

February 1999

Reverse Business Engineering - Modelle aus produktiven R/3-Systemen ableiten

Andreas Hufgard

IBIS Prof. Thome GmbH, Würzburg, hufgard@ibis-thome.de

Heike Wenzel-Däfler

Universität Würzburg, hwenzel@prothome.de

Follow this and additional works at: <http://aisel.aisnet.org/wi1999>

Recommended Citation

Hufgard, Andreas and Wenzel-Däfler, Heike, "Reverse Business Engineering - Modelle aus produktiven R/3-Systemen ableiten" (1999). *Wirtschaftsinformatik Proceedings 1999*. 23.

<http://aisel.aisnet.org/wi1999/23>

This material is brought to you by the Wirtschaftsinformatik at AIS Electronic Library (AISeL). It has been accepted for inclusion in Wirtschaftsinformatik Proceedings 1999 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Reverse Business Engineering – Modelle aus produktiven R/3-Systemen ableiten

Andreas Hufgard

IBIS Prof. Thome GmbH, Würzburg (hufgard@ibis-thome.de)

Heike Wenzel-Däfler

Universität Würzburg (hwenzel@proftHOME.de)

Inhalt

- 1 Reverse Engineering für Softwarebibliotheken wie SAP R/3**
 - 1.1 Klassisches Reverse Engineering
 - 1.2 Dokumentation der Einführung, Anwendung und dynamischen Anpassung
- 2 Reverse Business Engineering-Ansatz**
 - 2.1 Forward versus Reverse Business Engineering
 - 2.2 Modellelementermittlung und -verdichtung
 - 2.3 Empirische Datenbank und Analysemöglichkeiten
 - 2.4 Mustererkennung und fallbasiertes Schließen
- 3 RBE-Toolentwicklung für SAP R/3**
 - 3.1 Reverse Modelling Assistant (RMA)
 - 3.2 Anwendungsmöglichkeiten des RMA

Abstract

Je mehr sich betriebswirtschaftliche Softwarebibliotheken, wie bspw. R/3 von SAP durchsetzen, desto mehr vernimmt man auch "Klagen" der Anwender: komplexe und teure Einführungsprozesse, fehlende Übersicht, mangelndes Wissen über die Arbeitsweise des Systems und Schwierigkeiten bei der Anpassung werden regelmäßig auf der Negativliste angeführt.

Während für die erst genannten Probleme bereits einige methodische und toolgestützte Hilfestellungen angeboten werden, haben sich bisher weder die Theorie noch die Praxis eingehender mit der effizienten Nutzung und Adaption bereits implementierter Softwarebibliotheken beschäftigt. Da aber gerade eine kontinuierliche Anpassung an veränderte Umweltbedingungen zu den zentralen Erfolgsfaktoren einer effektiven Softwarenutzung zählt, erscheint es äußerst lohnend, den Anwender dabei durch ein methodisch durchgängiges und einfach handhabbares Werkzeug zu unterstützen.

Bei der von den Autoren entwickelten Vorgehensweise des Reverse Business Engineering werden produktive R/3-Systeme toolgestützt analysiert, um retrograd ein Modell der aktiv genutzten Elemente zu ermitteln und darzustellen. Nach Durchführung einer Vielzahl derartiger Projekte, kann eine Erfahrungsdatenbank aufgebaut werden, die dem Softwarehersteller hilft, sein System kundengerechter zu gestalten. Darüber hinaus können die gewonnenen Informationen auch zur Generierung von "Best-Practise-Lösungen" herangezogen werden. Die Methode des Reverse Business Engineering hilft also nicht nur dem einzelnen Unternehmen bei der Analyse und Dokumentation seines Softwaresystems, sondern kann auch dem Hersteller profunde Wettbewerbsvorteile verschaffen. Schließlich profitiert auch die Betriebswirtschaftslehre von den aggregierten Praxis-Erfahrungen.

1 Reverse Engineering für Softwarebibliotheken wie SAP R/3

Eine Softwarebibliothek definiert sich durch eine sehr große Funktionsbreite, variantenreich ausprägbare betriebswirtschaftliche Profile, durch eine betriebstyp- und branchenorientierte Systematik der Entwicklung, technische und methodisch mächtige Adaptionswerkzeuge und eine hohe Flexibilität und Revidierbarkeit im Rahmen der kontinuierlichen dynamischen Adaption (Hufgard 1994, S. 71). Bei der Nutzung eines solchen komplexen Systems kommt der permanenten Anpassung an veränderte Umweltbedingungen eine hohe Bedeutung zu. Wenn eine derartige dynamische Adaption nicht durchgeführt wird, wenn also keine fortwährende Diagnose und

Qualitätssicherung stattfindet, kommt es nur allzu häufig über erste auftretende Schwächen und Probleme zu Krisen!

Selbst für den Fall, daß es nicht zu einer Krise kommt, erscheint doch die Notwendigkeit einer kontinuierlichen Anpassung sowohl theoretisch wie praktisch allgemein anerkannt zu sein. So besteht auch in der Literatur Einigkeit darüber, daß der größte Teil des Aufwandes bei einem Softwaresystem nach dessen Einführung entsteht (Curth 1989, S. 7; Klösch 1995, S. 4; Müller 1997, S. 1 und Stahlknecht 1995, S. 160). Hauptsächlich hierfür ist der ständige Wandel der internen und externen Umweltbedingungen. Äußere Einflüsse bedingen innere Anpassungen und selbstinitiierte Verbesserungsmaßnahmen beeinflussen die Unternehmensorganisation. Folglich "klaffen" System und Realität immer auseinander: "Evolution is a natural phenomenon for such systems; it is not that someone forgot a requirement – the requirements inevitably change. There will always be old software." (Mueller 1997).

Die Vorgehensphilosophie und Methodik zur Lösung dieser Anforderung auf Basis einer Softwarebibliothek wie SAP R/3 wurden erstmals von Thome und Hufgard als "Continuous System Engineering" formuliert (Thome 1996). Ein wichtiger Baustein im Rahmen dieser Projektierungsphilosophie ist eine Analyse- und Diagnosekomponente der "realen, aktiven Geschäftsprozesse". Eine so geardete toolgestützte, permanente Istanalyse ist nicht nur zeitnaher, sondern liefert auch qualitativ bessere Ergebnisse und ist die Grundlage für ein "Reverse Business Engineering".

1.1 Klassisches Reverse Engineering

Das Konzept des Reverse Engineering steht nicht nur in sprachlicher Hinsicht dem Modellansatz des im Titel dieses Aufsatzes genannten Reverse Business Engineering nahe, sondern vor allem auch inhaltlich. Während allerdings das Reverse Engineering die Analyse auf den technischen Aspekt, nämlich die Analyse von Programmcodes, beschränkt, liegt beim Reverse Business Engineering der Schwerpunkt auf der betriebswirtschaftlichen Sicht und der Anwendungssituation. Aufgrund der sehr ähnlichen Zielsetzung erscheint eine kurze Auseinandersetzung mit den Methoden und Verfahren des Reverse Engineering sinnvoll.

Unter Reverse Engineering wird die Analyse eines bestehenden Systems, mit dem Zweck der Identifikation von Systemkomponenten und deren Beziehung untereinander, sowie der Erzeugung von Darstellungen des untersuchten Systems auf unterschiedlichen, höheren Abstraktionsniveaus verstanden (Klösch 1995, S. 8; Lehner 1991, S. 234 und Stahlknecht 1995, S. 162). Hierbei handelt es sich um einen reinen Untersuchungsvorgang, der nicht die Durchführung von funktionalen Änderungen beinhaltet (Gutzwiller 1991, S. 4 und Klösch 1995, S. 8).

Reverse Engineering-Maßnahmen werden notwendig, da zum einen die Anwendungssysteme oft eine schlechte Qualität vorweisen und unzureichend dokumentiert sind (Lehner 1991, S. 233), zum anderen die im Laufe der Zeit immer komplexer werdende Software auf der konzeptionellen Ebene leichter überschaubar und verständlicher ist, als auf der Implementierungsebene (Sneed 1992, S. 259). Auch moderne Software Engineering-Methoden machen das Reverse Engineering nicht überflüssig, da häufig Änderungen und Wartungsaktivitäten ohne Anwendung dieser Methoden stattfinden. Vielfach fehlt hier eine durchgängige Werkzeugunterstützung (Gutzwiller 1991, S. 23).

Reverse Engineering ist oft nur ein Teil eines Gesamtkonzeptes (Reengineering-/Wartungskonzept) (Gutzwiller 1991, S. 5 und Richter 1992, S. 128) und unterteilt sich selbst in die Bereiche Redokumentation und Redesign (Klösch 1995, S. 8-9; Lehner 1991, S. 235 und Müller 1997, S. 12). Im Rahmen der Redokumentation wird zunächst eine semantisch äquivalente Repräsentation des betrachteten Objektes erzeugt. Im nächsten Schritt wird dann, im Rahmen des Redesigns, ein Modell erstellt, das ein höheres Abstraktionsniveau besitzt. Letzterer Schritt erfordert zusätzliche Informationsquellen zum Programmcode, wie bestehende Dokumentationen, persönliche Erfahrungen des Wartungspersonals und allgemeines Wissen über das Problem und das Anwendungsgebiet (Stahlknecht 1995, S. 162).

Erfahrungen zeigen, daß ein erfolgreiches Reverse Engineering nur mit Werkzeugunterstützung denkbar ist, das Nachempfinden und Verstehen des ursprünglichen Designs oder strukturelle inhaltliche Verbesserungen jedoch Aufgabe des Entwicklers sind und bleiben werden (Wagner 1992, S. 15). Hieraus leitet sich die Anforderung ab, daß eine interaktive Kontrolle der Teilwerkzeuge und eine Bewertung der erzielten Ergebnisse jederzeit möglich sein muß. Ferner müssen die im Reverse Engineering-Prozeß erlangten Informationen werkzeugunterstützt weiterverarbeitet werden können (Stahlknecht 1995, S. 171).

1.2 Dokumentation der Einführung, Anwendung und dynamischen Anpassung

Die Dokumentation der Programme, Funktionen und Abläufe im Bereich von Softwarelösungen ist häufig ein Problem in der Praxis, da die Verantwortlichen die Aufgabe einer umfassenden Beschreibung ihrer Systeme vernachlässigen und es in der Folge zu teilweise erheblichen Verständnisproblemen bei den Anwendern/Entwicklern kommt (Arnold 1997, S. 64). Standardsoftware sind dagegen meist mit einer aussagekräftigen Dokumentation ausgestattet. Das Problem liegt hier also nicht in der Beschreibung der technischen Möglichkeiten der Software, sondern vielmehr mangelt es den Anwendern von Softwarebibliotheken an einer transparenten, einheitlichen Dokumentation der komplexen betriebswirtschaftlichen Konfiguration und vorgenommenen Anpassungen, die es - nicht nur einem Experten - ermöglicht, den Einführungsprozeß und die zugrundegelegten Anforderungen zu einem späteren Zeitpunkt nachzuvollziehen und eventuell ge-

zielt zu revidieren. Die Undurchsichtigkeit der einmal getroffenen und im Laufe des Adaptionprozesses im Rahmen des Continuous System Engineering immer wieder veränderten Einstellungen (Thome 1996, S. 78-81) führt auch dazu, daß die Diskrepanz zwischen den organisatorischen Anforderungen und existierenden Systemeinstellungen nicht erkannt wird und sich immer mehr vergrößert. Die Folge ist ein ineffizient und ineffektiv genutztes Standardanwendungssoftwarepaket. Ein System, das technisch betrachtet problemlos läuft, kann in Einzelbereichen, aufgrund der engen Verzahnung der einzelnen Prozesse, kontraproduktiv arbeiten (Computerwoche 1998, S. 19). Das Wissen über den Ist-Zustand des Systems und seiner Einstellungen ist ferner Voraussetzung für die weitere Adaption der Software an veränderte Anforderungen.

Ein weiterer, oftmals übersehener Aspekt einer nicht vorhandenen "aktiven" Dokumentation ist die Tatsache, daß selbst der Standardsoftwarehersteller, wie bspw. die SAP AG, keine oder nur sehr unzureichende Möglichkeiten hat, systematisch quantitative und qualitative Informationen über

- die Nutzung einzelner Standardprozesse,
- die Nutzung einzelner Standardfunktionen und
- realisierte Sonderlösungen bzw. Modifikationen

zu erlangen. Üblicherweise sind in Supportdatenbanken nur Fehler- und Problemsituationen bekannt. Ein solches "Sammelbecken" für Prozesse und Funktionen, Auswertungen und Schnittstellen, die im Laufe der Zeit von den Unternehmen, die SAP R/3 nutzen, realisiert worden sind, wäre sowohl für die SAP AG als auch für die Anwender von außerordentlich hohem Nutzen (Saller 1996, Folie 15), da dies wichtige Informationen zur Verbesserung und Weiterentwicklung der Software (die ja letztendlich dem Endanwender zu gute kommt) liefert.

2 Reverse Business Engineering-Ansatz

Der zunächst noch unbekannteste Begriff "Reverse Business Engineering" (RBE) wird beim Leser Assoziationen mit dem bereits etablierten Konzept des Business Process Re-Engineering hervorrufen. Und in der Tat geht es hier auch um die Gestaltung von Unternehmen und Prozessen. Aber im Gegensatz zu dem Modethema der neunziger Jahre strebt das "Reverse Business Engineering" nicht die radikale Änderung von Abläufen oder ganzen Organisationseinheiten (Hammer 1994) an, sondern will durch eine systematische, strukturiert und kontinuierlich betriebene Rückkopplung mit vielen aktiv genutzten Informationssystemen Verbesserungspotentiale in den Betrieben erschließen. Untersuchungsgegenstand für die "rückwärtsgerichtete Gestaltung" sind alle Prozesse im Unternehmen - sowohl die Kernprozesse, als auch die untergeordneten Funktionen dienen dabei als Basis für das "Reverse Business Engineering".

Aufgabe ist es nicht, wie es beim klassischen Reverse Engineering der Fall ist, mit Hilfe von Quellcode-Untersuchungen ein Abbild der Funktionalitäten der Software zu schaffen – diese sind in der Regel bei einer Softwarebibliothek bekannt. Aufgabe ist es vielmehr, basierend auf dem Gesamtmodell herauszufinden, welche Prozesse und Funktionen tatsächlich aufgrund der Customizing-Einstellungen, Stammdaten und Transaktionen wie oft und wie genau genutzt werden. Voraussetzung für das Erreichen eines solch‘ anspruchsvollen empirischen Ziels ist eine einheitliche und stabile Untersuchungsbasis wie SAP R/3, eine Modellstruktur wie das R/3-Referenzmodell und ein mächtiges RBE-Analysewerkzeug.

2.1 Forward versus Reverse Business Engineering

Im R/3-System liefert der sogenannte “Business Engineer” eine Modellstruktur für die Konfiguration der Softwarebibliothek auf die individuellen Anforderungen des Kunden. Diese Struktur führt den Anwender über mehrere (abstrakte) Modellebenen (Komponenten, Szenarien, Prozesse, Funktionen etc.) immer näher an das R/3-Customizing mit seinen Tabellen und Parametern heran (siehe Abbildung 1) (Schröder 1998).

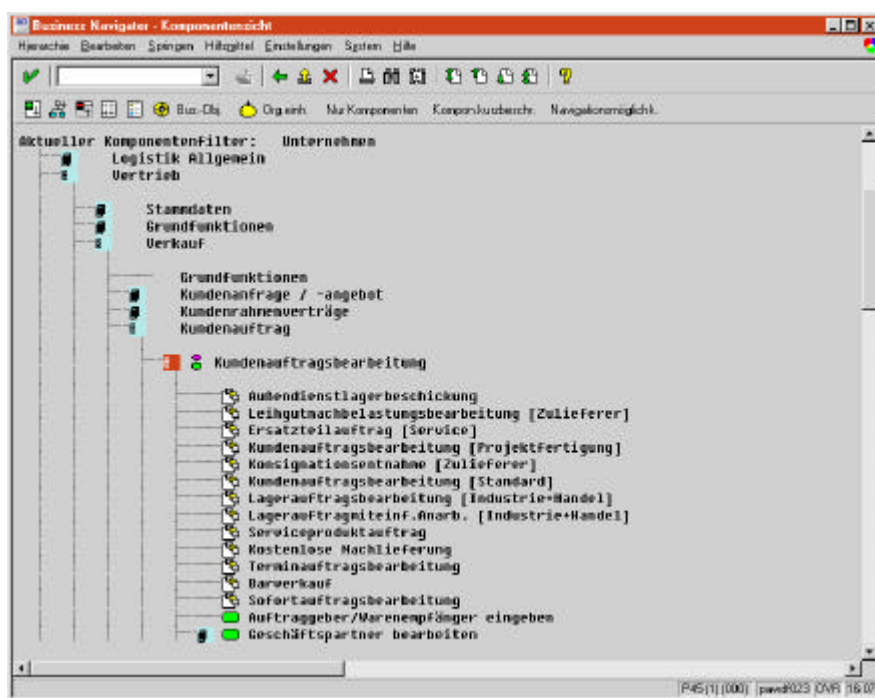


Abbildung 1: Modellhierarchie des Business Engineer

Beim “Reverse Business Engineering” wird die gleiche Struktur benutzt, die Zielsetzung ist aber genau entgegengerichtet: Es sollen ein oder mehrere bereits konfigurierte bzw. produktive R/3-Systeme analysiert und ihre Einstellungen im Modell abgebildet werden. Somit stellt das “Reverse Business Engineering” eine Umkehrung des normalen “Forward Business Engineering” (siehe Abbildung 2) dar.

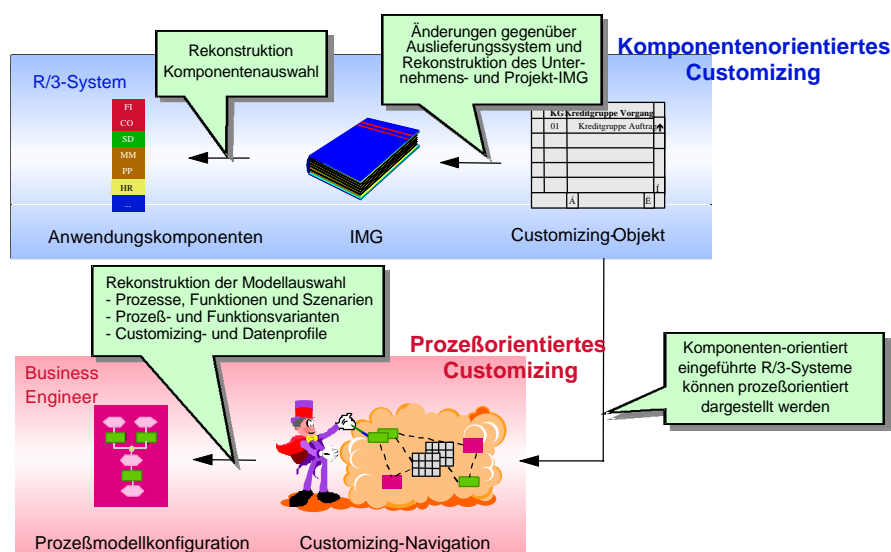


Abbildung 2: Reverse Business Engineering

Ziel ist es, den Einführungs- und Adaptionprozess eines R/3-Systems “von unten nach oben” nachzuvollziehen. Auf diesem Rückweg können produktive R/3-Systeme zeitnah und (halb-)automatisch nachdokumentiert und verifiziert werden. Die Analysemöglichkeiten über mehrere Systeme hinweg erlauben unterschiedlichste systematische Vergleiche bis hin zum Aufbau einer empirischen Datenbank, die Rückschlüsse über die richtige Nutzung der R/3-Softwarebibliothek im Einzelfall und insgesamt zuläßt. Damit kann einerseits die Anwendung von R/3 unterstützt und andererseits die Weiterentwicklung der Software und die Vorkonfiguration von Lösungsvarianten gezielt ausgerichtet und abgesichert werden.

2.2 Modellelementermittlung und -verdichtung

Die erste RBE-Aufgabe - die aktive Istanalyse - ist es, die Prozesse und Funktionen, die in der betriebswirtschaftlichen Individualbibliothek des Kunden aktiv sind, zu ermitteln.

Zunächst muß der Analyserahmen, also der zu betrachtende Teilbereich, eingrenzt und vorbereitet werden. Hierzu muß eine Analyse der

Parametereinstellung erfolgen: Mit einem Vergleich der tatsächlichen Einstellungen gegen das Auslieferungs- oder ein Referenzsystem können wertvolle Informationen gewonnen werden. Diese helfen zum einen diejenigen Komponenten und zugehörigen Customizing-Aktivitäten zu identifizieren, die nicht innerhalb eines Projektes selektiert wurden und somit zunächst für das RBE nicht relevant sind (Negativselektion). Zum anderen können Komponenten und zugehörige Customizing-Aktivitäten, die innerhalb eines Projektes selektiert wurden und somit für das RBE interessant sein können bzw. untersuchenswert sind, herausgefiltert werden (Positivselektion). Um den Analyserahmen weiter eingrenzen zu können, muß dann ein bestehendes System auf unterschiedliche Segmente untersucht werden (Organisationssegmentierung), z.B. welcher Buchungskreis oder welche Verkaufsorganisation von Interesse sind.

Im nächsten Schritt sollen für den Analyserahmen die gültigen Elemente der Prozeßmodellwelt retrograd ausgewählt werden. Hierzu müssen eigenständige "Analyse-Sets" zur Ermittlung der gültigen Funktionen und Prozesse herangezogen werden. Pro Modellelement wird ein sogenanntes "RBE-Set" definiert, das mittels einer heuristischen Vorgehensweise ermittelt, ob das betrachtete Modellelement im jeweiligen Anwendungsfall einen aktiven oder inaktiven Status aufweist. Inwieweit der Aussagegehalt noch weiter nach oben ins Modell getrieben werden kann, ist offen. Die Komponentenermittlung ist noch recht einfach, hingegen ist die Szenarioermittlung u. U. nicht eindeutig, da Abwahlentscheidungen bei Szenariodefinitionen und durch den Kunden jetzt gegeneinanderlaufen.



Abbildung 3: Ansätze zur Umsetzung des Reverse Business Engineering

Bei all den hier aufgeführten Analysen müssen auch Stammdateneinstellungen, Mengengerüste und Transaktionsdaten mit extrahiert werden, um die Nutzung einer Funktion auf Basis des Echtbetriebes auswerten zu können. Funktionen könnten zwar customized sein, aber trotzdem nicht verwendet werden!

Es muß auch festgestellt werden, daß ein RBE-Werkzeug nur spezifisch für eine Softwarebibliothek wie SAP R/3 entwickelt werden kann, da technische Eigenschaften und unterschiedliche betriebswirtschaftliche Eigenheiten eine Übertragung der RBE-Sets auf andere Standardanwendungssysteme, wie bspw. BAAN IV unmöglich machen.

2.3 Empirische Datenbank und Analysemöglichkeiten

Die zweite RBE-Aufgabe ist die Auswertung der gewonnenen Daten aus einer Vielzahl von Projekten. Es soll so eine "Business Engineering Library" als strukturierte Erfahrungsdatenbank aufgebaut werden. Hierbei gilt es, die Vielzahl von Zielsetzungen zu berücksichtigen, die bei aktuellen und zukünftigen Analysen interessant sein können.

Rein technisch gesehen, ist die BE-Library eine Sammlung von Datenextrakten, die wiederum aus einer Stichprobe von kritischen Customizingparametern, relevanten Stammdaten- bzw. Transaktionsdatenkennzahlen und Rahmendaten zum einzelnen Unternehmen bestehen. Für einen Betrieb wird es auch mehrere Extrakte geben, die sich zeitlich und nach Untersuchungssegmenten unterscheiden. Grundsätzlich müssen hier, angelehnt an die empirische Marktforschung, statistische Analysen wie Faktor- und Clusteranalyse möglich sein. Die Betriebswirtschaftslehre und Wirtschaftsinformatik stellen zusätzlich Strukturierungsraster und weitere analytischen Konzepte bereit, um die Datenbasis aufzubereiten.

2.4 Mustererkennung und fallbasiertes Schließen

Für die Problemerkennung und das Auffinden von "neuen Lösungen", die in der praktischen Anwendung erst erarbeitet werden, muß der Weg zum Diagnose- und Unterstützungssystem weitergegangen werden.

Ist bei einer aktiven Funktion eine unbekannte Parametereinstellung im Einsatz, muß sie mit extrahiert werden, um sie untersuchen und bewerten zu können. Meist ist es ratsam, ebenfalls das Umfeld mit zu analysieren. Bestimmte bekannte fehler- oder problembehaftete Konstellationen werden aufgezeichnet und bewertet. Im Diagnosefall werden die Empfehlungen und Schlußfolgerungen dem Kunden zur Verfügung gestellt. Darüber hinaus existieren noch weitere Techniken, die beim Ausbau der Diagnosemöglichkeiten herangezogen werden können.

3 RBE-Toolentwicklung für SAP R/3

Für die beiden letzten RBE-Aufgaben entwickelt das Forschungsteam der IBIS Prof. Thome GmbH und des Lehrstuhls von Prof. Thome zusammen mit dem Business Engineering Center der SAP AG z. Zt. die Konzepte und Werkzeuge weiter. Mit Erfahrungsberichten ist 1999 zu rechnen. Für die erste RBE-Aufgabe, der Modellermittlung auf Basis eines produktiven Systems, liegen bereits praktische Erfahrungen vor.

3.1 Reverse Modelling Assistant (RMA)

Der Reverse Modelling Assistant ist das Tool, das den Anwender bei der Analyse seines produktiven R/3-Systems unterstützt. Dabei stellt es die relevanten Daten aus dem System bereit, ermittelt mit Hilfe der Reverse Business Engineering-Sets (RBE-Sets) retrograd den Status der Modellelemente, präsentiert die Ergebnisse und bereitet sie für andere Implementierungs- und Dokumentationstools auf.

Die RBE-Sets sind das Kernstück des Werkzeugs, da sie das gesamte Wissen, das zur Ableitung von Modellen aus produktiven R/3-Systemen notwendig ist, in sich vereinen. Angesichts dieser Fülle an Experten-Wissen, müssen die RBE-Sets toolgestützt definiert, abgelegt und verwaltet werden können. Das abgelegte Wissen besteht aus:

- Prüfschritten,
- Datenabfragen und Regeln für die Beantwortung der Prüfschritte sowie
- Prüfungen.

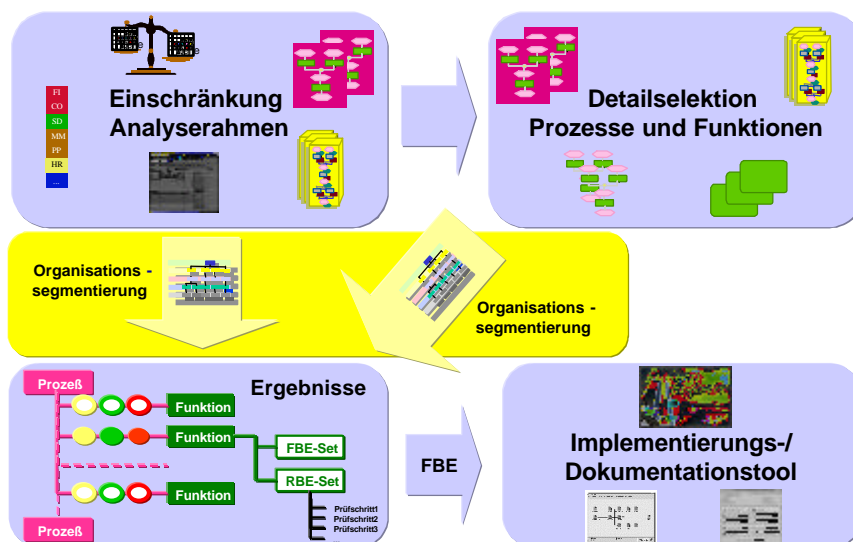


Abbildung 4: Übersicht der Hauptfunktionsblöcke des RMA

Das entwickelte Werkzeug wurde bereits bei einem Anwender (Hersteller von Baustoffen) eingesetzt und getestet. Hierbei wurden vor allem die Logistik-Module MM, PP und SD untersucht. Weitere Pilotprojekte folgen in den nächsten Monaten. Die allgemeine Verfügbarkeit des Tools für R/3-Anwendungsunternehmen ist für Januar 1999 geplant.

Im folgenden werden die vier Hauptfunktionsblöcke (siehe Abbildung 4) des "Reverse Modelling Assistant" vorgestellt. Hierbei wird auch auf praktische Erfahrungen, die im Rahmen des ersten Pilotprojektes gemacht wurden, eingegangen.

Nachdem der Kunde mit seinen allgemeinen Daten (z.B. Anschrift, Kundennummer) und den Informationen zu seinem R/3-System (z.B. Mandant, Release), angelegt wurde, kann der Analyserahmen für die weitere Untersuchung festgelegt werden. Als Basis für den Analyserahmen stehen das R/3-Referenzmodell, Branchenmodelle oder bereits vorhandene Kundenmodelle zur Verfügung, wobei jeweils zwischen einer komponentenorientierten und einer prozeßorientierten Darstellung gewählt werden kann (siehe Abbildung 5).

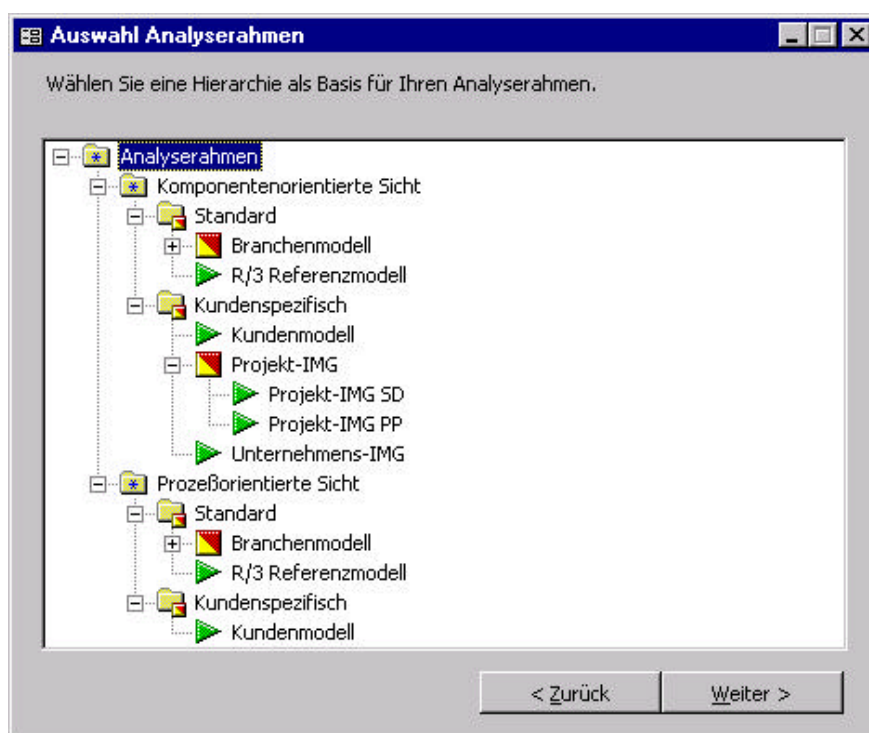


Abbildung 5: Selektieren einer Hierarchie als Basis für den Analyserahmen

Das Ergebnis ist eine Auswahl von Prozessen und Funktionen, die auf Basis der zuvor aus dem Kundensystem geladenen und in der Datenbank des RMA abgelegten Extrakte analysiert werden. Als Detektor für aktive, aber evtl. nicht selektierte R/3-Bereiche kann der "SAP R/3 Mandanten-Abgleich" genutzt werden, indem er Veränderungen von Customizing-Tabellen gegenüber dem Auslieferungsmandanten meldet. Zur Erstellung der Extrakte wird unter anderem auf verdichtete Auswertungen der im R/3-System vorhandenen Analysetools, wie z.B. den Transaktionsmonitor zurückgegriffen. Auf diese Weise kann die gewonnene Datenmenge möglichst gering und transportierbar gehalten werden. Im konkreten Anwendungsfall betrug die zu extrahierende Datenmenge z.B. unter 1 MB.

Nachdem die Basis für die Untersuchung festgelegt wurde, kann optional eine weitere Einschränkung des Analyserahmens (Detailselektion) vorgenommen werden (siehe Abbildung 6). Dies geschieht durch einfaches Drag & Drop der gewünschten Komponenten, Prozesse oder Funktionen von einem Fenster in das andere.

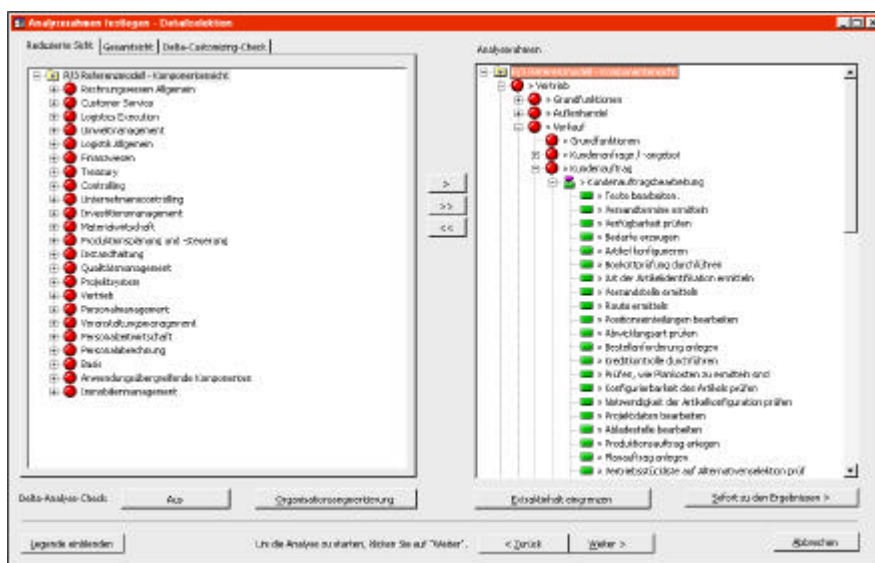


Abbildung 6: Reverse Modelling Assistant – Detailselektion

Auf Wunsch werden aktive Organisationsinstanzen, in Abhängigkeit vom selektierten Prüfumfang, aus den R/3-Tabellen gelesen und zur Selektion angeboten. Diese Funktionalität hat sich in der praktischen Anwendung als sehr hilfreich erwiesen, da die Aussagekraft der Analyseergebnisse über eine Organisationssegmentierung (Thome 1996) erheblich gesteigert werden kann: je enger das zu untersuchende Segment (z.B. Werk, Verkaufsorganisation) eingegrenzt werden kann, desto spezifischer können aktive Prozesse und Funktionen erkannt werden.

Der Pilotkunde verfügt bspw. über zehn Werke, die unterschiedlichste Funktionalitäten von R/3 nutzen.

Die Prüfung der selektierten Elemente kann nun “schnell” oder “vollständig” durchgeführt werden. Bei einer schnellen Prüfung bricht das System die Analyse ab, sobald es pro Untersuchungselement ein Ergebnis gefunden hat, während bei einer vollständigen Analyse alle im RBE-Set definierten Prüfschritte durchlaufen und am Ende pro Untersuchungselement das “beste” Ergebnis angezeigt wird.

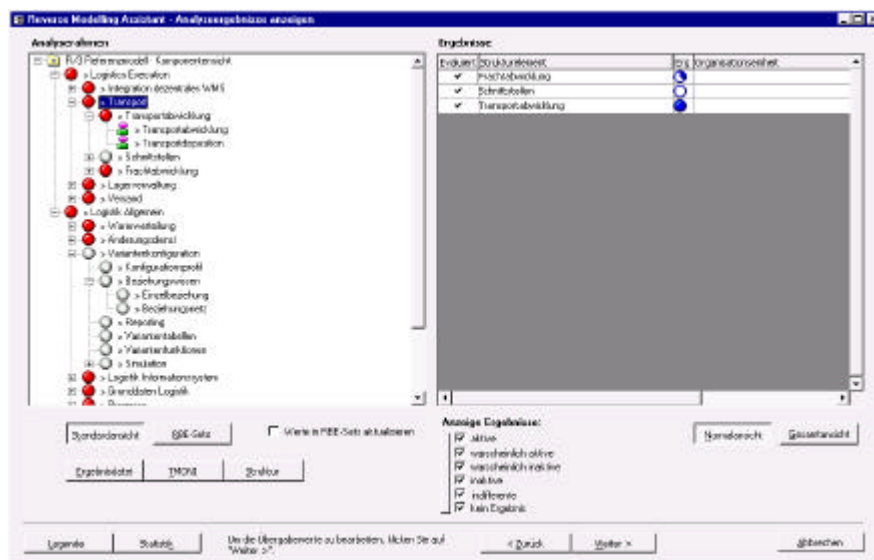


Abbildung 7: Reverse Modelling Assistant – Analyseergebnisse

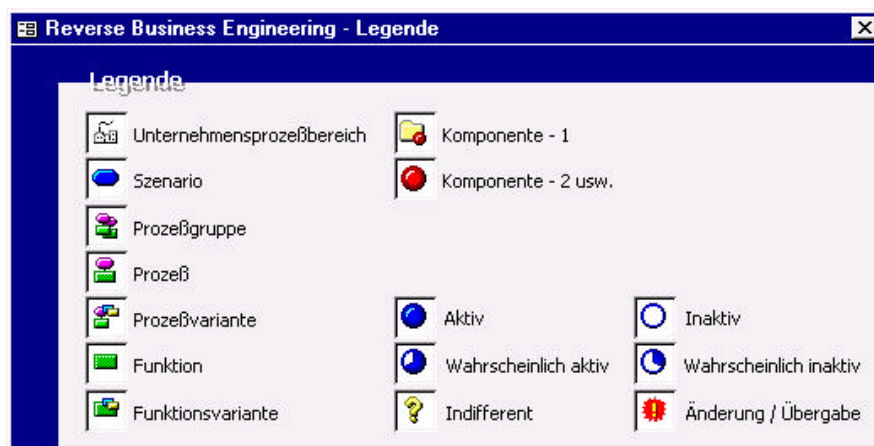


Abbildung 8: Reverse Modelling Assistant – Legende

Als Ergebnis der Analyse wird im RMA-Tool der Status der Komponenten, Prozesse und Funktionen (z.B. untersucht, aktiv, inaktiv), entsprechend graphisch aufbereitet, dargestellt (siehe Abbildung 7). Als Erklärungskomponente können über den Button "RBE-Sets" Informationen über die zuvor definierten, durchgeführten Prüfschritte und Detailergebnisse abgerufen werden.

Erfahrungsgemäß sind also an dieser Stelle noch manuelle Nacharbeiten notwendig, bevor die endgültigen Ergebnisse an ein Implementierungstool zur Weiterbearbeitung gegeben werden können (siehe Abbildung 9). Dies ist z.B. auch deshalb erforderlich, weil in vielen R/3-Projekten bestimmte Funktionalitäten für andere Zwecke "mißbraucht" werden. So wurde z.B. beim Pilotkunden der Einsatz von Prozeßaufträgen, jedoch keine Prozeßfertigung identifiziert. Die manuelle Nachbearbeitung ergab, daß dies zur Realisierung spezifischer Anforderungen sinnvoll war. Dieses Beispiel zeigt, daß der RMA auch zur Identifikation "atypischer" Einstellungskombinationen eingesetzt werden kann.

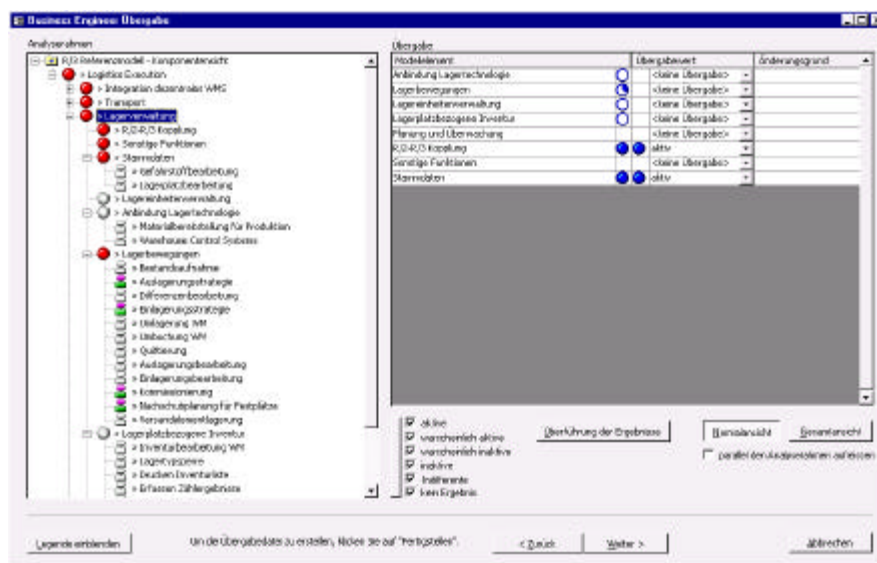


Abbildung 9: Übergabewerte festlegen

Ebenso hilfreich sind Auswertungen über Mengengerüste und Zeitreihenanalysen von Aktivitäten pro Modellelement. Gerade in den ersten Monaten der produktiven Nutzung eines R/3-Systems helfen diese Werte die Anwendungssituation richtig zu beurteilen.

Nach erfolgreicher Überarbeitung der Status, können die Ergebnisse an den ASAP-Business Engineer oder an ein anderes Prozeßdarstellungs- und -analysetool übergeben werden. Sollen die Ergebnisse in ASAP weiterverarbeitet

macht". Mit Hilfe des Reverse Modelling Assistant können solche bereits realisierten Projekte transparent reproduziert und die Dokumentation auf Modellebene automatisch erstellt werden. Diese automatische Erstellung schafft bei den Anwendern Akzeptanz für die Modelle und erleichtert den Wechsel auf eine prozeßorientierte Vorgehensweise im Folgeprojekt.

Der Reverse Modelling Assistant kann jedoch auch bereits in der Implementierungsphase eingesetzt werden. Zum einen können teilweise realisierte Projekte in die Modellwelt überführt und prozeßorientiert weitergeführt werden, zum anderen liefert die Analyse der getroffenen Einstellungen jederzeit den *Realisierungsstand* im Projekt.

Durch den Abgleich der Modellauswahl gegen die tatsächlichen Einstellungen, kann jederzeit *verifiziert* werden, ob der Ist-Zustand des Systems noch dem über *Modellentscheidungen* definierten Soll-Zustand entspricht. Werden hier gewollte Abweichungen festgestellt, kann das Modell automatisch nachgezogen und so aktuell gehalten werden. War die Diskrepanz zwischen SOLL und IST nicht gewünscht, können die Systemeinstellungen entsprechend angepaßt werden.

Indem der Reverse Modelling Assistant eine einheitliche Dokumentationsbasis auf Modellebene schafft, ist ein *Benchmarking* und ein *Vergleich* von verschiedenen Projekten möglich. So können z.B. Vorlageprojekte (Templates) mit den tatsächlichen Einstellungen abgeglichen werden, um anschließend, auf Basis von Bewertungsschemata, bereits frühzeitig im Projekt Schlußfolgerungen zu ziehen (z.B. über Kostenkalkulation oder Einführungszeit).

Ein weiteres wichtiges Einsatzgebiet des RMA ist die Analyse des *Modelldeckungsgrades*. Da die Analyse auf Basis des R/3-Referenzmodells durchgeführt wird, können z.B. über bisher unbekannte Funktions- oder Prozeßvarianten von aktiv identifizierten Funktionen oder Prozessen Sonderlösungen und Modifikationen herausgefiltert werden. In einem nächsten Schritt kann dann ermittelt werden, ob die Modifikation überhaupt nötig war und die Funktionalität nicht auch über den Standard hätte abgebildet werden können. Ist dies nicht der Fall, und wird eine Sonderlösung in mehreren Projekten identifiziert, kann sie evtl. in den Standard aufgenommen werden.

Die Einsatzziele fordern ein dezentral einsetzbares PC-basiertes Werkzeug, das über Schnittstellen mit vielen R/3-Systemen und unterschiedlichen Einführungs- und Modellierungswerkzeugen kommunizieren muß.

Literaturverzeichnis

- Arnold, B. (1997): Darf's ein bißchen mehr sein? In: Computerwoche 43 (1997), S. 63-66.
- o. V. (1998): Neue Tools sollen R/3-Anwendern Investitionsklarheit verschaffen. In: Computerwoche 14 (1998), S. 19.

- Curth, M. A./Giebel, M. L. (1989): Management der Software-Wartung. Stuttgart 1989.
- Gutzwiller, T. (1991): State of the art des Reverse Engineering. St. Gallen 1991.
- Hammer, M./Champy, J. (1994): Business Reengineering. Die Radikalkur für das Unternehmen. Frankfurt/New York, 1994.
- Hufgard, A. (1994): Betriebswirtschaftliche Softwarebibliotheken und Adaption. München 1994.
- Kargl, H. (1996): Controlling im DV-Bereich. 3. Aufl., München u. a. 1996.
- Klösch, R./Gall, H. (1995): Objektorientiertes Reverse Engineering - Von klassischer zu objektorientierter Software. Berlin u.a. 1995.
- Lehner, F. (1991): Softwarewartung: Management, Organisation und methodische Unterstützung. München u. a. 1991.
- Müller, B. (1997): Reengineering: Eine Einführung. Stuttgart 1997.
- Mueller, H. A. (1997): Reverse Engineering - Strategies for Software Migration. Vortrag im Rahmen der Sixth European Software Engineering Conference. Zürich 1997.
- Richter, L. (1992): Wiederbenutzbarkeit und Restrukturierung oder Reuse, Reengineering und Reverse Engineering. 34Aufl., In: Wirtschaftsinformatik. Wiesbaden 1992.
- Saller, W. (1996): Einführung und Erfahrung mit SAP R/3. Vortrag im Rahmen der SAPHIRE 1996. Wien 1996.
- Schröder, G. (1998): Strategische Vision der R/3-Einführung. Vortrag im Rahmen der SAP-IBIS-Tagung. Würzburg 1998.
- Sneed, H. M. (1992): Softwarewartung und -wiederverwendung Bd II: Software-sanierung. Köln 1992.
- Stahlknecht, P./Drasdo, A. (1995): Methoden und Werkzeuge der Programm-sanierung. In: Wirtschaftsinformatik. Wiesbaden 1995.
- Thome, R./Hufgard, A. (1996): Continuous System Engineering: Entdeckung der Standardsoftware als Organisator. Würzburg 1996.
- Wagner, B. et. al. (1992): Reverse Engineering: Sanierung, Dokumentation und Strukturierung vorhandener Software. Ehningen bei Böblingen 1992.