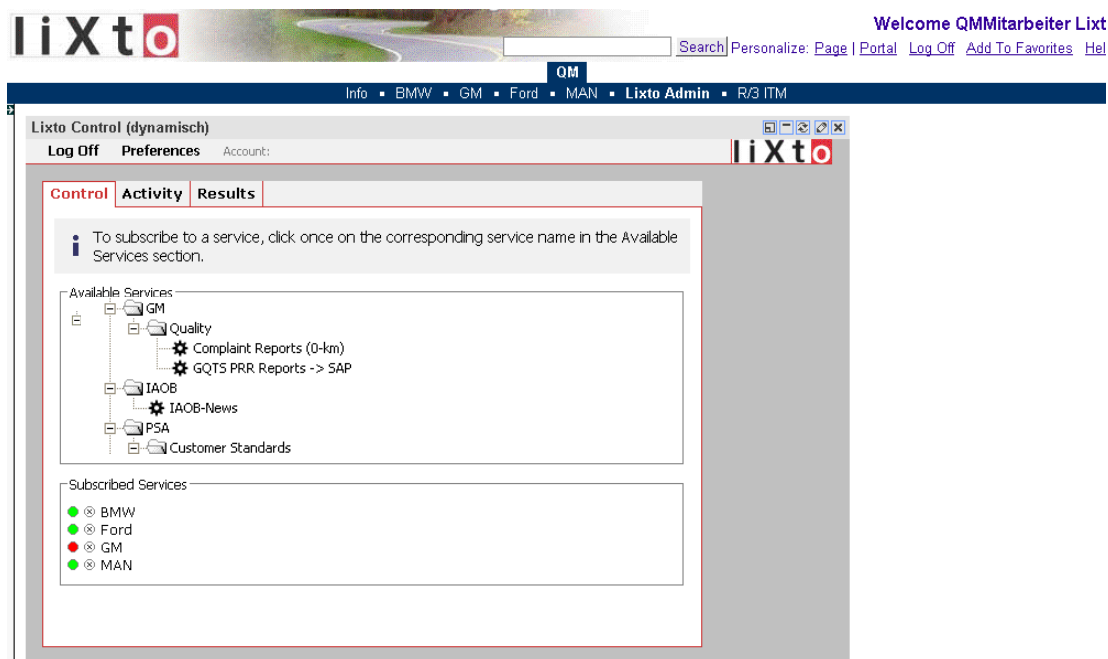


Abbildung 39: Portal Lixto Admin

Durch die Aktivierung des entsprechenden Services werden einmalig die notwendigen Benutzerparameter abgerufen. Diese werden verschlüsselt im System hinterlegt und bei jeder Serviceausführung verwendet. Auch eine Änderung kann vom Benutzer jederzeit über die Oberfläche des Portals durchgeführt werden.

Lixto Control (dynamisch) [Help](#)

Personalize

LixtoUser:

Passwort:

Info

Unique Name: lixto_admin_dyn

Created: 9/17/2003

Updated: 9/23/2003

Description: iView has no description.

[Save](#) | [Preview](#) | [Default Settings](#) |

Abbildung 40: Portal Service Benutzerparameter

Die getroffenen Einstellungen werden bei erfolgreicher Konfiguration sofort übernommen und es wird mit der ersten Durchführung des Services begonnen. In unserem Beispiel werden die aktuellsten Daten der Automobilherstellerseite von GM im entsprechenden Format dargestellt. Die Darstellung entspricht der HTML Ausgabe vom Transformation Server, welche die extrahierten und umgewandelten Originalmeldungen beinhaltet (siehe Abbildung 36).

The screenshot shows the Lixto portal interface. At the top, there is a search bar and navigation links. Below the header, there is a navigation bar with 'Info', 'BMW', 'GM', 'Ford', 'MAN', 'Lixto Admin', and 'R/3 ITM'. The main content area is divided into two panes. The left pane shows a list of GM Bulletins with columns for 'Datum' and 'Subject'. The right pane shows 'General Motors' information, including a 'LOGIN' button, a GM logo, and a table of data. The table has columns for 'PRR Nummer', 'Lieferantenname', 'Antwort', 'letzte Aktivität', 'PRR Typ', and 'Primäre NIC'. The data rows include information about suppliers like AUTOTEILE MUSTERSTÜCK CORP, NACAM FRANCE SA, LIXTO GMBH LIEFERANT, GMBH KG MUSTER, SYSTEM INC, AUTO SALES CORP, MOBIL AG DEMO, and LIXTO GMBH.

PRR Nummer	Lieferantenname	Antwort	letzte Aktivität	PRR Typ	Primäre NIC
20030415-013	AUTOTEILE MUSTERSTÜCK CORP	IR	29.05.2003	Shipping	Schedule
20030415-011	NACAM FRANCE SA	CR	27.05.2003	Quality	Coating Discrepancies
20030415-010	LIXTO GMBH LIEFERANT	NR	26.05.2003	Shipping	Handling Damage
20030415-009	GMBH KG MUSTER	NR	24.05.2003	Quality	Machining
20030415-007	SYSTEM INC	IR	22.05.2003	Customer Satisfaction	Communication Issues
20030415-006	AUTO SALES CORP	IR	21.05.2003	Quality	Failure to Supply
20030415-003	MOBIL AG DEMO	NR	19.05.2003	Shipping	Electronic Communication Discrepancies
20030415-001	LIXTO GMBH	NR	24.05.2003	Quality	Timeliness / Deadline Issues

Abbildung 41: Portal General Motors (abonniert)

Jeder hinterlegte und vordefinierte Workflow kann nach demselben Prinzip in Form eines Lixto Pipe Services abonniert werden. Der gesamte Interaktionsprozess mit dem Endanwender wird über das SAP Enterprise Portal gesteuert.

5.5 Integration des SAP R/3 Systems

Die Qualitätsmeldungsdaten sollen neben der Anzeige im Enterprise Portal auch direkt in ein angebundenes SAP R/3 System übernommen werden. Diese Funktionalität wird durch ein iView realisiert. Die einzelnen Funktionsbausteine sind so entwickelt worden, dass der Informationsaustausch nach Benutzerbestätigung im Portal eigenständig im Hintergrund ablaufen kann oder Teile des iViews vollkommen automatisiert vom Lixto Transformation Server angesteuert werden können. Zusätzlich ist es auch möglich spezielle Informationen des R/3 System auszulesen und anwendungsorientiert darzustellen.

5.5.1 Konfiguration der SAP R/3 Einbindung

Die Konfiguration der Verbindung und individueller Benutzerparameter für die Anzeige und den Datenaustausch zum R/3 Qualitätsmodul erfolgt im Enterprise Portal unter dem Punkt R/3 ITM (siehe Abbildung 37). Es handelt sich bei dieser Verbindung um eine RFC⁸ Verbindung.

The screenshot shows the 'SAP Application Transaction Launcher' configuration page. The form contains the following fields and values:

System *	TU_R3
TCode *	
SAP GUI Type	WebGui
Screen Parameter	
Process First Screen	false
OK Code Field	
Use SP01	false

Below the form is a 'Submit' button. At the bottom of the page, a red message reads: 'Properties marked with * are required'.

⁸ Im SAP Umfeld versteht man unter Remote Function Call (RFC) ein Verfahren, mit dem Funktionen in einem entfernten System aufgerufen werden können. Die RFC Kommunikation mit SAP Systemen wird über das TCP/IP Protokoll übertragen.

Abbildung 42: Portal R/3 Konfiguration

Die wichtigen Parameter der Konfiguration sind „System“ und „TCode“. Der Parameter „System“ steuert die eindeutige Verbindung zum angeschlossenen R/3 System und „TCode“ definiert die erste Transaktion, um Daten abzurufen. Die zusätzlichen Einstellungen können definiert werden, um Anzeigeoptionen zu ändern und das Mapping benutzergesteuerter Felder individuell anzupassen.

5.5.2 iView Anwendungsfälle

Über das im Zuge dieser Diplomarbeit entwickelte iView ist es möglich, verschiedene Aktionen auszulösen. Grundsätzlich wird unterschieden, ob vom Transformation Server extrahierte Daten in das R/3 System geschrieben werden oder ob Daten vom R/3 System gelesen werden. Bei den ausgelesenen Informationen kann eine Qualitätsmeldungsliste angezeigt werden. Diese kann über einzelne Qualitätsmeldungen erweitert dargestellt werden und entspricht der „Tooltip“ Funktionalität im Enterprise Portal bei der HTML Ansicht. Es stehen folgende Funktionsaufrufe zur Verfügung:

- ***Daten aus R/3 lesen*** (Kapitel 5.5.2.1)
- ***Qualitätsmeldungsliste anzeigen*** (Kapitel 5.5.2.2)
- ***Qualitätsmeldung anzeigen*** (Kapitel 5.5.2.3)
- ***Daten vom TS ins R/3 schreiben*** (Kapitel 5.5.2.4)

Grafisch lassen sich diese Funktionsbausteine in Form eines Anwendungsfalldiagramms darstellen. Als auslösende Faktoren können Benutzer, der Lixto Transformation Server und das SAP R/3 System angesehen werden. Die Funktionen selbst werden durch Ellipsen visualisiert, wobei die Pfeile mit ihren Beschriftungen die Beziehung der Funktionen untereinander und die Striche die Beziehung zwischen den auslösenden Faktoren und den einzelnen Funktionen darstellen.

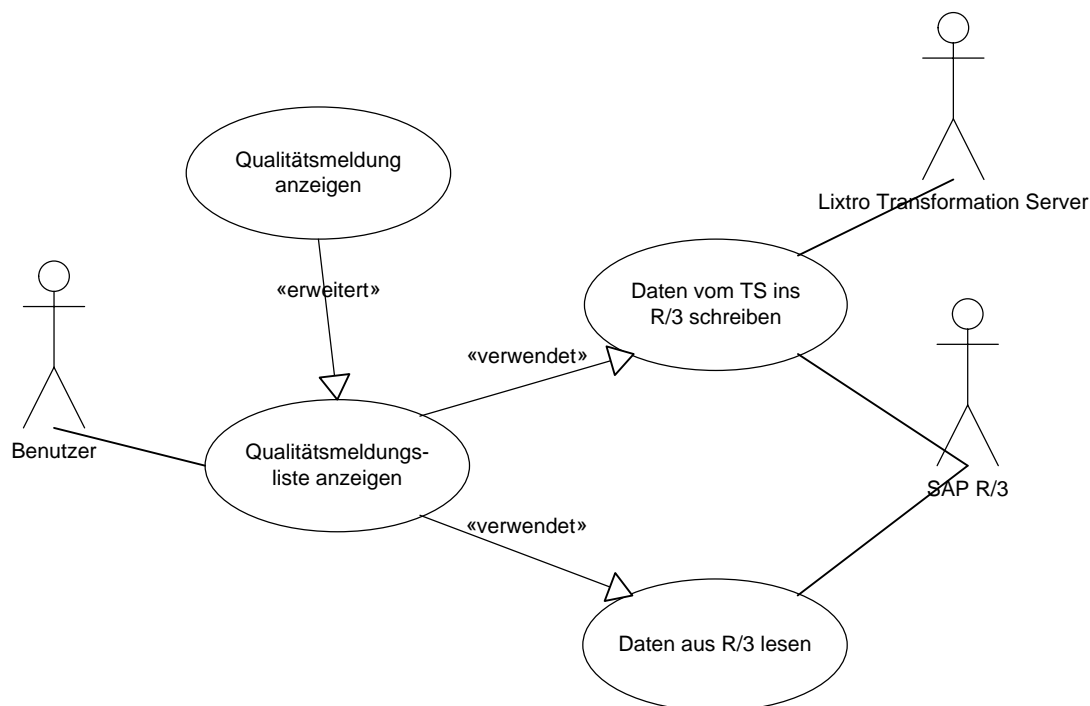


Abbildung 43: iView Anwendungsfalldiagramm

Folgende Kapitel beschreiben die einzelnen Anwendungsfälle aus Abbildung 43 näher:

5.5.2.1 Daten aus R/3 lesen

Unter der Beschreibung „Daten aus R/3 lesen“ verstehen wir eine allgemeine Funktion, welche die aktuellen Qualitätsmeldungen aus dem R/3 System ausliest. Dabei wird zuerst eine Verbindung aufgebaut und über bereitgestellte SAP BAPI⁹ Aufrufe werden die Qualitätsmeldungen abgerufen. Bei der Verbindung zum R/3 werden die vom Portal bereitgestellten Verbindungsinformationen verwendet.

Diese globale Funktion wird von den iView Programmteilen „Qualitätsmeldungsliste anzeigen“ und „Qualitätsmeldung anzeigen“ verwendet.

5.5.2.2 Qualitätsmeldungsliste anzeigen

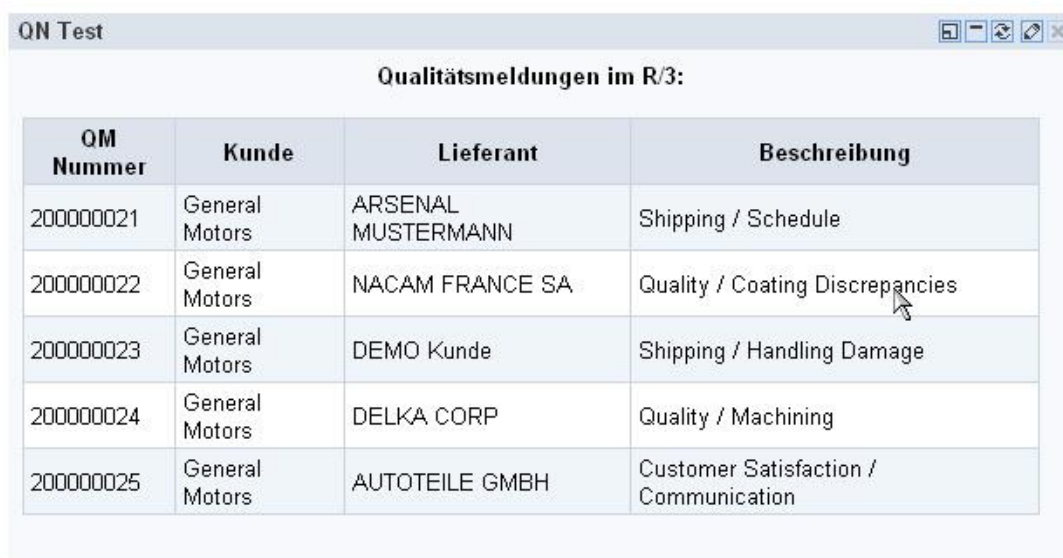
Durch die Aktivierung des iViews als Folge einer Benutzerinteraktion wird die Liste der aktiven Qualitätsmeldungen des R/3 Systems angezeigt. Das Öffnen der

⁹ BAPI steht für Business Application Programming Interface ist eine standardisierte Programmierschnittstelle der SAP-Business Objekte. BAPIs ermöglichen es externen Programmen, auf die Daten und Geschäftsprozesse des SAP R/3 zuzugreifen.

Portalseite genügt, um das entsprechende iView zu aktivieren. Dabei wird der Anwendungsfall „Daten aus R/3 lesen“ ausgeführt und das Ergebnis wird in Form einer Kurzübersicht dargestellt. Diese Übersicht beinhaltet folgende Daten:

- **Qualitätsmeldungsnummer** (NOTIFICAT)
- **Kunde** (CUSTOMER → Name)
- **Lieferant** (VENDOR → Name)
- **Qualitätsmeldungsbeschreibung** (DESCRIPTION)

Die nachstehende Abbildung stellt die ins SAP Enterprise Portal eingebundene Kurzübersicht dar.



QM Nummer	Kunde	Lieferant	Beschreibung
200000021	General Motors	ARSENAL MUSTERMANN	Shipping / Schedule
200000022	General Motors	NACAM FRANCE SA	Quality / Coating Discrepancies
200000023	General Motors	DEMO Kunde	Shipping / Handling Damage
200000024	General Motors	DELKA CORP	Quality / Machining
200000025	General Motors	AUTOTEILE GMBH	Customer Satisfaction / Communication

Abbildung 44: Qualitätsmeldungsliste

Durch das Selektieren eines einzelnen Listenelements kann der Anwendungsfall „Qualitätsmeldung anzeigen“ aufgerufen werden.

5.5.2.3 Qualitätsmeldung anzeigen

Die Selektion einer Qualitätsmeldung aus der Übersicht stellt diese mit zusätzlichen Detailinformationen dar. Anhand der eindeutigen Qualitätsmeldungsnummer können die entsprechenden Daten ausgelesen werden. Auch hier wird der Anwendungsfall „Daten aus R/3 lesen“ ausgeführt. Folgende Daten werden in der Detaildarstellung angezeigt, wobei die aktuell

angezeigten Informationen nur ein Teilausschnitt der in SAP R/3 verfügbaren Feldinformationen sind:

- **Qualitätsmeldungsnummer** (NOTIFICAT)
- **Qualitätsmeldungstyp** (NOTIF_TYPE)
- **Datum der Meldung** (NOTIFTEXT)
- **Kunde** (CUSTOMER → Name)
- **Lieferantenname** (VENDOR → Name)
- **Qualitätsmeldungsbeschreibung** (DESCRIPTION)
- **Materialteilbeschreibung** (MATERIAL → Teilbeschreibung)



Abbildung 45: Qualitätsmeldung

Über die Auswahl des „zurück“ Button aus der Detailansicht kann wieder in die übergeordnete Qualitätsmeldungsliste gewechselt werden.

5.5.2.4 Daten vom TS ins R/3 schreiben

Die Anwendungsfälle „Daten vom TS ins R/3 schreiben“ und „Daten aus R/3 lesen“ sind getrennt konzipiert und entwickelt worden, um die Übertragung der Daten vom Transformation Server ins R/3 und umgekehrt modular aufzubauen. Dies ermöglicht den autonomen Aufruf dieses Anwendungsfalls, um Daten vollkommen automatisiert vom Transformation Server in das R/3 System zu übernehmen.

Alle automatisch extrahierten und weiterverarbeiteten Qualitätsmeldungen werden über den Transformation Server im Portal dargestellt. Zusätzlich werden die im XML Format vorhandenen Meldungen über eine HTTP Verbindung vom iView eingelesen. Dort werden die einzelnen Datenfelder anhand der Mapping-Definitionen (siehe Kapitel 5.5.4) umgewandelt. Dieser Vorgang wird auch als „parsen“¹⁰ bezeichnet. Die neu aufbereiteten Qualitätsmeldungen werden über die RFC Verbindung und entsprechende BAPI Funktionsaufrufe im R/3 Qualitätsmodul angelegt.

Das abgebildete Aktivitätsdiagramm soll den internen Ablauf des Anwendungsfalls „Daten vom TS ins R/3 schreiben“ grafisch darstellen und die einzelnen Schritte bzw. einfache Kontrollmechanismen verständlich machen.

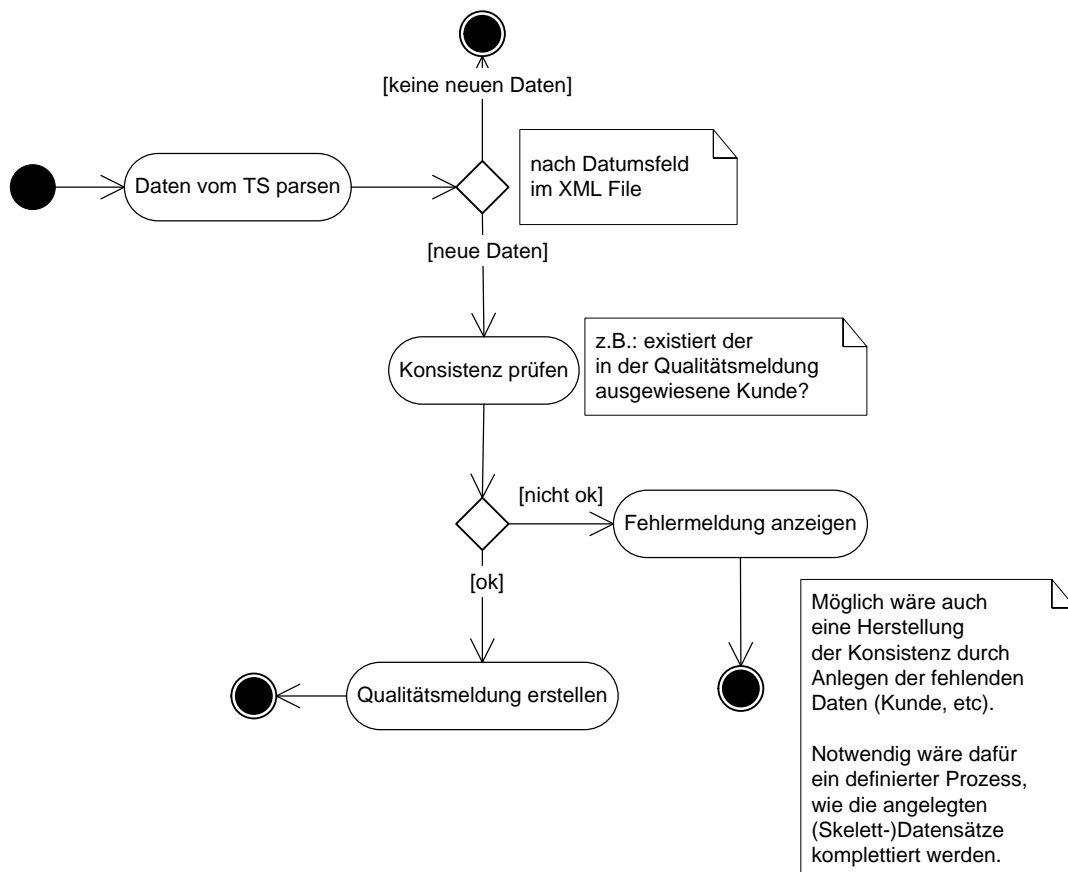


Abbildung 46: iView Aktivitätsdiagramm

¹⁰ Unter „parsen“ versteht man den technischen Vorgang, eine Eingabe in ein für die Weiterverarbeitung brauchbares Format zu zerlegen und umzuwandeln.

Im Aktivitätsdiagramm werden die zwei wichtigsten Kontrollschritte angeführt. Die eingelesenen Daten müssen auf Ihre Aktualität überprüft werden. Nur neue Daten werden weiterverarbeitet und bereits vorhandene Meldungen werden nicht übernommen. Anschließend müssen die Daten auf ihre Konsistenz geprüft werden. Dabei müssen hinterlegte Schlüsselfelder (wie Kunden- oder Lieferantenstammdaten) bereits vorhanden sein, um eine korrekte Erstellung zu gewährleisten. In der aktuell realisierten Version des iViews wird eine Fehlermeldung angezeigt, falls notwendige Referenzen fehlen. In einem weiteren Entwicklungsschritt wäre es aber durchaus realisierbar, die fehlenden Stammdaten im R/3 System automatisch erstellen zu lassen und damit die Voraussetzungen für eine korrekte Meldungsanlage zu schaffen.

5.5.3 Entwicklungsdokumentation

Bei der Entwicklung des iViews sind einige Java Klassen entstanden, welche den Datenaustausch ermöglichen. In unserem konkreten Fall sind diese mit Hilfe der Entwicklungsumgebung Eclipse¹¹ programmiert worden.

Das Klassendiagramm soll die Zusammenhänge der einzelnen Java Klassen Files beschreiben:

¹¹ Eclipse ist ein Open-Source-Framework zur Entwicklung von Software. Primär wird Eclipse als Java Entwicklungsumgebung benutzt, wobei über eine Vielzahl von Plug-ins auch andere Sprachen (z.B. C/C++, C#, Perl, PHP,...) eingebunden werden können.

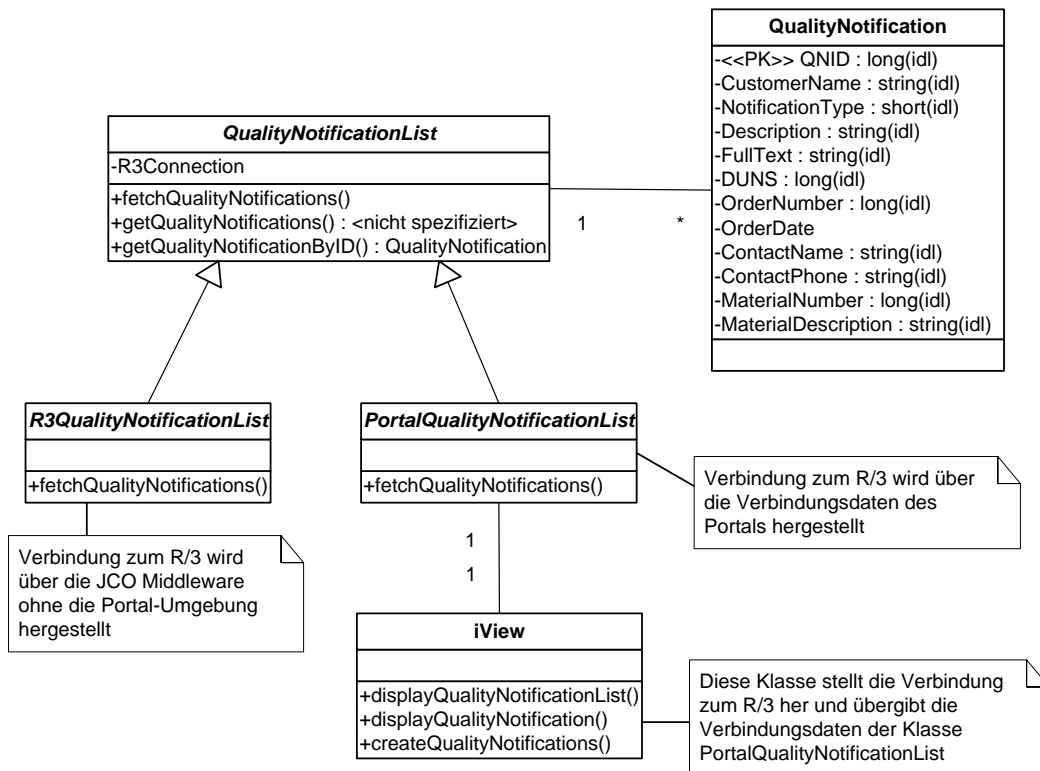


Abbildung 47: iView Klassendiagramm

In der Klasse „QualityNotification“ sind die einzelnen Felder einer Qualitätsmeldung definiert. Diese Struktur wird von der Klasse „QualityNotificationList“ verwendet um einzelne Meldungen anzuzeigen und anzulegen bzw. ganze Meldungslisten darzustellen. Diesen Aufgaben entsprechen auch die drei implementierten Operationen „fetchQualityNotifications()“, „getQualityNotifications()“ und „getQualityNotificationByID()“. Die beiden Klassen „R3QualityNotificationList“ und „PortalQualityNotificationList“ sind für die Verbindung zum R/3 System mit unterschiedlichen Parametern und die Funktionsaufrufe verantwortlich, wobei zusätzlich einzelne Parameter von der Klasse „iView“ übergeben werden.

Die beiden Java Klassen „PortalQualityNotificationList.java“ und „R3QualityNotificationList.java“ werden als Source Code Beispiele explizit im Anhang dieser Diplomarbeit (Kapitel II.a und Kapitel II.b) angeführt.

Der chronologische Ablauf der iView Anwendungsfälle kann aus technischer Sicht in Form eines Sequenzdiagramms der einzelnen Funktionsbibliotheken dargestellt

werden.

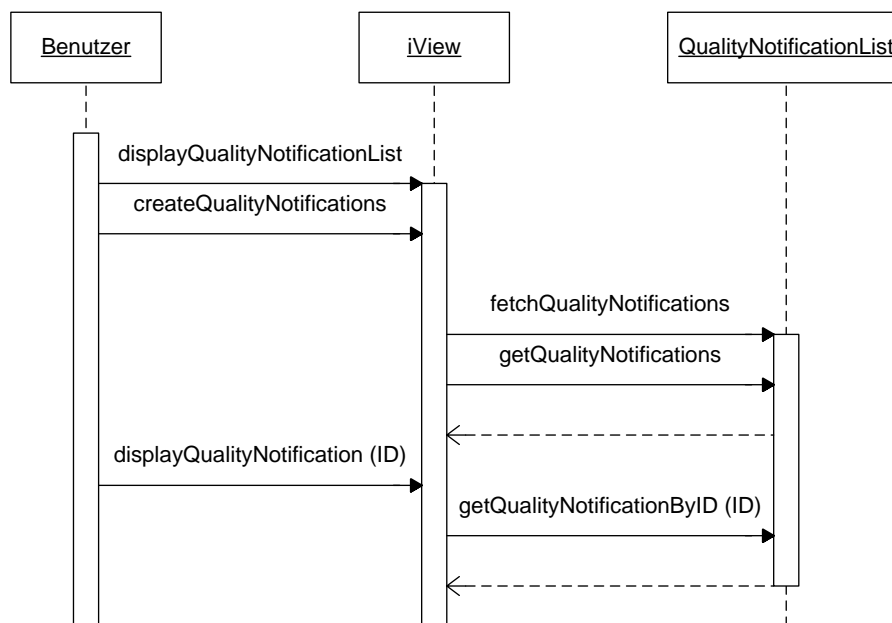


Abbildung 48: iView Sequenzdiagramm

Die Funktionen „displayQualityNotificationList“, „createQualityNotifications“ und „displayQualityNotification (ID)“ werden vom Benutzer im Enterprise Portal ausgelöst und stoßen weiter Kommunikationsaufrufe wie „fetchQualityNotifications“, „getQualityNotifications“ und „getQualityNotificationByID (ID)“ an. Die Rückgabewerte der Klasse „QualityNotificationList“ werden an die Ebene des iViews ausgegeben, von welcher auch die weiteren Schritte, wie die Anzeige im Portal, veranlasst werden.

5.5.4 Mapping der Qualitätsmeldungen

Um Qualitätsmeldungen in den Anwendungsfällen „Daten vom TS ins R/3 schreiben“ und „Daten aus R/3 lesen“ systemübergreifend übernehmen zu können, ist eine Zuordnung der extrahierten Herstellerdaten im XML Format mit den Feldern der R/3 Qualitätsmeldung notwendig.

In unserem Beispiel wird diese über eine Mapping Tabelle, welche auch Bestandteil des entwickelten iViews ist, realisiert. Nachfolgende Tabelle zeigt die

Zuordnung der einzelnen Mapping-Paare.

GM PRR AKTIVITÄTEN		R/3 QUALITÄTSMANAGEMENT			QualityNotification	
Feldbeschreibung	Typ	Feldname	Typ	Zusatz	Attribut	Typ
PRR-Nummer	#	NOTIFICAT	#	Liste	QNID («PK»)	#
Lieferanten.Name	txt	über CUSTOMER			CustomerName	txt
Antwort Status	txt	NOTIFTEXT			FullText	txt
letzte Aktivität	date	NOTIFTEXT			FullText	txt
PRR Typ	txt	NOTIF_TYPE	txt, #	Liste	NotificationType	#
Primärer N/C	txt	DESCRIPTION	txt		Description	txt
Sekundärer N/C	txt	DESCRIPTION	txt		Description	txt
Teilnummer	#	MATERIAL	txt, #	Liste	MaterialNumber	#
Werks Code	txt, #			wenn R/3 code	FactoryCode	#
DUNS	#	CUSTOMER	#	Liste	DUNS	#
Kontakt Code	txt, #	CONTACT		wenn R/3 code	ContactCode	#
Werk Kontakt	txt	über CONTACT	txt		ContactName	txt
Telefonnummer	txt	über CONTACT	txt		ContactPhone	txt
Problembeschreibung	txt	NOTIFTEXT	txt		FullText	txt
DLS/LS		NOTIFTEXT			FullText	txt
Teilbeschreibung	txt	über MATERIAL			MaterialDescription	txt
Anzahl vermutet	#	NOTIFTEXT			FullText	txt
Anzahl geprüft	#	NOTIFTEXT			FullText	txt
Anzahl Abweichungen	#	NOTIFTEXT			FullText	txt
Geschätzte Teile mit Qualitätsproblemen	#	NOTIFTEXT			FullText	txt
Lieferdatum	date	über SALES	date		OrderDate	date
Lieferscheinnummer	#	SALES_ORD	#	Liste	OrderNumber	#
Spediteur Code	txt, #	NOTIFTEXT			FullText	txt

Tabelle 4: Mapping-Tabelle der Qualitätsmeldungen

Zusammenfassung

Ziel dieser Diplomarbeit war es, bei industriellen Großunternehmen mit sehr heterogenen Systemlandschaften den Einsatz zusätzlicher Softwarelösungen zu rechtfertigen und potentielle Erleichterungen und Verbesserungen aufzuzeigen. Konkret wurden Arbeitsprozesse der Automobilindustrie, im speziellen Bereich der Qualitätsmeldungen, analysiert und in Teilbereichen durch ausgewählte Anwendungskomponenten getauscht. Als Zielstellung war nicht nur die technische Integration und Kommunikation von verschiedenen Unternehmensapplikationen von Bedeutung, sondern auch die Berücksichtigung von anwenderorientierten und ressourcenschonenden Aspekten.

Die Evaluierung dieses Workflows erfolgte in einem dreistufigen Modell, wobei folgende Varianten unterschieden wurden:

- **Qualitätsprozess ohne Adaptierung**
- **Qualitätsprozess mit Lixto Software**
- **Qualitätsprozess mit Lixto Software im SAP Enterprise Portal**

Jede dieser Varianten wurde bei der Analyse in drei weitere Teilprozesse unterteilt, um die jeweiligen Abläufe noch besser beschreiben und die zugehörigen Aufwände genauer abschätzen zu können. Die Unterteilung fällt folgendermaßen aus:

- **Navigation und Benutzerparameter**
- **Vergleich und Informationsübernahme**
- **Informationsverteilung und Rückmeldungskontrolle**

Das Ergebnis der Evaluierungsmodelle war, dass sich in unserem Anwendungsfall bereits der Einsatz einer zusätzlichen Softwarekomponente rentieren würde. Der Einsatz der Lixto Software würde bei der Bearbeitung von drei Herstellerseiten bereits ein Einsparungspotential des Aufwands von 54,6 % bedeuten. Durch die

weitere Einbindung des SAP Enterprise Portals in den Qualitätsprozess könnte der Gesamtaufwand auf 27,7 % gesenkt werden, was einem Einsparungspotentials von 72,3 % des gesamten Zeitbedarfs entspricht.

Es darf jedoch nicht vergessen werden, dass ein entsprechender Einsatz von zusätzlichen Softwarelösungen nicht nur ein Sparpotential im Bezug auf den Zeitfaktor mit sich bringt, sondern auch einen erheblichen Aufwand während der Implementierungsphase und im Zuge des Wartungsbetriebs bedeutet. Weiters wurden die Anschaffungs- und Systembetreuungskosten (Softwarelizenzen, Hardware,...) bei der Evaluierung nicht im Detail berücksichtigt.

Aus meiner Sicht rentiert sich der Einsatz beider zusätzlicher Anwendungen erst ab einer gewissen Unternehmensgröße. Diese gewährleistet einen hohen Nutzungsbedarf und heterogene Informationsstrukturen im Bereich der Qualitätsmeldungen. Bei Großunternehmen ist auch die Wahrscheinlichkeit mehrfacher Einsatzgebiete außerhalb des Qualitätsmeldungsbereichs gegeben. Dadurch kann die Rolle der zentralen Integrationsplattform von der Lixto Suite übernommen werden, und das Enterprise Portal von SAP übernimmt die Rolle der zentralen Portalanwendung.

Literatur- und Internetquellen

- [axw06] AXWAY
XIB Integration Broker Übersicht (V1.2)
Stand: 2003
- [bfg01a] Baumgartner Robert, Flesca Sergio, Gottlob Georg
The Elog Web Extraction Language
Stand: 2001
- [bfg01b] Baumgartner Robert, Flesca Sergio, Gottlob Georg
Declarative Information Extraction, Web Crawling, and Recursive Wrapping with Lixto
Stand: 2001
- [ec308] Website: *Electronic Commerce Competence Center – EC3*
Internetadresse: <http://www.ec3.at>
Stand: Oktober 2008
- [ks96] K. P. Kistner und M. Steven
Betriebswirtschaftslehre im Grundstudium
Stand: 1996
- [kap04] Kapow Technologies
RoboSuite Technical White Paper
Stand: 2004
- [lix04] Lixto Software GmbH
Lixto Transformation Server User Manual 1.18
Stand: 2004
- [lix06] Lixto Software GmbH
Lixto Visual Wrapper Tutorial 3.0
Stand: 2006
- [lix08a] Website: *Lixto Software GmbH*
Internetadresse: <http://www.lixt.com>
Stand: Oktober 2008

- [lix08b] Lixto Software GmbH
MyLixto User Manual 5.0
Stand: 2008
- [lix08c] Lixto Software GmbH
Lixto Transformation Server User Manual 5.0
Stand: 2008
- [lix08d] Lixto Software GmbH
Lixto Transformation Server Administration Manual 5.0
Stand: 2008
- [lix08e] Lixto Software GmbH
Lixto Automotive Analytics User Manual 4.8
Stand: 2008
- [lix08f] Lixto Software GmbH
Lixto Service Manager User Manual 4.8
Stand: 2008
- [lix08g] Lixto Software GmbH
Lixto Transformation Server Tutorial 5.0
Stand: 2008
- [lix08h] Lixto Software GmbH
Lixto Visual Developer User Manual 5.0
Stand: 2008
- [sap02] SAP AG
How to import Business Packages to mySAP Enterprise Portal 5.0
Stand: November 2002
- [sap05] SAP AG
EP200 SAP NetWeaver Portal System Administrator (52)
Stand: April 2005
- [sap06] SAP AG
EP200 SAP NetWeaver Portal System Administrator (62)
Stand: April 2006
- [sap08a] Website: SAP AG
Internetadresse: <http://www.sap.com>
Stand: Oktober 2008

- [sap08b] Website: *SAP AG – iView Studio*
Internetadresse: *http://www.iviewstudio.com*
Stand: Oktober 2008
- [wh04] Wiedemair Tymon, Herzog Marcus
My Lixto - TS End User Interface (V1.0/d1)
Stand: April 2004
- [wa01] Waser Alois
Test Automation – A Case Study
Stand: Oktober 2002

Anhang

I Marken

Lixto ist eine eingetragene Marke der *Lixto Software GmbH*.

SAP R/3[®], SAP Enterprise Portal[®], SAP NetWeaver, SAP Logon und SAP iView sind eingetragene Marken der *SAP AG*.

HTML, XML und XHTML sind Standards oder eingetragene Marken des *W3C[®], World Wide Web Consortium, Massachusetts Institute of Technology*.

JAVA[®] und JAVASCRIPT[®] sind eingetragene Marken der *Sun Microsystems, Inc.*

Windows 2000 Server[®], SQL Server 2000[®], Exchange Server[®] und Internet Information Service (IIS) sind eingetragene Marken der *Microsoft Corporation*.

Apache Web Server ist eine eingetragene Marke der *Apache Software Foundation*.

Alle weiteren Namen, Logos oder Produkte, die in dieser Diplomarbeit erwähnt werden, werden nur für Kennzeichnungszwecke benutzt und können eingetragene Warenzeichen ihrer jeweiligen Inhaber sein.

II Source Code

Die in folgendem Abschnitt angegebene Quellcodes (Source Codes) sind ausgewählte Beispiele, auf welche aus den projektspezifischen Kapiteln dieser Diplomarbeit verwiesen wird und die zum besseren Verständnis der technischen Umsetzung beitragen sollen.

II.a iView Integration „PortalQualityNotificationList.java“

Im Zuge des Projekts sind zur Kommunikation mit iViews einige Java-Klassen entwickelt worden, welche in Form von Eclipse-Projekten abgelegt wurden. Als Beispiel wird die Klasse „PortalQualityNotificationList.java“ angeführt, mit der Qualitätsmeldungen aus dem SAP R/3 System ausgelesen werden. Diese werden anschließend im Enterprise Portal dargestellt bzw. über den Transformation Server weiterverarbeitet.

```
package com.lixto.sap.qm;
import java.util.*;
import java.text.*;
import com.sap.mw.jco.*;

public class PortalQualityNotificationList extends
QualityNotificationList{
    private static final String SID = "CLIENTPOOL1";
    private static final int maxConn = 10;
    protected JCO.Client client = null;
    public PortalQualityNotificationList(JCO.Client client) throws
JCO.Exception {
        this.client = client;}
    public void fetchQualityNotifications(Properties params) throws
JCO.Exception, NumberFormatException {
        Hashtable CustomerNameCache = new Hashtable();
        Hashtable MaterialDescriptionCache = new Hashtable();
        JCO.Repository myRepository = new JCO.Repository("myRepository",
client);
        JCO.Function function = myRepository
            .getFunctionTemplate("BAPI_QNOTIFICAT_GETKEYFIGURES")
```

```
.getFunction();
JCO.ParameterList input = function.getImportParameterList();
JCO.ParameterList export = function.getExportParameterList();
JCO.ParameterList tables = function.getTableParameterList();
JCO.Structure kfs = input.getStructure("KEYFIGURESTRUCTURE");
kfs.setValue("1", "VERSION");
NumberFormat format = NumberFormat.getInstance();
format.setGroupingUsed(false);
format.setMaximumFractionDigits(0);
format.setMinimumIntegerDigits(4);
format.setMaximumIntegerDigits(4);
String Age=format.format(Long.parseLong(params.getProperty("Age")));
kfs.setValue(Age, "NO_DAYS");
if (params.containsKey("MaterialNumber")
    && params.containsKey("CustomerNumber")) {
    kfs.setValue("X", "MAT_CUST_I");
    format.setMinimumIntegerDigits(18);
    format.setMaximumIntegerDigits(18);
    String MaterialNumber = format.
        format(Long.parseLong(params.getProperty("MaterialNumber")));
    kfs.setValue(MaterialNumber, "MATERIAL");
    format.setMinimumIntegerDigits(10);
    format.setMaximumIntegerDigits(10);
    String CustomerNumber = format.
        format(Long.parseLong(params.getProperty("CustomerNumber")));
    kfs.setValue(CustomerNumber, "CUST_NO");
} else if (params.containsKey("MaterialNumber")
    && params.containsKey("VendorNumber")) {
    kfs.setValue("X", "MAT_VEND_I");
    format.setMinimumIntegerDigits(18);
    format.setMaximumIntegerDigits(18);
    String MaterialNumber = format.
        format(Long.parseLong(params.getProperty("MaterialNumber")));
    kfs.setValue(MaterialNumber, "MATERIAL");
    format.setMinimumIntegerDigits(10);
    format.setMaximumIntegerDigits(10);
    String VendorNumber = format.
        format(Long.parseLong(params.getProperty("VendorNumber")));
    kfs.setValue(VendorNumber, "VEND_NO");
} else if (params.containsKey("MaterialNumber")) {
    kfs.setValue("X", "MAT_INDIC");
    format.setMinimumIntegerDigits(18);
```

```
format.setMaximumIntegerDigits(18);
String MaterialNumber = format.
format(Long.parseLong(params.getProperty("MaterialNumber")));
kfs.setValue(MaterialNumber, "MATERIAL");
} else if (params.containsKey("CustomerNumber")) {
kfs.setValue("X", "CUST_INDIC");
format.setMinimumIntegerDigits(10);
format.setMaximumIntegerDigits(10);
String CustomerNumber = format.
format(Long.parseLong(params.getProperty("CustomerNumber")));
kfs.setValue(CustomerNumber, "CUST_NO");
} else if (params.containsKey("VendorNumber")) {
kfs.setValue("X", "VEND_INDIC");
format.setMinimumIntegerDigits(10);
format.setMaximumIntegerDigits(10);
String VendorNumber = format.
format(Long.parseLong(params.getProperty("VendorNumber")));
kfs.setValue(VendorNumber, "VEND_NO");
}
client.execute(function);
JCO.Table qntable = tables.getTable("LISTOFNOTIFICATIONS");
for (int i = 0; i < qntable.getNumRows(); i++) {
qntable.setRow(i);
long ID = Long.parseLong(qntable.getString("NOTIF_NO"));
QualityNotification qn = new QualityNotification(ID);
qn.setNotificationType(qntable.getString("NOTIF_TYPE"));
qn.setDescription(qntable.getString("SHORT_TEXT"));
long MaterialNumber=Long.parseLong(qntable.getString("MATERIAL"));
qn.setMaterialNumber(MaterialNumber);
qn.setDatum(qntable.getString("NOTIF_DATE"));
if (qntable.getString("CUST_NO") != null) {
String CustNo = qntable.getString("CUST_NO");
if (!CustomerNameCache.containsKey(CustNo)) {
JCO.Function custdetail =
myRepository
.getFunctionTemplate("BAPI_CUSTOMER_GETDETAIL2")
.getFunction();
JCO.ParameterList custdetail_input =
custdetail.getImportParameterList();
JCO.ParameterList custdetail_export =
custdetail.getExportParameterList();
custdetail_input.setValue(CustNo, "CUSTOMERNO");
```

```
client.execute(custdetail);
JCO.Structure custadd =
    custdetail_export.getStructure("CUSTOMERADDRESS");
String CustomerName =
    custadd.getString("NAME")
        + custadd.getString("NAME_2")
        + custadd.getString("NAME_3")
        + custadd.getString("NAME_4");
qn.setCustomerName(CustomerName);
CustomerNameCache.put(CustNo, CustomerName);
} else {
    qn.setCustomerName((String)CustomerNameCache.get(CustNo));
}
}
if (qntable.getString("MATERIAL") != null) {
    String Material = qntable.getString("MATERIAL");
    if (!MaterialDescriptionCache.containsKey(Material)) {
        JCO.Function matdetail =
            myRepository
                .getFunctionTemplate("BAPI_MATERIAL_GET_DETAIL")
                .getFunction();
        JCO.ParameterList matdetail_input =
            matdetail.getImportParameterList();
        JCO.ParameterList matdetail_export =
            matdetail.getExportParameterList();
        matdetail_input.setValue(Material, "MATERIAL");
        client.execute(matdetail);
        JCO.Structure matdata =
            matdetail_export.getStructure("MATERIAL_GENERAL_DATA");
        String Description =
            matdata.getString("MATL_DESC");
        qn.setMaterialDescription(Description);
        MaterialDescriptionCache.put(Material, Description);
    } else {
        qn.setMaterialDescription((String)MaterialDescriptionCache.get(Material));
    }
}
this.qnlist.add(qn);
}};
}
```

II.b iView Integration „R3QualityNotificationList.java“

Die zweite wichtige Java Klasse, die im Zuge dieses Projekts eingesetzt wurde ist „R3QualityNotificationList.java“. Mit ihr ist es möglich Qualitätsmeldungen vom Lixto Transformation Server in das zugehörige SAP R/3 System zu schreiben und Datenkonsistenzprüfungen durchzuführen.

```
package com.lixtosap.qm;
import com.sapportals.htmlb.page.DynPage;
import com.sapportals.htmlb.page.PageException;
import com.sapportals.portal.htmlb.page.PageProcessorComponent;
import com.sapportals.htmlb.*;
import com.sapportals.htmlb.event.*;
import com.sapportals.htmlb.table.*;
import com.sapportals.htmlb.enum.*;
import com.sapportals.portal.prt.component.*;
import com.sapportals.portal.prt.service.jco.*;
import com.sap.mw.jco.*;
import java.util.*;
import java.net.*;
import java.io.*;

public class R3QualityNotificationList extends PageProcessorComponent {
    public DynPage getPage() {
        return new R3QualityNotificationDisplayDynPage();
    }
    public static class R3QualityNotificationDisplayDynPage extends DynPage
    {
        private static boolean QNotifsAdded = false;
        private long ID = 0;
        private String rs = "";
        private PortalQualityNotificationList qnl = null;
        private String debug = "No Message";
        public void doInitialization() {
            retrieveData();
        }
        public void doProcessAfterInput() throws PageException {}
        public void doProcessBeforeOutput() throws PageException {
            int control = 1;
            try {
                URL myURL = new URL("http", "capella.lixtosap.tuwien.ac.at",
```

```
        "/tsout/auto/QMMITARBEITER/iview_info.txt");
    InputStream myStream = myURL.openStream();
    byte[] myArray = new byte[myStream.available()];
    myStream.read(myArray);
    String myString = new String(myArray);
    myString = myString.substring(
        myString.indexOf("<body>") + 6 ,
        myString.indexOf("</body>")
    ).trim();
    debug = myString;
    control = Integer.parseInt(myString);
} catch (MalformedURLException murle) {
    control = 1;
} catch (IOException ioe) {
    control = 1;}
switch (control) {
    case 0:
        if (ID == 0) showList(0);
        else showDetail();
        break;
    case 1:
        if (ID == 0) showList(1);
        else showDetail();
        break;
    case 2:
        if (!QNotifsAdded) addR3QMEL();
        if (ID == 0) showList(1);
        else showDetail();
        break;}}

public void retrieveData() {
    IPortalComponentRequest myRequest = (IPortalComponentRequest)
    this.getRequest();
    ResourceBundle myResource = myRequest.getResourceBundle();
    IPortalComponentProfile myProfile =
    myRequest.getComponentContext().getProfile();
    IJCOClientService clientService = (IJCOClientService)
    myRequest.getService(IJCOClientService.KEY);
    IJCOClientPoolEntry poolEntry =
    clientService.getJCOClientPoolEntry("TU_JCO", myRequest);
    JCO.Client client = poolEntry.getJCOClient();
    client.connect();
    Properties param = new Properties();
```

```
param.setProperty("Age", myProfile.getProperty("Age"));
if (!myProfile.getProperty("CustomerNumber").equals("ignore"))
    param.setProperty("CustomerNumber",
        myProfile.getProperty("CustomerNumber"));
if (!myProfile.getProperty("MaterialNumber").equals("ignore"))
    param.setProperty("MaterialNumber",
        myProfile.getProperty("MaterialNumber"));
if (!myProfile.getProperty("VendorNumber").equals("ignore"))
    param.setProperty("VendorNumber",
        myProfile.getProperty("VendorNumber"));
qnl = new PortalQualityNotificationList(client);
qnl.fetchQualityNotifications(param);
client.disconnect();
poolEntry.release();}

public void showList(int control) {
    Form myForm = this.getForm();
    TableView QNTable = new TableView("QMEL_LISTE");
    Vector caption = new Vector();
    caption.addElement("QM Nummer");
    caption.addElement("Kunde");
    caption.addElement("Lieferant");
    caption.addElement("Beschreibung");
    retrieveData();
    Vector qndata = qnl.getQualityNotificationList();
    Vector data = new Vector();
    if (control != 0) {
        int max = (qndata.size() <= 8) ? qndata.size() : 8;
        for (int i = 0; i < max; i++) {
            Vector row = new Vector();
            row.add(Long.toString(((QualityNotification)
                qndata.elementAt(i)).getID()));
            row.add("General Motors");
            row.add(((QualityNotification)qndata.elementAt(i))
                .getCustomerName());
            row.add(((QualityNotification)qndata.elementAt(i))
                .getDescription());
            data.add(row);}}
    TableViewModel model = new DefaultTableViewModel(data, caption);
    QNTable.setModel(model);
    QNTable.setDesign(TableViewDesign.ALTERNATING);
    QNTable.setHeaderVisible(true);
    QNTable.setFooterVisible(false);
```

```
QNTable.setFillUpEmptyRows(true);
QNTable.setSelectionMode(TableSelectionMode.NONE);
QNTable.setOnCellClick(1, "selectDetail");
QNTable.setOnCellClick(2, "selectDetail");
QNTable.setOnCellClick(3, "selectDetail");
QNTable.setOnCellClick(4, "selectDetail");
GridLayout grid = new GridLayout(2,1);
grid.setCellPadding(8);
GridLayoutCell cell11 = new GridLayoutCell("cell11");
GridLayoutCell cell21 = new GridLayoutCell("cell21");
cell11.setHAlignment(CellHAlign.CENTER);
cell21.setHAlignment(CellHAlign.CENTER);
TextView tv_header = new TextView("Qualitätsmeldungen im R/3:");
tv_header.setDesign(TextViewDesign.EMPHASIZED);
cell11.setContent(tv_header);
cell21.setContent(QNTable);
grid.addCell(1,1,cell11);
grid.addCell(2,1,cell21);
form.addComponent(grid);    }
public void showDetail() {
    Form myForm = this.getForm();
    Button back = new Button("back");
    back.setText("zurück");
    back.setOnClickListener("selectList");
    TextView tv_ID = new TextView();
    TextView tv_NotificationType = new TextView();
    TextView tv_Customer = new TextView();
    TextView tv_Vendor = new TextView();
    TextView tv_Datum = new TextView();
    ScrollContainer sc_Description = new ScrollContainer("Description");
    sc_Description.setHeight("30");
    ScrollContainer sc_FullText = new ScrollContainer("FullText");
    sc_FullText.setHeight("100");
    ScrollContainer sc_Material = new ScrollContainer("Material");
    sc_Material.setHeight("30");
    QualityNotification qn = qnl.getQualityNotificationByID(ID);
    tv_ID.setText(Long.toString(qn.getID()));
    tv_NotificationType.setText("Kundenreklamation");
    tv_Customer.setText("General Motors");
    tv_Vendor.setText(qn.getCustomerName());
    tv_Datum.setText(qn.getDatum());
    sc_Description.addText(qn.getDescription());
```



```
sc_FullText.addText("Nicht verfügbar!");
sc_Material.addText(qn.getMaterialDescription());
GridLayout layout = new GridLayout(3,1);
layout.setCellSpacing(5);
GridLayout top = new GridLayout(1,2);
top.setCellSpacing(5);
top.setWidth("100%");
GridLayout top_left = new GridLayout(2,2);
top_left.setWidth("50%");
top_left.setCellSpacing(5);
GridLayoutCell tl_11 = new GridLayoutCell("top_left_11");
GridLayoutCell tl_12 = new GridLayoutCell("top_left_12");
GridLayoutCell tl_21 = new GridLayoutCell("top_left_21");
GridLayoutCell tl_22 = new GridLayoutCell("top_left_22");
GridLayoutCell tl_31 = new GridLayoutCell("top_left_31");
GridLayoutCell tl_32 = new GridLayoutCell("top_left_32");
TextView label_Meldungsnummer = new TextView("Meldungsnummer:");
label_Meldungsnummer.setDesign(TextViewDesign.EMPHASIZED);
tl_11.setContent(label_Meldungsnummer);
tl_12.setContent(tv_ID);
TextView label_Meldungstyp = new TextView("Meldungstyp:");
label_Meldungstyp.setDesign(TextViewDesign.EMPHASIZED);
tl_21.setContent(label_Meldungstyp);
tl_22.setContent(tv_NotificationType);
TextView label_Meldungsdatum = new TextView("Meldungsdatum:");
label_Meldungsdatum.setDesign(TextViewDesign.EMPHASIZED);
tl_31.setContent(label_Meldungsdatum);
tl_32.setContent(tv_Datum);
tl_11.setVAlignment(CellVAlign.TOP);
tl_12.setVAlignment(CellVAlign.TOP);
tl_21.setVAlignment(CellVAlign.TOP);
tl_22.setVAlignment(CellVAlign.TOP);
tl_31.setVAlignment(CellVAlign.TOP);
tl_32.setVAlignment(CellVAlign.TOP);
top_left.addCell(1, 1, tl_11);
top_left.addCell(1, 2, tl_12);
top_left.addCell(2, 1, tl_21);
top_left.addCell(2, 2, tl_22);
top_left.addCell(3, 1, tl_31);
top_left.addCell(3, 2, tl_32);
GridLayout top_right = new GridLayout(2,2);
top_right.setWidth("50%");
```

```
top_right.setCellSpacing(5);
GridLayoutCell tr_11 = new GridLayoutCell("top_right_11");
GridLayoutCell tr_12 = new GridLayoutCell("top_right_12");
GridLayoutCell tr_21 = new GridLayoutCell("top_right_21");
GridLayoutCell tr_22 = new GridLayoutCell("top_right_22");
tr_11.setVAlignment(CellVAlign.TOP);
tr_12.setVAlignment(CellVAlign.TOP);
tr_21.setVAlignment(CellVAlign.TOP);
tr_22.setVAlignment(CellVAlign.TOP);
TextView label_Kunde = new TextView("Kunde:");
label_Kunde.setDesign(TextViewDesign.EMPHASIZED);
tr_11.setContent(label_Kunde);
tr_12.setContent(tv_Customer);
TextView label_Lieferant = new TextView("Lieferant:");
label_Lieferant.setDesign(TextViewDesign.EMPHASIZED);
tr_21.setContent(label_Lieferant);
tr_22.setContent(tv_Vendor);
top_right.addCell(1, 1, tr_11);
top_right.addCell(1, 2, tr_12);
top_right.addCell(2, 1, tr_21);
top_right.addCell(2, 2, tr_22);
GridLayoutCell t_11 = new GridLayoutCell("top_11");
GridLayoutCell t_12 = new GridLayoutCell("top_12");
t_11.setContent(top_left);
t_12.setContent(top_right);
t_11.setVAlignment(CellVAlign.TOP);
t_12.setVAlignment(CellVAlign.TOP);
top.addCell(1, 1, t_11);
top.addCell(1, 2, t_12);
GridLayout center = new GridLayout (2,2);
center.setWidth("100%");
center.setCellSpacing(10);
GridLayoutCell c_11 = new GridLayoutCell("center_11");
GridLayoutCell c_12 = new GridLayoutCell("center_12");
GridLayoutCell c_21 = new GridLayoutCell("center_12");
GridLayoutCell c_22 = new GridLayoutCell("center_22");
GridLayoutCell c_31 = new GridLayoutCell("center_13");
GridLayoutCell c_32 = new GridLayoutCell("center_23");
TextView label_Kurzbeschreibung = new TextView("Beschreibung:");
label_Kurzbeschreibung.setDesign(TextViewDesign.EMPHASIZED);
c_11.setContent(label_Kurzbeschreibung);
c_12.setContent(sc_Description);
```

```
TextView label_Material = new TextView("Materialbeschreibung:");
label_Material.setDesign(TextViewDesign.EMPHASIZED);
c_21.setContent(label_Material);
c_22.setContent(sc_Material);
c_11.setVAlignment(CellVAlign.TOP);
c_12.setVAlignment(CellVAlign.TOP);
c_21.setVAlignment(CellVAlign.TOP);
c_22.setVAlignment(CellVAlign.TOP);
center.addCell(1, 1, c_11);
center.addCell(1, 2, c_12);
center.addCell(2, 1, c_21);
center.addCell(2, 2, c_22);
FlowLayout bottom = new FlowLayout();
bottom.addComponent(back);
GridLayoutCell layout_11 = new GridLayoutCell("layout_11");
GridLayoutCell layout_21 = new GridLayoutCell("layout_21");
GridLayoutCell layout_31 = new GridLayoutCell("layout_31");
layout_11.setContent(top);
layout_21.setContent(center);
layout_31.setContent(bottom);
layout.addCell(1, 1, layout_11);
layout.addCell(2, 1, layout_21);
layout.addCell(3, 1, layout_31);
form.addComponent(layout);}

public void selectDetail(Event e) {
    TableCellClickEvent tcce = (TableCellClickEvent)e;
    if (qnl == null) {
        retrieveData(); }
    ID = ((QualityNotification)qnl.getQualityNotificationList()
        .elementAt(tcce.getVisibleRowIndex() - 1)).getID();}

public void selectList(Event e) {
    ID = 0;}

public void addR3QMEL() {
    IPortalComponentRequest myRequest =
        (IPortalComponentRequest) this.getRequest();
    IJCOClientService clientService = (IJCOClientService)
        myRequest.getService(IJCOClientService.KEY);
    IJCOClientPoolEntry poolEntry =
        clientService.getJCOClientPoolEntry("TU_JCO", myRequest);
    JCO.Client client = poolEntry.getJCOClient();
    client.connect();
    JCO.Repository myRepository = new JCO.Repository("myRepository",
```

```
client);
JCO.Function function =
    myRepository
        .getFunctionTemplate("BAPI_QNOTIFICAT_CREATE")
        .getFunction();
JCO.ParameterList input = function.getImportParameterList();
JCO.Structure notifheader = input.getStructure("NOTIF_HEADER");
notifheader.setValue("Q1", "NOTIF_TYPE");
notifheader.setValue("000000000000000091", "MATERIAL");
notifheader.setValue("MUSTER", "CUSTOMER");
notifheader.setValue("Quality / Failure to Supply", "DESCRIPT");
client.execute(function);
notifheader.setValue("Q1", "NOTIF_TYPE");
notifheader.setValue("000000000000000091", "MATERIAL");
notifheader.setValue("ELEMENTS", "CUSTOMER");
notifheader.setValue("Shipping / Electronic Communication
Discrepancies", "DESCRIPT");
client.execute(function);
notifheader.setValue("Q1", "NOTIF_TYPE");
notifheader.setValue("000000000000000091", "MATERIAL");
notifheader.setValue("AUTOTEILE", "CUSTOMER");
notifheader.setValue("Quality / Timeliness Issues ", "DESCRIPT");
client.execute(function);
client.disconnect();
poolEntry.release();
QNotifsAdded = true;
}
}
}
```

II.c Zusatzfunktion Tooltip

Die Zusatzfunktion Tooltip bietet bei der Anzeige von Qualitätsnachrichten die Möglichkeit, weitere Informationen zu den einzelnen Datensätzen anzuzeigen. Bei Tooltip handelt es sich um Javaskript, welches über den XSL Sourcecode in den Transformation Server Prozess eingebunden (tooltip.js) wird und weiter Informationen im HTML Format verarbeiten kann.

```
var delayTime = 700;
var showTime = 5000;
var tooltipDefaultStyle = " background: infobackground;
                           color: infotext;
                           font: statusbar;
                           padding: 1;
                           border: 1 solid black;
                           position: absolute;
                           z-index: 99;
                           visibility: hidden;
                           ";
var tooltipStart = "<table id=\"internalTooltipSpan\" cellspacing=0
  cellpadding=0 style=\"\" + tooltipDefaultStyle + \"\"><tr><td>";
var tooltipEnd = "</td></tr></table>";
var showTimeout;
var hideTimeout;
var shown = false;
var x;
var y;

function getReal(el) {
  temp = el;
  while ((temp != null) && (temp.tagName != "BODY")) {
    if (temp.getAttribute("tooltip")) {
      el = temp;
      return el;
    }
    temp = temp.parentElement;
  }
  return el;
}

function document.onmousemove() {
  x = window.event.x;
  y = window.event.y;
}

function document.onmouseover() {
  fromEl = getReal(event.fromElement);
  toEl = getReal(event.toElement);
  if ((toEl.getAttribute("tooltip")) && (toEl != fromEl)) {
    showTimeout = window.setTimeout("displayTooltip(toEl)", delayTime);
  }
}

function document.onmouseout() {
```

```
fromEl = getReal(event.fromElement);
toEl = getReal(event.toElement);
if ((fromEl.getAttribute("tooltip")) && (toEl != fromEl)) {
    window.clearTimeout(showTimeout);
    hideTooltip();
}
}

function displayTooltip(el) {
    if (!document.all.internalTooltipSpan) {
        document.body.insertAdjacentHTML("BeforeEnd", tooltipStart +
            el.getAttribute("tooltip") + tooltipEnd);
    }
    else {
        internalTooltipSpan.outerHTML = tooltipStart +
            el.getAttribute("tooltip") + tooltipEnd;
    }
    var toolStyle = el.getAttribute("tooltipstyle");
    if (toolStyle != null) {
        internalTooltipSpan.style.cssText = tooltipDefaultStyle + toolStyle;
    }
    internalTooltipSpan.style.left = x - 3;
    internalTooltipSpan.style.top = y + 20;
    dir = getDirection();
    if (typeof(swipe) == "function")
        window.setTimeout("swipe(internalTooltipSpan, dir);",1);
    else if (typeof(fade) == "function")
        window.setTimeout("fade(internalTooltipSpan, true, 5)",1);
    else
        internalTooltipSpan.style.visibility = "visible";
    shown = true;
    hideTimeout = window.setTimeout("hideTooltip()", showTime);
}

function hideTooltip() {
    if (shown) {
        window.clearTimeout(hideTimeout);
        internalTooltipSpan.style.visibility = "hidden";
        shown = false;
    }
}

function getDirection() {
    var pageWidth, pageHeight, scrollTop;
    pageHeight = document.body.clientHeight;
    pageWidth = document.body.clientWidth;
    toolTipTop = internalTooltipSpan.style.pixelTop;
    toolTipLeft = internalTooltipSpan.style.pixelLeft;
    toolTipHeight = internalTooltipSpan.offsetHeight;
    toolTipWidth = internalTooltipSpan.offsetWidth;
    scrollTop = document.body.scrollTop;
    scrollLeft = document.body.scrollLeft;
    if (toolTipWidth > pageWidth)
        internalTooltipSpan.style.left = scrollLeft;
    else if (toolTipLeft + toolTipWidth - scrollLeft > pageWidth)
        internalTooltipSpan.style.left = pageWidth - toolTipWidth +
            scrollLeft;
    if (toolTipTop + toolTipHeight - scrollTop > pageHeight) {
        internalTooltipSpan.style.top = toolTipTop - toolTipHeight - 22;
        return 8;
    }
    return 2;
}
```