

---

## 5 Realisierung

Das folgende Kapitel ist in zwei Abschnitte untergliedert. Im ersten Teil sollen wichtige Aspekte des integrierten Workflow-Managements an Universitäten anhand exemplarischer Realisierungen verdeutlicht werden. Die Darstellung der einzelnen Realisierungen zu Workflows, Workflow-Mustern oder die Integration von Workflows und Internet soll ausgehend von Portalen<sup>411</sup> stattfinden. Nach der Darstellung der Aspekte des integrierten Workflow-Management aus Sicht der Anwendung und der Modellierung, sollen abschließend die Implementierungen zu den beschriebenen Workflows bzw. Integrationsaspekten vorgestellt werden. Die Implementierungen fanden im wesentlichen in den Sprachen wie ABAP<sup>412</sup>, C und HTML<sup>413</sup> statt. Im zweiten Abschnitt wird die Realisierung eines Ausschnitts zum „Konzept des Flexiblen und Regelbasierten Workflow-Managements“<sup>414</sup> vorgestellt. Die Prozeßinitiierung als ausgewählte Workflow-Operation bildet den Ausgangspunkt der durchgehenden Realisierung. Der Aufbau der Prozeßbeziehungen wird dabei - aus Sicht der Modellierer - anhand der Strukturen zur Modellierung sowie der erforderlichen Aufgaben aufgezeigt. Mithilfe des Analysewerkzeugs „VisuFlex“<sup>415</sup> werden die modellierten Prozeßbeziehungen visualisiert und die flexible Gestaltung der Prozeßbeziehungen aufgezeigt. Auch hier werden Aspekte der konkreten Implementierung vorgestellt.

Entsprechend der obigen Zweiteilung ist auch die Darstellung der Realisierung in zwei Abschnitte unterteilt. Im ersten ist die Implementierung zum integrierten Workflow-Management an Universitäten und im zweiten dann die Realisierung zum „Konzept des Flexiblen und Regelbasierten Workflow-Managements“ dargestellt.

---

<sup>411</sup> Im Rahmen von Portalen werden wichtige Dienste für Studenten, grundsätzlich auch für Mitarbeiter, zusammengefaßt. So wird ein strukturierter Zugang zu den einzelnen Diensten gegeben, d.h. das Portal dient als deren Klammer.

<sup>412</sup> vgl. Kretschmer, R./Weiss, W. (1996); Mende, U. (1998); Matzke, B. (1996)

<sup>413</sup> vgl. Abschnitt 2.3

<sup>414</sup> vgl. Abschnitt 4.5

<sup>415</sup> VisuFlex wurde im Rahmen der vorliegenden Arbeit entwickelt und vom Autor konzipiert.

---

Die beschriebene Realisierung wurde einerseits als Bestandteil einer speziell auf Universitäten zugeschnittenen Branchenlösung „Campus Management“<sup>416</sup> der SAP AG auf Basis des SAP R/3 – Systems realisiert. Andererseits konnten im Rahmen eines Forschungs- und Entwicklungsprojektes am Lehrstuhl für Wirtschaftsinformatik der Universität Passau Implementierungen im Hinblick auf die Unterstützung der Organisation der Lehre sowie zu Forschungszwecken auf dem SAP R/3 – System des Lehrstuhls umgesetzt werden.

---

<sup>416</sup> vgl. Glaser, I./Schabel, F. (1999), S.10f

## 5.1 Implementierung zum integrierten Workflow- Management an Universitäten

Im Rahmen des Projekts „Campus Management“ wurden die Implementierungen Bestandteil einer eigens für Universitäten entwickelten Branchenlösung. Dabei zeigten die Prozesse, daß diese im Kern landes- bzw. kontinentübergreifend waren, dennoch aber spezifische Unterschiede aufwiesen.

Als die vorliegende Arbeit angefertigt wurde, lag der Schwerpunkt der Entwicklungen auf den Prozessen der Studentenadministration, der Planung u.a. von Programmen und Lehrveranstaltungen, der Durchführung von Lehr- und Prüfungsveranstaltungen und der Dienste aus dem Finanzbereich, wie die Stipendienverwaltung und die Abrechnung.

Weiterführende Aspekte der Lehre bzw. Dienstleistungen für Studenten aus Sicht der Lehrstühle werden an Implementierungen gezeigt, die am Lehrstuhl für Wirtschaftsinformatik in Passau durchgeführt wurden.

Die jeweiligen Leistungen, und damit auch der Zugang zu den Prozessen, sind systematisch sowohl im „Campus Management“ als auch am Lehrstuhl für Wirtschaftsinformatik in Portalen zusammengefaßt. Daher werden nachfolgend die beiden Portale vorgestellt. Anschließend werden zwei Workflows skizziert, der Workflow „Zulassung“, der als Workflow-Muster gestaltet wurde, und der Workflow „Problemmeldung“, der als vollständiger Workflow am Lehrstuhl für Wirtschaftsinformatik eingesetzt wird. Die Darstellung der Prozesse wird dann um die Integrationsaspekte ergänzt. Anhand des Workflows „Evaluation“ soll die durchgängige Integration des Workflows im WWW aufgezeigt werden.

Abschließend werden die Modellierungen und Programme, die den Implementierungen zugrundeliegen, in den verschiedenen Sprachen exemplarisch verdeutlicht. Die auf der Architektur der SAP R/3-Systeme aufbauenden Implementierungen betreffen dabei die Programmiersprache ABAP, die Seitenbeschreibungssprache HTML, HTML Business<sup>417</sup> und die auf ABAP aufgesetzte Makrosprache des

---

<sup>417</sup> vgl. die SAP eigene Spracherweiterung zu HTML in Abschnitt 2.3

---

Business Object Repository (BOR). Das BOR bildet dabei die Ebene zwischen den Workflows und den eigentlichen Programmen.

### 5.1.1 Portale als Klammer von Prozessen und Diensten

Um mit den notwendigen Inhalten vertraut zu werden, ist die Beschreibung des Begriffs „Portal“<sup>418</sup> erforderlich. Ferner sind die Hintergründe für den Einsatz aufzuzeigen.

Portale stellen eine systematische Weiterentwicklung der Organisation von Diensten dar, die meist auf verschiedene Web-Seiten verteilt werden. Im folgenden wird daher unter ihnen eine in einem Webbrowser lauffähige WebServer-Applikation<sup>419</sup> verstanden werden, die eine Vielzahl von Informationen katalogisiert aufbereitet und dem Anwender den Einstieg über eine Hypertext-Startseite ermöglicht. Je nach Zielgruppe der Portale werden Consumer-Portale<sup>420</sup>, also solche, die sich an Endkunden wenden, und Corporate Portale<sup>421</sup>, die also Dienste für eigene Mitarbeiter oder Geschäftspartner zusammenfassen, unterschieden. Mitarbeiterportale werden auch als Self-Service-Portale für Mitarbeiter bezeichnet.

Durch die Portale versprechen sich die Dienstanbieter verschiedene Vorteile. So steigt die Zufriedenheit der Teilnehmer durch das systematische Aufteilen der Informationen. Gerade bei Consumer Portalen, wo die typischen Internetkunden leicht verständliche und einfach zu bedienende Applikationen erwarten, ist die Kundenzufriedenheit ein entscheidender Akzeptanzfaktor.

Auch die Auslagerung verschiedener Arbeiten als Self-Services der Mitarbeiter in den Corporate Portalen, wie z.B. die Reisekostenabrechnung, erfordert hier ein schnelles Auffinden der relevanten Informationen und eine einfache Bedienbarkeit.

---

<sup>418</sup> vgl. zur Begriffsbildung von Portalen: Patricia Seybold Group (1999), S.3; Koulopoulos, Th.M. (2000), S.32-35; Plumtree Software, Inc. (1999), S.3; Röttgers, J. (2000), S.A-1; Vering, M. (2000), S.36-39; ZD Webopedia (2000), o.S.

<sup>419</sup> Applikationen, die auf den Web-Servern ablaufen. Dabei stellen Web-Server u.a. WWW-Seiten bereit (vgl. hierzu Abschnitt 2.3).

<sup>420</sup> vgl. Consumer Portale wie AOL.com oder yahoo.com; Röttgers, J. (2000), S.20-39

<sup>421</sup> vgl. Corporate Portale wie mySAP.com (Workplace) in SAP (2000), S.20f; Koulopoulos, Th.M. (2000), S.32-35

Portale können auch als elektronische Marktplätze<sup>422</sup> gestaltet werden, wo verschiedene Anbieter Dienste bereitstellen. Für Unternehmen können so Transaktionskosten vermindert werden, z.B. durch schnellere Anbahnung oder Abwicklung der Geschäfte. Der Portalanbieter<sup>423</sup> erhöht durch eine Vielzahl von mitwirkenden Anbietern die Angebotsvielfalt und damit die Attraktivität des eigenen Marktplatzes.

Den Universitäten wird durch die Portale die Möglichkeit eröffnet, Informationen und Dienste, die den Studenten zur Verfügung gestellt werden sollen, systematisch anzubieten zu können<sup>424</sup>. Es sollen im folgenden zwei exemplarische Consumer Portale vorgestellt werden. Zum einen ein Portal, das ausgewählte Dienste der Studentenadministration anbietet, und zum anderen ein Portal, das Dienste vorlesungsbegleitend aus Sicht des Lehrstuhls zur Verfügung stellt<sup>425</sup>.

Das nachfolgende Portal „Student Service Bench“ (SSB) faßt verschiedene Dienste der Studentenadministration sowie das Lehrangebots zusammen. So können Studenten ihren Terminkalender einsehen oder Anträge auf Rückmeldung stellen. Dieses Portal ist Teil der Software „Campus Management“ und kann von Universitäten zu individuellen Portalen erweitert werden.

---

<sup>422</sup> vgl. Marktplätze in: mySAP.com (Marketplace) in SAP (2000); Graf, P. (2000), S.16f

<sup>423</sup> vgl. SAP als Bereitsteller von Portalen z.B. für die Chemieindustrie in Graf, P. (2000), S.16f

<sup>424</sup> vgl. Abschnitt 2.1

<sup>425</sup> vgl. URL: <http://www.winf.uni-passau.de/self/index>

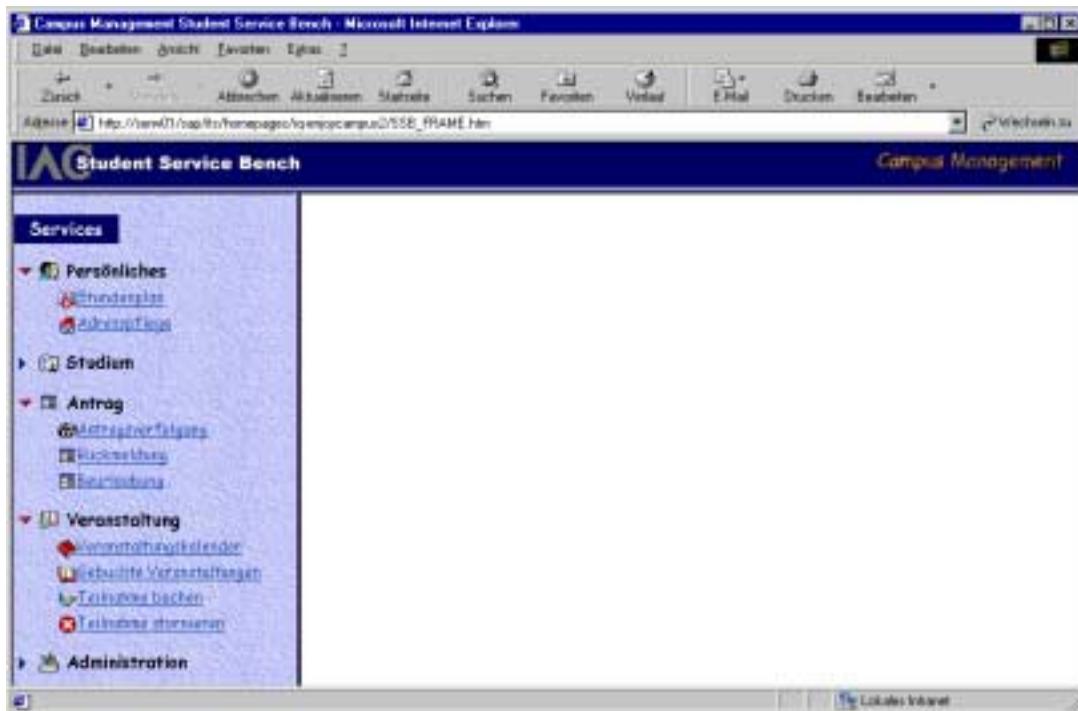


Abbildung 5-1: Portal „Student Service Bench“

Durch das Angebot können somit Studenten jederzeit wichtige Informationen abrufen. Das elektronische Stellen von Anträgen, z.B. auf Zulassung oder Beurlaubung, erleichtert den Studenten eine termingerechte, aber auch korrekte Antragstellung. Der Status wichtiger Prozesse kann über den Service der „Antragsverfolgung“ erfragt werden. Ist ein Zulassungsantrag auf Studienwunsch genehmigt, erhält der Student schon vor dem postalischen Eintreffen die wichtige Information. Durch diese Angebote erhöht die Universität ihre Angebotspalette. Auch die Darstellung der universitären Veranstaltungen sowie der Buchung durch Studenten, die in den Kalender des Studenten eingetragen werden, unterstützen bei einer effizienten Studienorganisation.

Aus dem Portal „Student Service Bench“ soll im folgenden Teilabschnitt der Workflow „Zulassung“, dann die Integration des Workflows mit dem WWW am Beispiel der Antragstellung aus dem WWW und deren Überwachung dargestellt werden.

Das Portal des Lehrstuhls für Wirtschaftsinformatik umfaßt exemplarisch vorlesungsbegleitende Dienste (vgl. Abb. 5-2). Spezielle Veranstaltungen und Seminare des Lehrstuhls können online eingesehen und kostenfrei gebucht werden. Die

Buchungen können dann rechtzeitig auf Engpässe z.B. bei den Räumen hinweisen<sup>426</sup>. Weiterhin können Probleme der Studenten, die sich beim Lösen von Aufgaben oder bei Rechnerübungen ergeben, online geschildert und prozeßgesteuert bearbeitet werden.



Abbildung 5-2: Portal des Lehrstuhls für Wirtschaftsinformatik

Der Vorlesungsbetrieb kann von den Diensten im WWW begleitet werden. Überdies können Anträge auf Rechnerkennungen online gestellt werden, Studenten können sich als Interessenten registrieren lassen, um so Impulse für Veranstaltungen zu geben bzw. auch um über neue Veranstaltungen schneller informiert zu werden. Diese flankierenden Dienste tragen zu der positiven Einschätzung der Studenten bei. Der Dienst der Problemmeldung „Fragen-SAP R/3-System“ sowie die Funktion des Workfloweingangskorbs als wichtiger Bestandteil der Workflow-Integration werden in den folgenden Abschnitten genauer untersucht.

<sup>426</sup> Die Buchungen haben auch einen didaktischen Hintergrund. So fühlen sich die Studenten einerseits der gebuchten Veranstaltung im Hinblick auf Vorbereitung und Präsenz stärker verpflichtet, außerdem lernen sie die Möglichkeiten der Nutzung von Self-Services auf der Grundlage von Standardsoftware kennen.

### 5.1.2 Konkrete Workflows

Als Beispiele für ein Workflow-Muster bzw. einen vollständigen Workflow soll das Workflow-Muster „Zulassung“ sowie der Workflow „Problemmeldung“ vorgestellt werden. Das Workflow-Muster „Zulassung“, das bereits in Abschnitt 3 entwickelt wurde, nimmt als zentraler Geschäftsprozeß von Universitäten einen breiten Darstellungsraum ein.

Der zugrundeliegende Workflow „Studienwunsch“ findet im Rahmen der Zulassung statt. Die Zulassungsprozesse unterscheiden sich in den verschiedenen Ländern bzw. Universitäten deutlich voneinander. Dennoch weisen alle Prozesse dieser Art einen ähnlichen Prozeßkern auf<sup>427</sup>. Gemeinsame Prozeßkerne sollen durch zugehörige Workflow-Muster beschrieben sowie die Benutzerübersicht bei Ablauf des Workflow-Musters exemplarisch dargestellt werden. Workflow-Muster müssen individuell von einsetzenden Unternehmen spezifisch angepaßt, d.h. spezialisiert, werden. Der Prozeß „Studienwunsch“ geht von der Variante der Einzelbearbeitung aus. Die ebenfalls häufige Variante der Massенbearbeitung, wo auf Wartelisten bzw. Ranglisten zahlreiche Bewerber verwaltet und schließlich die Plätze auf einem Schlag vergeben werden, wird nicht explizit betrachtet. Wenn Studienplätze außerhalb der Einzelbearbeitung vergeben werden, entfallen Schritte des Prozesses. Daher können Einzel- und Massенbearbeitung parallel betrieben werden. In der Praxis ist die Massенverarbeitung als eigene Anwendung realisiert<sup>428</sup>. Ob eine Einzel- oder die Massенbearbeitung vorliegt, hängt vom jeweiligen Studiengang ab. Bei besonders guten Bewerbern kann die Universität auch im Rahmen der Massенbearbeitung die workflow-unterstützte Einzelfallbearbeitung durchführen, also beide Verfahren mischen.

Der Workflow zu diesem verallgemeinerten Prozeßkern „Studienwunsch“ wurde auf Basis des WFMS des SAP R/3-Systems durchgeführt (vgl. Abb. 5-3). Dadurch ergeben sich im Vergleich zum allgemeinen Teil geringe Unterschiede in der Workflow-Modellierung, so z.B. bei der Terminüberwachung. Das Workflow-

---

<sup>427</sup> vgl. Abschnitt 3



Muster ist dabei aus den entsprechenden Prozessschritten aufgebaut: „Antrag prüfen“ zur Prüfung der formalen Voraussetzung sowie zur Prüfung der akademischen Voraussetzungen sowie „Versenden des Schriftverkehrs“, die der Sachbearbeiter durchführen soll. Ein sogenannter Hintergrundschritt „Student zuweisen“ ermittelt zu Beginn alle bewerberrelevanten Daten. Auch der Schriftverkehr kann automatisiert werden. Im Muster wird bei unvollständigen Unterlagen über eine Schleifenbedingung dem Studenten die Möglichkeit der Nachreichung gegeben. Über die automatische Terminüberwachung des WFMS wird die Wahrung von Fristen und Terminen umgesetzt.

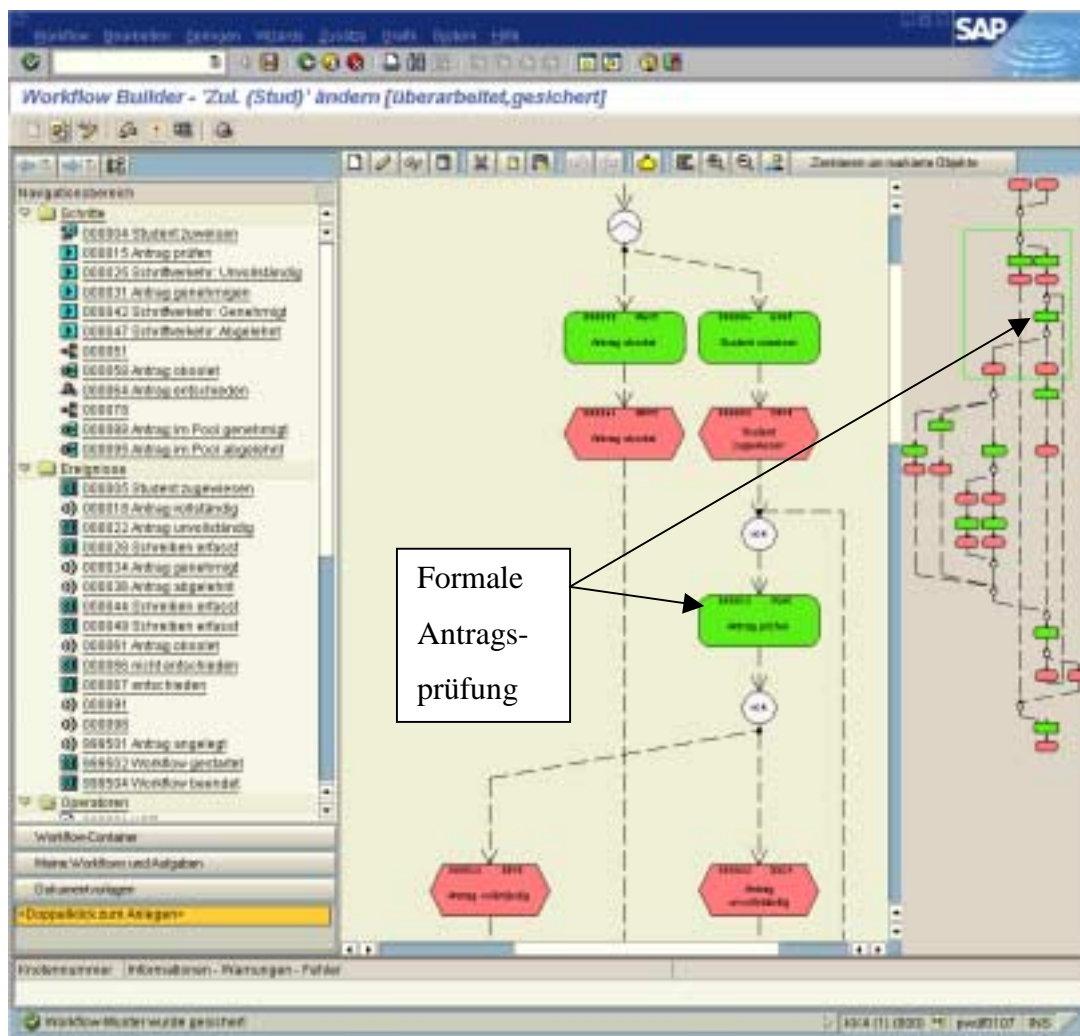
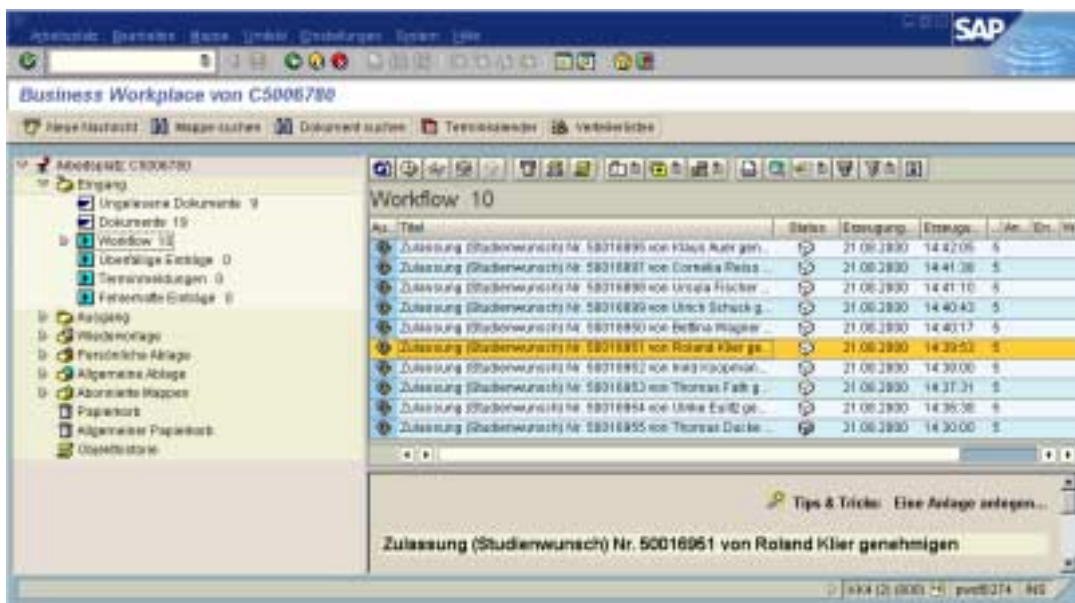


Abbildung 5-3: Workflow-Muster „Studienwunsch“

<sup>428</sup> Die Unterstützung durch einen Workflow könnte die Termingerechtigkeit der Vergabe sicherstellen.

Die folgende Modellierung des Workflow-Musters ist in der vorliegenden Form durch das WFMS des SAP R/3-Systems ablauffähig. Das Workflow-Muster wird gestartet, sobald die Antragsdaten erfaßt wurden. Bei Ablauf des Prozesses bekommt der zuständige Sachbearbeiter - nach Ende eines Hintergrundschritts des Laufzeitsystems, wo Studentendaten von der Datenbank gelesen werden - die Aufgabe „Antrag prüfen“ zugewiesen. Diese Aufgabe wird im sog. „Integrierten Eingangskorb“ angeboten (vgl. Abb. 5-4).



**Abbildung 5-4: „Integrierter Eingangskorb“**

Wählt der Sachbearbeiter im „Integrierten Eingangskorb“ die Aufgabe aus und startet sie, wird die dem Prozeßschritt zugrundeliegende Aufgabe ausgeführt. In diesem Fall wird die Anwendung „Antragsverwaltung“ aktiviert und dem Sachbearbeiter die Prüfung des Antrags ermöglicht (vgl. Abb. 5-5). Die „Antragsverwaltung“ besitzt einen Status, den „Antragsstatus“, über den der Antrag beschieden wird. Weiterhin sind die Daten des Antrags als „Antragspositionen“ aufgeteilt. So können die relevanten Felder des Antrags gezielt geprüft werden. Im dargestellten Beispiel setzt der Sachbearbeiter den Antrag auf „genehmigt“, wodurch die Abarbeitung des Prozesses mit dem nächsten Schritt fortgesetzt werden kann.

Der „Integrierte Eingangskorb“ ist die Anwenderschnittstelle des WFMS des SAP R/3-Systems. In diesem erhalten die Beteiligten des Workflows die Aufgaben der Prozessschritte zur Ausführung dargestellt<sup>429</sup>.

The screenshot displays the SAP 'Antrag: Ändern' (Request: Change) screen. The interface is divided into several sections:

- Header:** 'Antrag: Ändern' with sub-tabs 'Antrag prüfen', 'Kompletzanzeige', and 'Anzeige\_Historie'.
- Antrag (Request):**
  - Antrag: 50016951
  - Beginndatum: 15.08.1999 bis 31.12.9999
  - Antragskopf: Antragsart (Zulassung (Stud)), Grund ( ), Antragsgruppe ( ), Verknüpfungs-ID ( ), Gültig ab ( ), Gültig bis ( ), Benutzername (C5006780), Änderungsdatum (21.08.2000).
- Antragsteller (Applicant):**
  - AntragstellerTyp: ST Student
  - Kennzeichen: 50016998
  - Vorname: Roland
  - Nachname: Klier
- Verbale Beschreibung (Verbal Description):**
  - Allgemeine Beschreibung: Genehmigt ohne Beanstandungen
- Antragstatus (Request Status):**
  - Neuer Status: 0 genehmigt
  - Aktueller Status: 4 vollständig
- Status History Table:**

Antragsstatus	Nachfolgerstatus	Gültig ab	Gültig bis	Benutzer	Datum
	erfasst	15.08.1999	21.08.2000	C5006780	21.08.2000
erfasst	vollständig	21.08.2000	31.12.9999	C5006780	21.08.2000

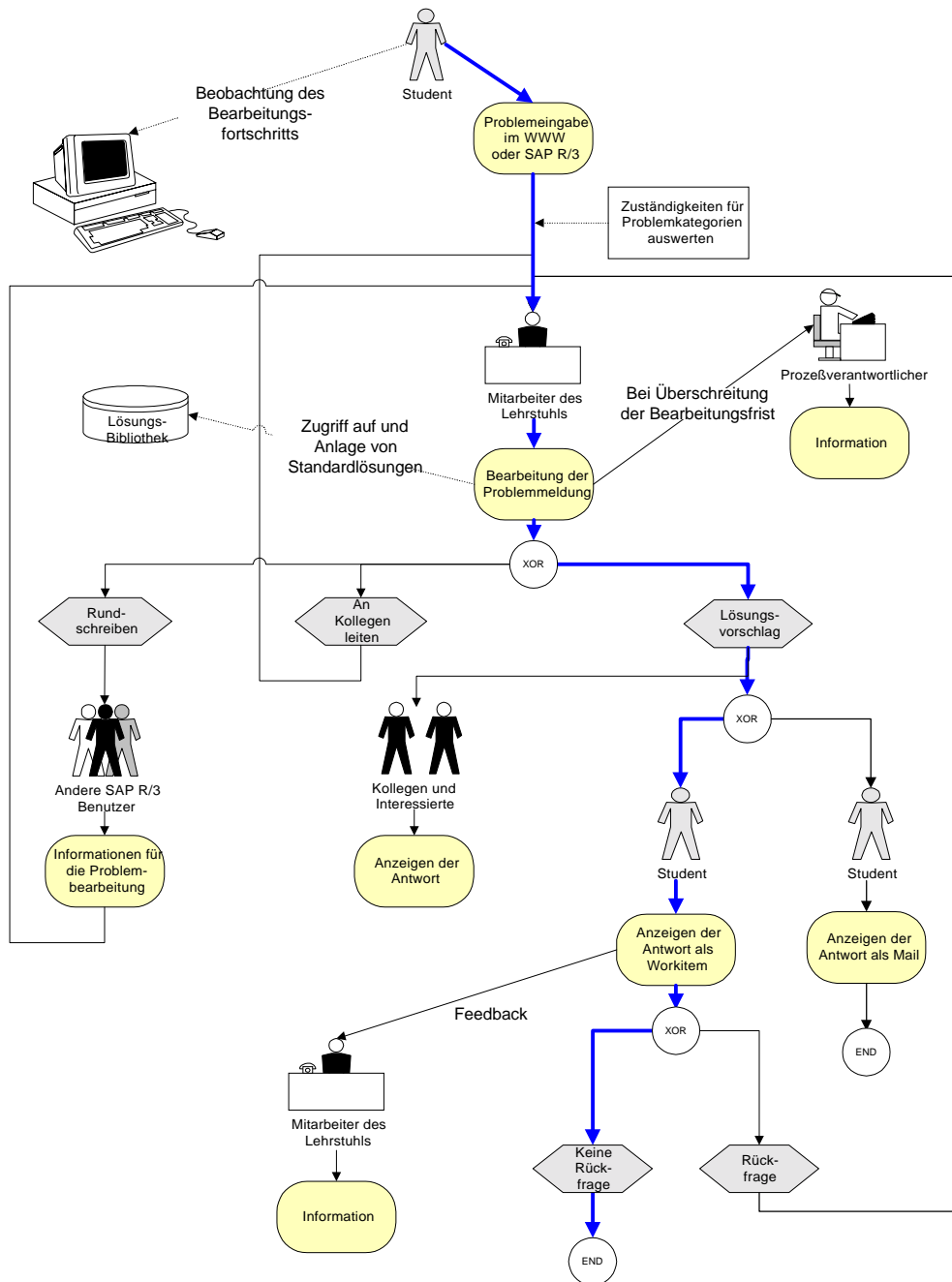
Abbildung 5-5: Antragsprüfung durch Sachbearbeiter als Prozessschritt

<sup>429</sup> vgl. Abschnitt 2

---

Der nächste Prozeß „Problemmeldung“ ist hingegen als vollständiger Prozeß lehrstuhlspezifisch, also entsprechend den Anforderungen des Lehrstuhls, modelliert. Die schematische Darstellung dieses Prozesses ist in Abb. 5-6 zu sehen. Dabei gibt der Student online seine Problemmeldung ein. Der Service „Problemmeldung“ ist, wie oben beschrieben, Bestandteil des Lehrstuhlportals.

Die Problemmeldung wird zur Bearbeitung an einen der Experten des betroffenen Teilgebiets geschickt. Dieser kann ggf. die Aufgabe an Kollegen weiterleiten oder Rückfragen stellen und schließlich die Lösung an den Studenten schicken. Stellt sich bei der Problemmeldung ein „internes“ Problem heraus, z.B. Fehler in der Installation oder unzureichendes Customizing, besteht die Möglichkeit per internem Rundschreiben eine generelle Lösung herbeizuführen. So könnten die verantwortlichen Personen Verbesserungen der Installation oder Erweiterungen bzw. Änderungen im Customizing beschließen und durchführen.



**Abbildung 5-6: Übersicht des Prozesses „Problemmeldung“**

Zu diesem Prozeß wurde auch ein entsprechender Workflow entwickelt. Die Zuordnung der Aufgaben zu den Mitarbeitern, das Verschicken der Aufgaben sollte automatisch, d.h. vollständig elektronisch, verlaufen. Der Student kann die Lösung per email oder in seinem „Integrierten Eingangskorb“ erhalten. Bei unvollständiger Fehlerspezifikation ist hingegen eine Rückfrage erforderlich.

Die - vom Studenten ausgefüllte - Eingabemaske im Self-Service ist in Abb. 5-7 zu sehen. Dort ist die Kategorie des Problems festzulegen, die für die Bearbeiterzuordnung genutzt wird, sowie die Problemmeldung einzutragen. Mit dem Anlegen der Problemmeldung wird der Workflow „Problemmeldung“ gestartet.

The screenshot shows a web browser window with the following content:

Lehrstuhl für Wirtschaftsinformatik - Microsoft Internet Explorer

http://www.wifw.uni-passau.de/wifw/index.html

Navigation: [Einschung](#) [Leben](#) [Veranstaltungen](#) [Mitarbeiter](#) [SAP® R/3®](#) [Self-Service](#) [Self-Service](#)

### FRAGEN ZUM R/3-SYSTEM DES LEHRSTUHL

MATRIKELNUMMER	67342	NUMMER	1038
NAME	Stefan Sand	ERFASSUNG/DATUM	22.08.2000
BENUTZERKENNUNG	SANDS2	ERFASSUNG/ZEIT	22:13:40
SEMESTER	5	STATUS	IN ERFASSUNG
STUDIENFACH	Betriebswirtschaftslehre	SPRACHE	Deutsch
E-MAIL	stefan.sand@stud2.rz.uni-passau.de	PRIORITÄT	mittel
TELEFONNUMMER	0651/33289		

PROBLEMKATEGORIE:  MODULKÜRSEL:

**MARKUS SCHNECKENBURGER**

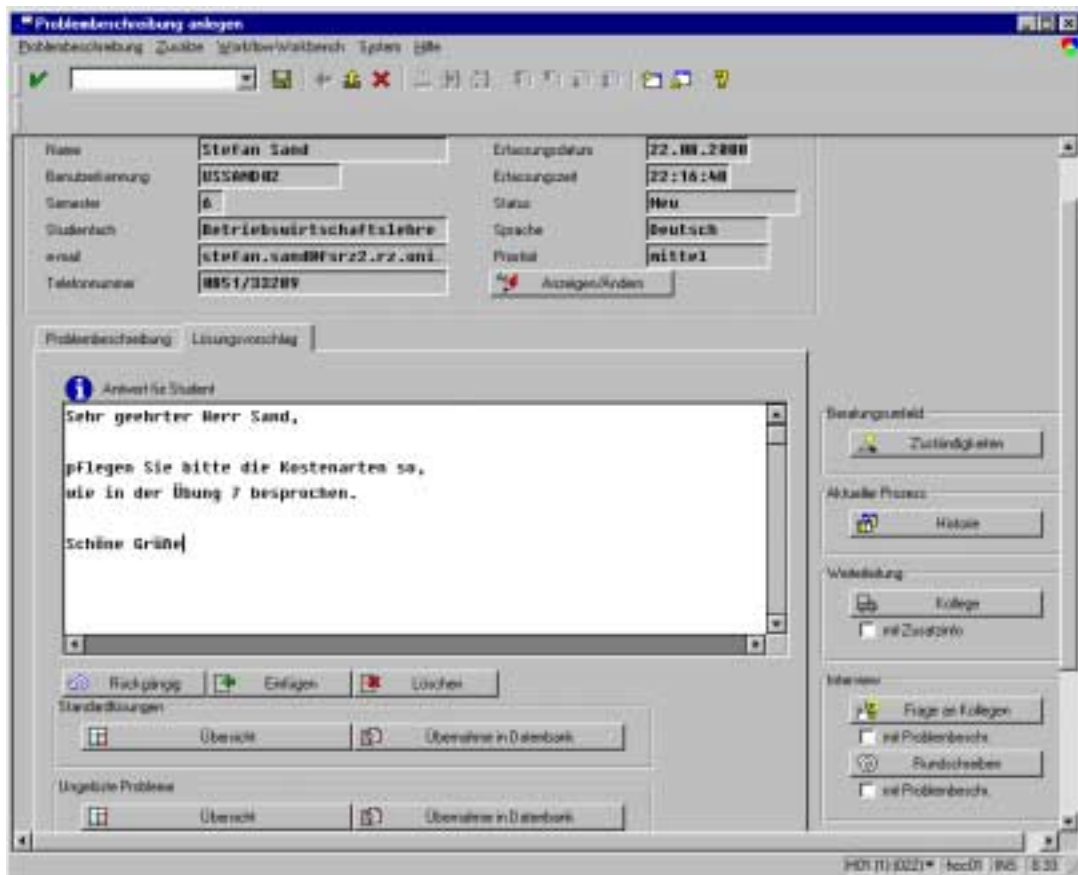
TRANSAKTIONSCODE:

MEN/FOLGE VOR DER PROBLEMMELDUNG:

Wir können sich bei der möglichen Erstellung Ihrer Problemmeldung anregen lassen.

**Abbildung 5-7: Eingabe einer Problemmeldung im WWW**

Die Arbeitsoberfläche ist in Abb. 5-8 zu sehen. Der Ansprechpartner für das jeweilige Gebiet am Lehrstuhl erhält per Workflow die Aufgabe „Problemmeldung bearbeiten“ zugeschickt. Er kann die Beschreibung des Studenten auf Vollständigkeit der Fehlerbeschreibung überprüfen und eine Lösung ausarbeiten. Überdies besteht die Möglichkeit einen Kollegen anzuschreiben oder ein Rundschreiben zu starten.



**Abbildung 5-8: Bearbeitung einer Problemmeldung durch einen Mitarbeiter des Lehrstuhls**

Der Workflow besteht, wie im rechten Teilfenster der folgenden Abb. 5-9 zu sehen, bei grober Unterscheidung aus zwei verbundenen Teilen. Aus dem ersten Teil ist ein Ausschnitt im linken Teilfenster dargestellt. Der erste Teil des Workflows behandelt die Problembehandlung, während im zweiten die Nachbearbeitung des Problems stattfindet. Im dargestellten Ausschnitt findet die Unterscheidung im Prozeß unter Verwendung eines CASE-Konstrukts statt. Abhängig von den Aktionen des Mitarbeiters sind dabei verschiedene Alternativen modelliert. Der Mitarbeiter kann sich entscheiden, ob ein Rundschreiben an Kollegen gestartet werden, eine Rückfrage an den Kollegen stattfinden oder auf eine Rückfrage geantwortet werden soll.

Die Problemmeldung enthält verschiedene Elemente<sup>430</sup> zur Einzelprozeßflexibilisierung, wie die dynamische Bearbeiterauswahl, d.h. Festlegung der Bearbeiter

<sup>430</sup> vgl. „Late-Modeling“ in Abschnitt 4.4

zur Laufzeit, oder die dynamische Auswahl der Art des Bearbeitungsverfahrens, d.h. parallele oder sequentielle Abstimmung<sup>431</sup>. Auch die Integration in das Konzept des Flexiblen und Regelbasierten Workflow-Managements ist durchgeführt worden. Das Konzept zur Einbindung ist in Abschnitt 5.2 dargestellt.

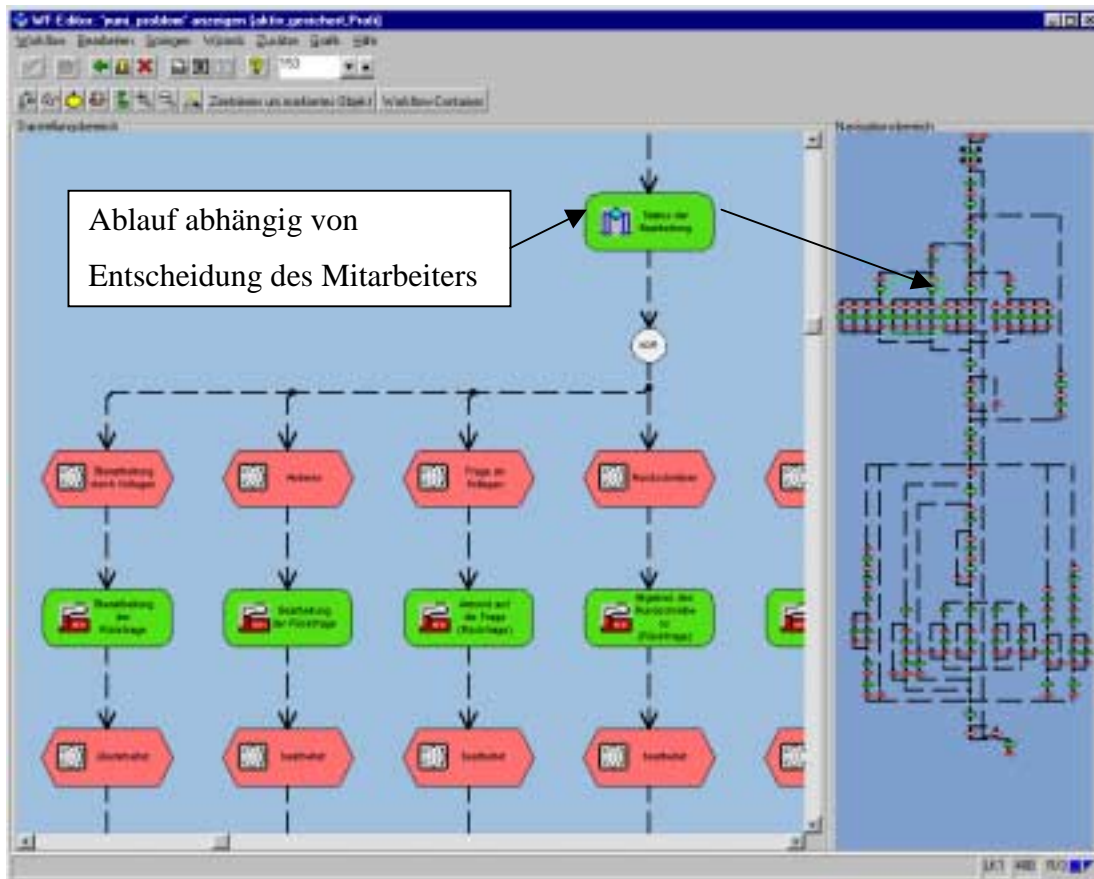


Abbildung 5-9: Ausschnitt des Workflows „Problemmeldung“

<sup>431</sup> Ähnliche Mechanismen liegen auch den Workflows „Studienberatung“ und „Sprechstundenvereinbarung“ zugrunde.



### 5.1.3 Integrationsaspekte

Als wesentlicher Integrationsaspekt des Workflow-Managements mit dem WWW sind zu nennen:

- die Initiierung der Prozesse aus dem WWW,
- die Bearbeitung der Prozesse sowie
- die Möglichkeit, einen Überblick über den Prozeßstand zu erlangen.

Diese Aspekte sollen nun am Beispiel der Prozesse „Studienwunsch“ sowie „Evaluation“ dargestellt werden.

#### **Überblick über Prozeßstand**

Wie im obigen Abschnitt zum Muster-Workflow „Studienwunsch“ gezeigt, besteht deren Muster aus mehreren Schritten. Der Sachbearbeiter kann den Antrag in verschiedene Status setzen, wie „genehmigt“, „abgelehnt“, „unvollständig“ oder „vollständig“. Die traditionelle Methode, den Studenten per Post zu informieren, kann inzwischen per email geschehen. Eine sinnvolle Alternative dazu stellt das Angebot an den Studenten dar, den Status seiner Anträge per Self-Service direkt abzufragen. Der Self-Service „Antragsverfolgung“ ist Bestandteil des Portals „Student Service Bench“. Vor dem Aufruf der Dienste ist allerdings eine Autorisierung des Studenten erforderlich.

Abb. 5-10 zeigt das Ergebnis der Abfrage eines Studenten. Der Student hat sich den Status der Zulassung für einen Zeitraum anzeigen lassen. Die Auswahl erfolgte im rechten oberen Frame. Die Ergebnisse der Abfrage, d.h. die Antragsstatus, werden als Liste der gestellten Anträge im rechten unteren Frame dargestellt.

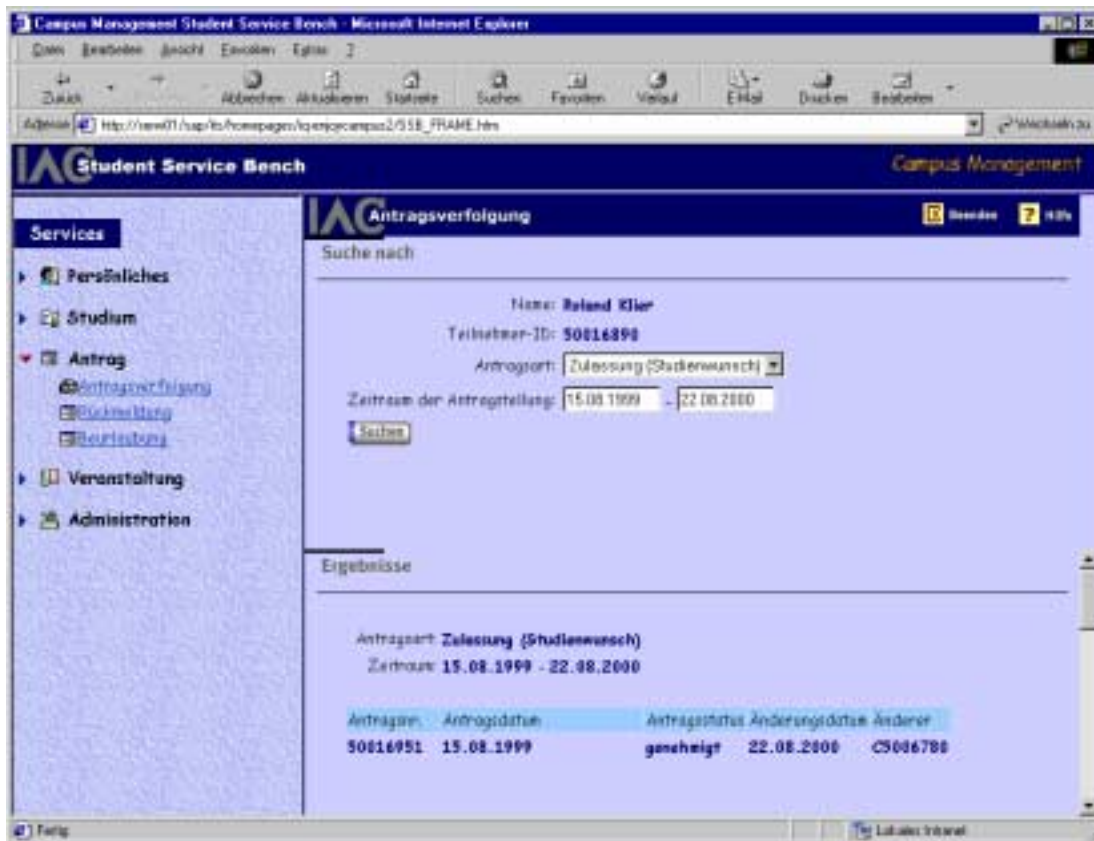
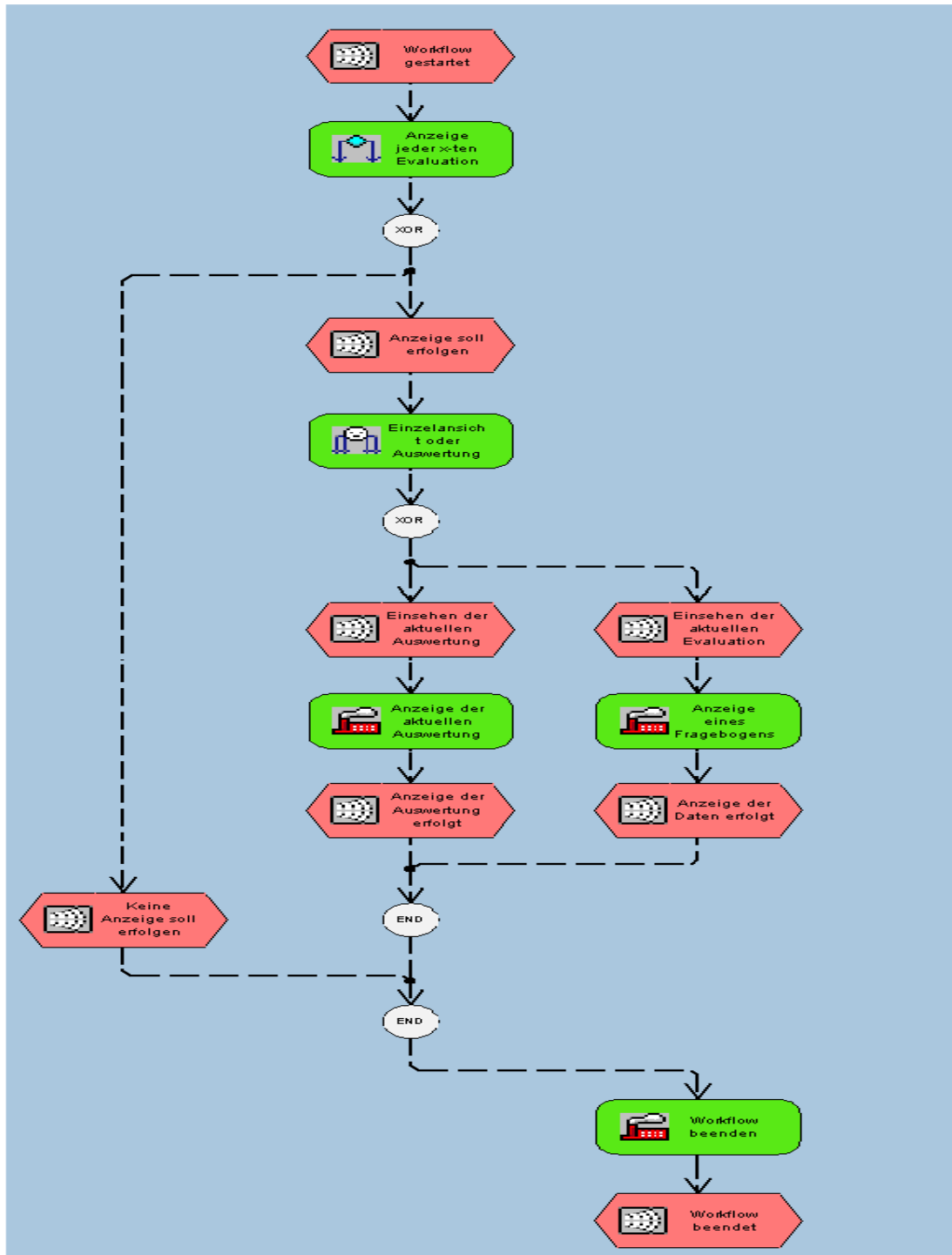


Abbildung 5-10: Self-Service „Antragsverfolgung“

### Prozeßinitiierung aus dem WWW

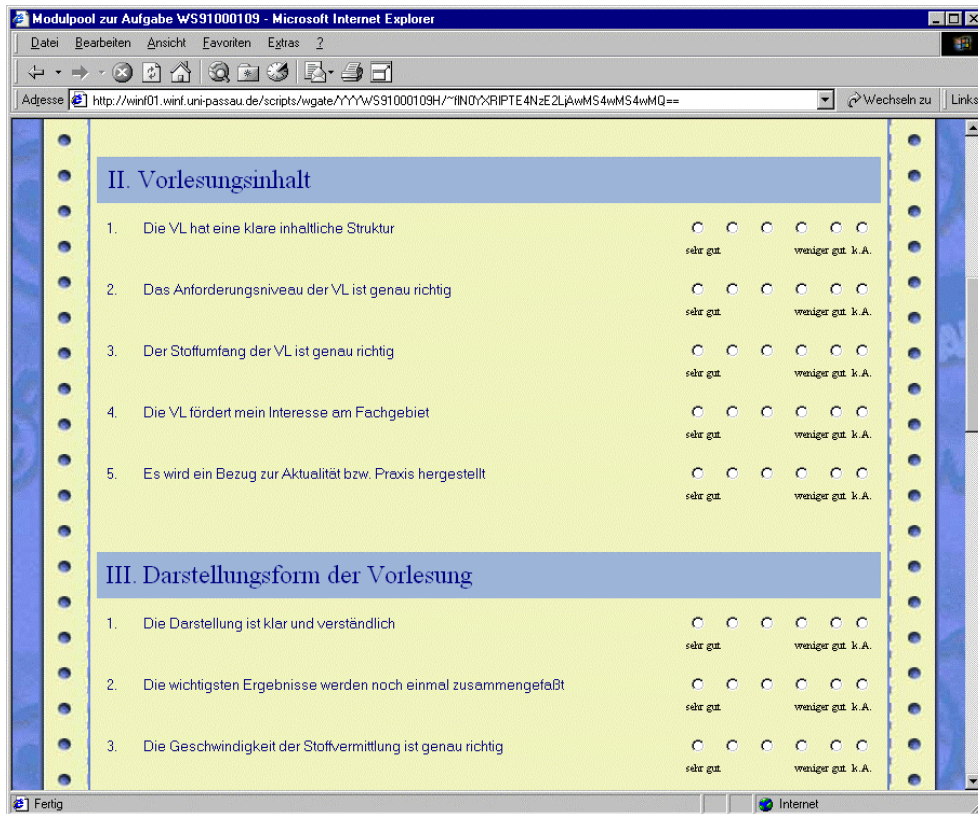
Die Prozeßinitiierung wurde bereits am Prozeß der Problemmeldung gezeigt (vgl. Abb. 5-2). Demnach können Problemmeldungen online im Portal angelegt werden. Als weiteres Beispiel für das Anstoßen eines Prozesses aus dem WWW ist der Prozeß „Evaluation“, an dem auch die durchgängige Bearbeitung gezeigt werden soll.

Der Prozeß der Evaluation repräsentiert die Möglichkeit einer kontinuierlichen Bewertung von Vorlesungen durch die Studenten. Diese können ihre Ergebnisse im Self-Service erfassen. Nach der Erfassung werden die Daten jeder Evaluation mit den bisherigen zu einer Gesamtbewertung verdichtet. Die veranstaltende Person kann sich dann wahlweise die Einzel- oder die Gesamtergebnisse anzeigen lassen. So ist eine schnelle Reaktion auf positive, aber auch unerwartete Bewertungen möglich. Der in der Abb. 5-11 dargestellte Workflow „Evaluation“ besteht aus mehreren Schritten, wobei die Auswahl als Workflow-Statement modelliert wurde.



**Abbildung 5-11: Workflow „Evaluation“**

Die Abfolge der Initiierung aus einem Self-Service ist in der folgenden Abb. 5-12 zu sehen. Um eine Evaluation durch die Studenten durchführen zu lassen, müssen vorher anonyme ID und Zahlen für Mehrfachbewertungen erzeugt werden. So können die neuen Bewertungen als Reaktion auf eine veränderte Vorlesungs-gestaltung beobachtet werden.



**Abbildung 5-12: Self-Service „Evaluation“ – Eingabe durch den Studenten**

Das Ergebnis der Bewertung kann jederzeit über WWW-Clients eingesehen werden.

### **Durchgängige Prozeßbearbeitung im WWW**

Eine durchgängige Prozeßbearbeitung setzt eine Benutzerschnittstelle zu dem Workflow-Management im WWW voraus. Die einzelnen Anwendungen, die in den Prozeßschritten aufgerufen werden, sollten daher sowohl im traditionellen Eingangskorb als auch im WWW-Eingangskorb ablaufen. Wie im Portal des Lehrstuhls zu sehen ist, steht dort der Dienst des „Integrierten Eingangskorbs“ im WWW zur Verfügung. Um diesen Service der SAP AG nutzen zu können, sind Erweiterungen in den Anwendungen erforderlich, um die parallele Bearbeitung komplexer Aufgaben des Workflow-Managements auch im WWW zu gewährleisten.

Die durchgängige Bearbeitung der Evaluation soll weiter dargestellt werden. Der Aufruf des „Integrierten Eingangskorbs“ ist aus dem Self-Service möglich. Dort sind die Aufgaben, ähnlich dem R/3-Eingangsprozeß als Liste zu sehen, diesmal

im rechten unteren Frame. Startet der Sachbearbeiter die Aufgabe, erhält er das aggregierte Gesamtergebnis der einzelnen Evaluationen (vgl. Abb. 5-13).

Stand der Auswertung: 25.09.1999										
Veranstaltungs-ID:	Veranstaltung:	Notenverteilung					Kennzahlen			
1.	2.	3.	4.	5.	wn	avg.	>3	Prg		
II.	Vorlesungsinhalt									
1.	Die VL hat eine klare inhaltliche Struktur	12	9	4	1	144	1,81	3,85	1	
2.	Das Anforderungsniveau der VL ist genau richtig	5	9	7	4	144	2,40	16,00	2	
3.	Der Stoffumfang der VL ist genau richtig	5	5	7	5	144	2,65	26,09	3	
4.	Die VL fördert mein Interesse am Fachgebiet	4	3	5	2	7	144	3,24	42,86	4
5.	Es wird ein Bezug zur Aktualität bzw. Praxis hergestellt	3		3	2	5	150	3,46	53,85	5
III.	Darstellungsform der Vorlesung									
1.	Die Darstellung ist klar und verständlich	3	2	1	1	5	148	3,25	50,00	4
2.	Die wichtigsten Ergebnisse werden noch einmal zusammengefaßt	1	2	3	4	1	148	3,18	45,45	1
3.	Die Geschwindigkeit der Stoffvermittlung ist genau richtig	1	1	5	3	1	148	3,18	36,36	2
4.	Das Begleitmaterial (Skripten und/oder angegebene Literatur) ist hilfreich		4	1	2	2	148	3,22	44,44	3
5.	Der Vorlesungsstoff ist für mich nachvollziehbar	2	1	1	3	3	148	3,40	60,00	5
6.	Hilfsmittel (Tafel, Projektor o.ä.) werden sinnvoll eingesetzt	1	1	1	2	3	148	4,00	71,43	6
IV.	Zur Person des Dozenten (alle Fragen gelten analog auch in der weiblichen Form)									
1.	Der Dozent wirkt gut vorbereitet		2	2	1		149	2,80	20,00	2
2.	Der Dozent vermittelt ein persönliches Engagement		2	1	1		150	2,75	25,00	1
3.	Der Dozent geht auf Fragen und Wünsche der Studenten ein	1	1			2	149	3,25	50,00	5
4.	Der Dozent kann auch schwierige Sachverhalte gut erklären		2		3	1	148	3,50	66,67	6
5.	Der Dozent beweist Flexibilität bei Verständnisschwierigkeiten		1	2	1		150	3,00	25,00	3
6.	Der Dozent ist für die Studenten nach der VL ansprechbar		2	1		1	149	3,00	25,00	4
V.	Gesamtbeurteilung	1	2	3	4	5	wn	avg.	>3	Prg

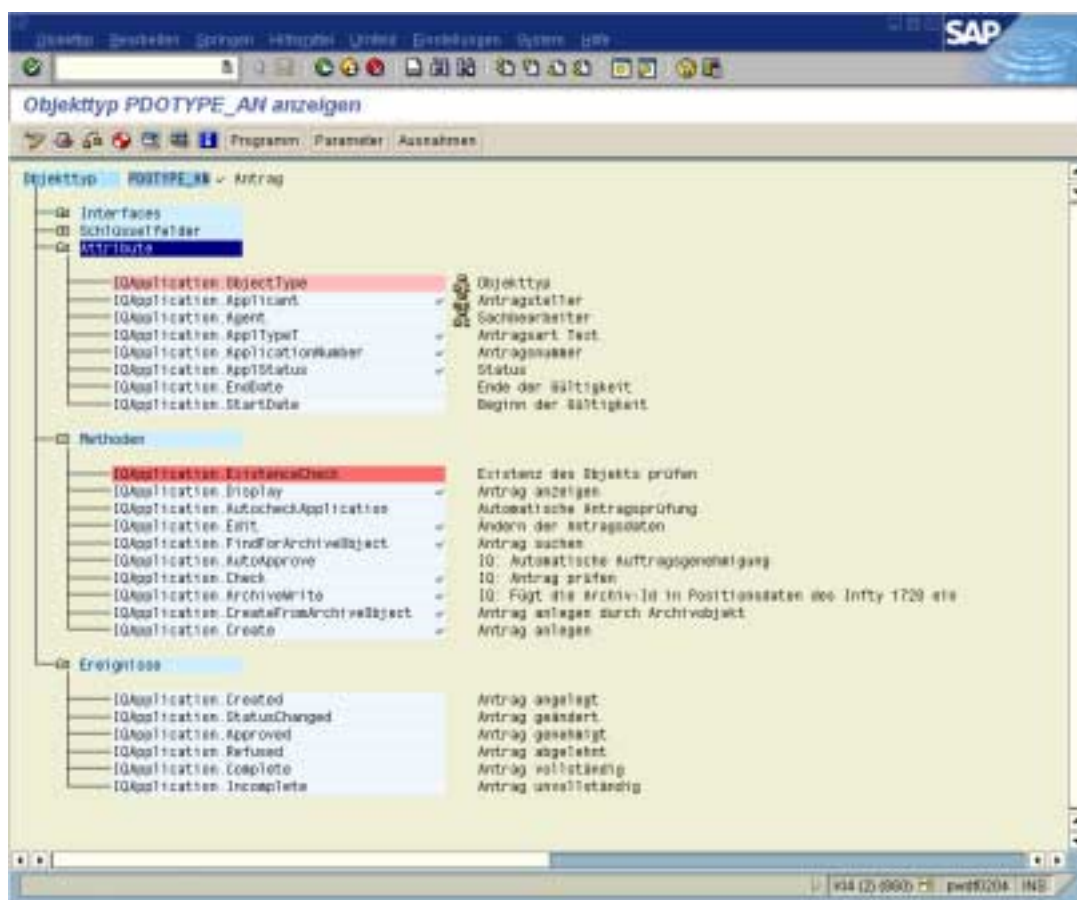
Abbildung 5-13: Aggregiertes Ergebnis der Evaluation

#### 5.1.4 Implementierung

Die Implementierung soll im folgenden für ausgesuchte Details der Workflow-Entwicklung sowie der Workflow-Integration beschrieben werden. Die Workflow-Modellierung hat, wie schon gezeigt, verschiedene Aspekte. Neben dem Verhaltens- und Funktionsaspekt sowie dem Organisationsaspekt, die im Workflow-Editor des SAP R/3-Systems modelliert werden, beginnt die eigentliche Implementierung auf Prozeßschrittebene. Dort müssen die Anwendungen aufgerufen werden<sup>432</sup>. Der Ansatz der Anwendungsintegration bei der Workflow-Gestaltung im SAP R/3 unterhalb der Workflow-Prozesse ist objektorientiert. Die einzelnen Prozeßschritte sind- wenn nicht wieder Schritte der Prozeßsteuerung,

<sup>432</sup> vgl. „Operationsaspekt“ in Abschnitt 2.2

wie Schleifen oder Teilprozeßaufrufe, - Methoden von Objekten<sup>433</sup>. Diese Objekte werden im sogenannten Business Object Repository (BOR) spezifiziert. Grundlage der einzelnen BOR-Objekte sollte allerdings eine generelle Objektmodellierung der Objekte und deren Beziehungen sein, die der Anwendung zugrundeliegen. Das BOR verfügt über eine eigene Makrosprache. Zum Objektmodell im „Campus Management“ ist das Objekt „Antrag“ im BOR spezifiziert worden (vgl. Abb. 5-14). Im BOR sind zu Objekten u.a. Attribute und Methoden zu definieren. Im Falle des Objekts „Antrag“ sind dies Attribute wie „Status“ aber auch Verweise auf andere Objekte wie den „Antragsteller“.

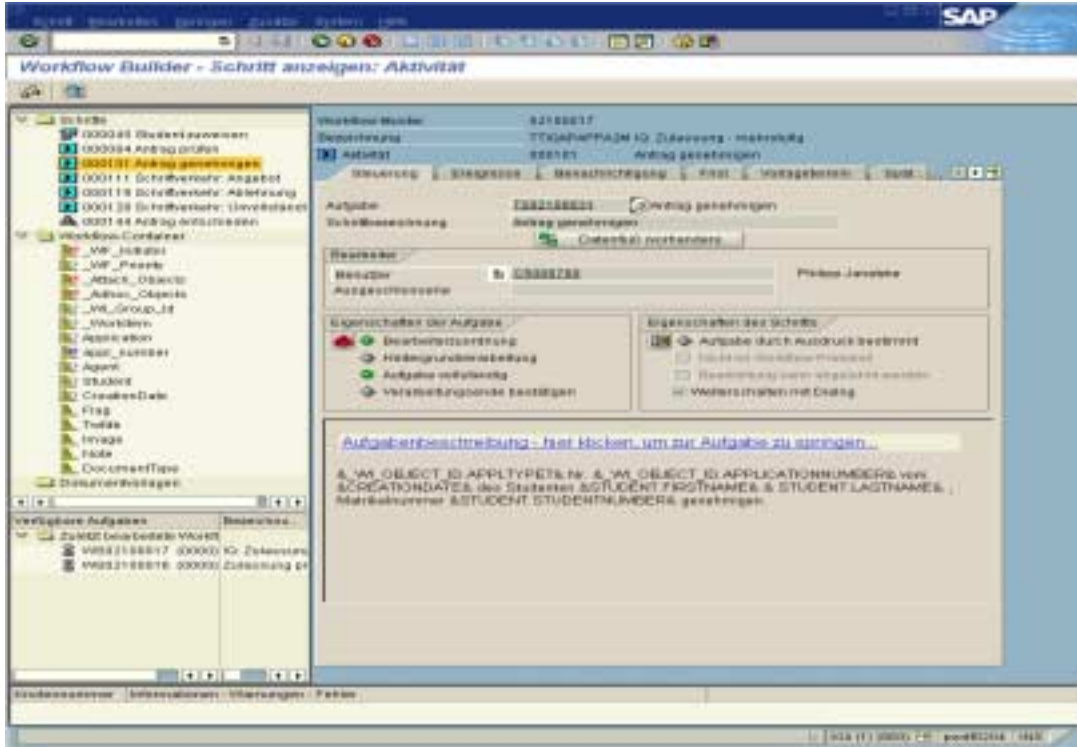


**Abbildung 5-14: Objekt „Antrag“ im Business Object Repository**

Eine zentrale Methode des Antrags ist die des Prüfens, die im Workflow-Muster zweimal verwendet wird. Der Einbezug der Methoden ist über Aufgaben möglich, die am Prozeßschritt als sog. Aufgabe hinterlegt werden (vgl. Abb. 5-15). Als zu-

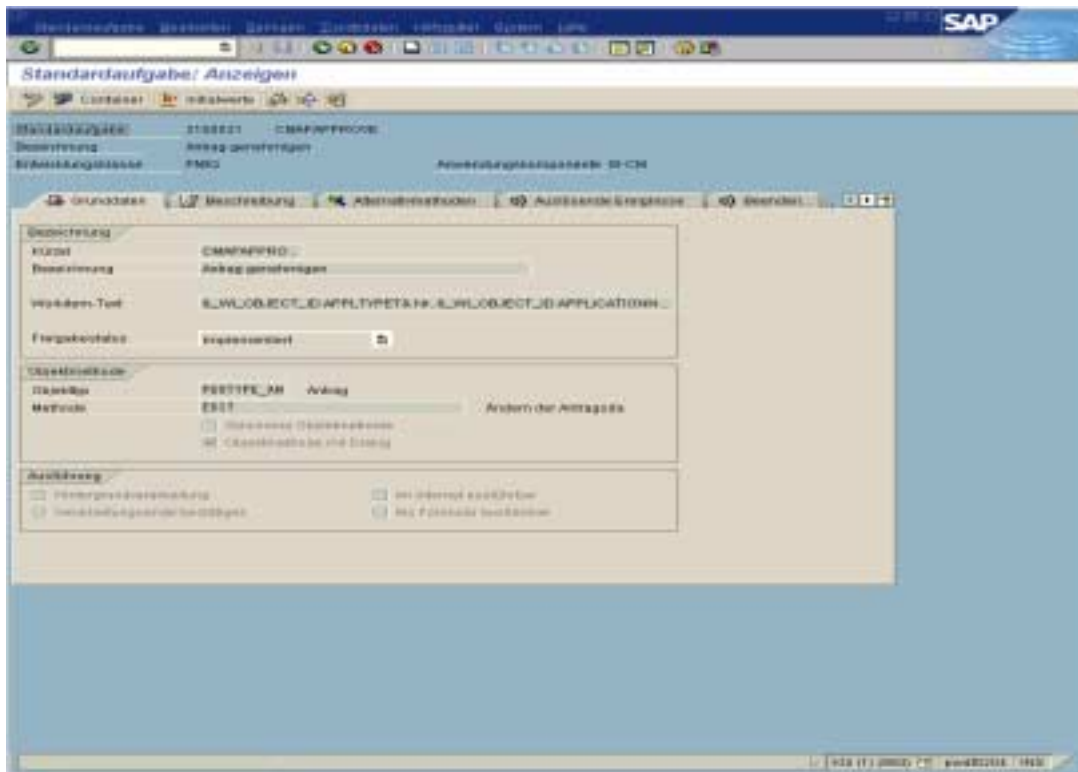
<sup>433</sup> vgl. Abschnitt 2.3

ständiger Bearbeiter wurde der Sachbearbeiter C5006780 eingetragen. Bei der Definition der Aufgabe schließlich ist die Methode „Edit“<sup>434</sup> des Objekts „Antrag“ hinterlegt (vgl. Abb. 5-16).



**Abbildung 5-15: Einbettung der Aufgabe TS0210031, in der die Methode „Edit“ des Antragsobjekts verwendet wird, im Prozessschritt „Antrag genehmigen“**

<sup>434</sup> Die Methode „Edit“ wird anstelle von „Check“ verwendet, wenn neben der Prüfung auch Änderungen der Daten vorgesehen sind.



**Abbildung 5-16: Aufruf der Methode „Edit“ des Objekts „Antrag“ in der Aufgabe TS0210031**

Nun soll die Verwendung der Makrosprache des Methodenaufrufs beschrieben werden. Die Makros bieten u.a. die Möglichkeit Daten, die per Datenfluß an die Methode übergehen werden, aus dem sogenannten Container zu entnehmen. Der Zugriff auf Attribute des Objektes lassen sich mit dem Makro `swc_get_property` durchführen (vgl. Abb. 5-17), die Entnahme von Containerwerten wiederum ist durch das Makro `swc_get_element` möglich.

Die Variablen können, durch den Datenfluß zur Verfügung gestellt und per Makro abgeholt, dann in der eigentlichen programmierten Anwendung verwendet werden. Durch Aufruf der Funktion `Antrag prüfen „HRIQ_APPLICATION_EDIT“`, der in der Programmiersprache ABAP implementiert ist, wird schließlich die Anwendung „Antragsverwaltung“ eingebunden. Das Objekt „Antrag“ stellt damit durch seine Methoden und Attribute eine objektorientierte Kapselung der „Antragsverwaltung“ für den Workflow dar. Die eigentliche Funktionalität ist in der SAP R/3 spezifischen Basissprache ABAP zu programmieren.



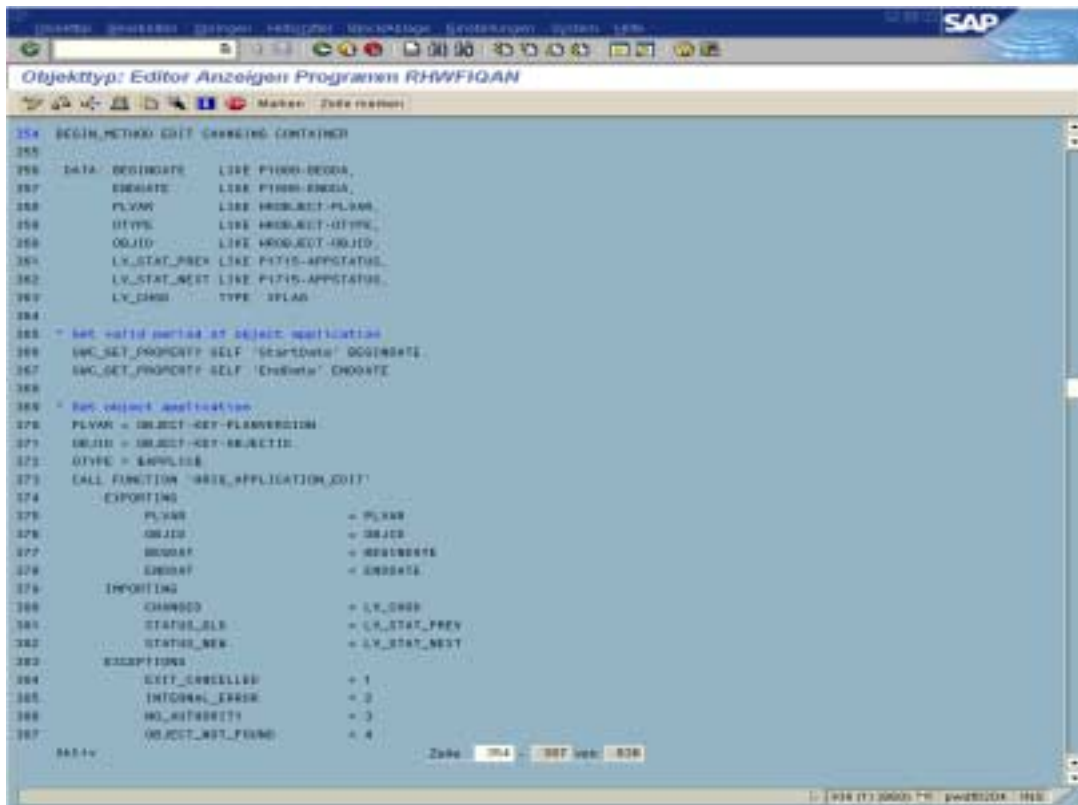


Abbildung 5-17: Implementierung BOR-Methode Antrag

Die Integration des WWW innerhalb des SAP R/3-Systems, wobei das WFMS auch als integrierte Anwendung Teil des Systems ist, geschieht über den speziellen Internet Transaktionsserver (ITS)<sup>435</sup>

Der Self-Service „Antragsverfolgung“, der indirekt Informationen über antragsbasierte Prozesse auswertet, ist im folgenden genauer beschrieben. Der Aufruf des Self-Service „Antragsverfolgung“ (vgl. Abb. 5-1) wird als Parameter über die CGI –Schnittstelle an den ITS übergeben. Der ITS besteht aus zwei Teilen. Der eine Bestandteil, der sog. WGate, leitet den CGI-Aufruf an den AGate, den anderen Teil, weiter. Damit ist die Aufgabe des WGate insbesondere die Weitergabe des Aufrufs. Die Funktion der Kopplung ist wiederum im AGate implementiert. Dort werden dann die zu dem Service gehörigen Anwendungen aufgerufen. In diesem Fall sind dies sog. „Web-Transaktionen“. „Web-Transaktionen“ bilden im SAP R/3-System Einheiten, die in ABAP programmiert werden und sich über verschiedene Bildfolgen erstrecken können. Diese speziellen Transaktionen für das WWW laufen dann im SAP R/3-System ab und senden das Startbild an den ITS.

<sup>435</sup> vgl. Abschnitt 2.3

Dieser setzt die Daten des Bildes in eine HTML-Schablone ein und sendet die gefüllte Schablone an den WWW-Browser.

In Abb. 5-18 ist die Schablone, das sog. „HTML-Template“, zu sehen. Dies ist eine typische HTML-Seite, die spezifische Befehle der Sprachverwaltung von HTML durch SAP, den sog. HTML-Business, ergänzt<sup>436</sup>. Beispielsweise sind die einzufüllenden Variablen benannt, so daß der ITS die richtige Zuordnung der Variablen zu den Bildschirmfeldern vornehmen kann.

```

<html>
<head>
<title>~windowtitle~</title>
<link REL="stylesheet" HREF="~mimetURL("service="global", "style/style.ess")" TYPE="text/ess">
</head>
<body bgcolor="~TTTTT" text="~BBBBBB" link="~BFAAAB" vlink="~BFAAAB" leftmargin=0 topmargin=0 marginheight=0 marginwidth=0>
<table border=0 cellspacing=0 cellpadding=0 width=100% bgcolor=black>
<tr>
<td align=left>
<td align=right><B HREF="~ajgateURL("Bcnde="/~noc", "target="parent")" target="parent">
</a>
<B HREF="~mimetURL("name="info/docu_piqsaptrack.stml")" target="_new">
</a>
</td>
</tr>
</table>



<table border=0 cellpadding=0 cellspacing=0 width=100%>
<form action="~ajgateURL("TARGET="SUB2", "FORCETARGET="TRUE")" method="post">
<tr>
<td align=right><input type="text" name="SEARCH"></td>
<td align=right><input type="submit" value="Suchen"></td>
</tr>
</form>

<table border=0 cellspacing=2 cellpadding=2>
<tr>
<td align=right class=Label> Name:</td>
<td colspan="2"> ~WA_APP-STUENCH_FIRSHOME ~WA_APP-STUENCH_LBSTHOME</td>
</tr>
<tr>
<td align=right class=Label> Battendeid:</td>
<td colspan="2"> ~WA_APP-STUENCH_HR</td>
</tr>
<tr>
<td align=right class=Label> Appttype:</td>
<td colspan="2">
<SELECT class=Fieldinput NAME="APPLICATIONTYPES-APPTYPE" size=1 repeat with i from 1 to apptype id.dlm
<OPTION IF {APPLICATIONTYPES-APPTYPE == apptype_id[i]} selected ~wa~ value="~apptype_id[i]">~appt:
'esd' </SELECT></td>
</tr>
</table>

```

Abbildung 5-18: HTML Schablone für Aufruf Antragsverfolgung

Der Anwender, d.h. Aufrufer des Self-Service, sieht dann das komplette Bild des SAP R/3 als HTML-Seite. In der WWW-Seite kann er dann über Buttons und Listboxen die Antragsart spezifizieren. Das Ergebnis wird wieder über den ITS an das R/3-System geschickt. Die Transaktion, die zwischenzeitlich im Status „war-

<sup>436</sup> vgl. Abschnitt 2.3

tend“ war, kann weiterlaufen. Man sieht, daß sich der ITS wie der SAPGUI, die typische Client-Anwendung des SAP R/3-Systems, verhält.

Dieser Kopplungsmechanismus durch den ITS wird auch bei der Initiierung der Prozesse angewendet. Der Self-Service „Evaluation“ ist ebenfalls eine „Web-Transaktion“. Startet der Anwender den Self-Service, wird die Transaktion aufgerufen, das HTML-Template gefüllt und an den Anwender geschickt. Dieser macht die Eingaben und schickt sie über den ITS an die „Web-Transaktion“. Diese sammelt dann die Daten und führt die Anbindung an den Workflow durch, indem sie mit den gesammelten Daten eine Instanz des BOR-Objekts „Evaluation“ anlegt (vgl. Abb. 5-19). Dazu wird die Methode „Create“ des Objekts aufgerufen. Durch den Anlegebaustein „Create“ des Objekts wird dann dessen Ereignis „created“ ausgelöst.

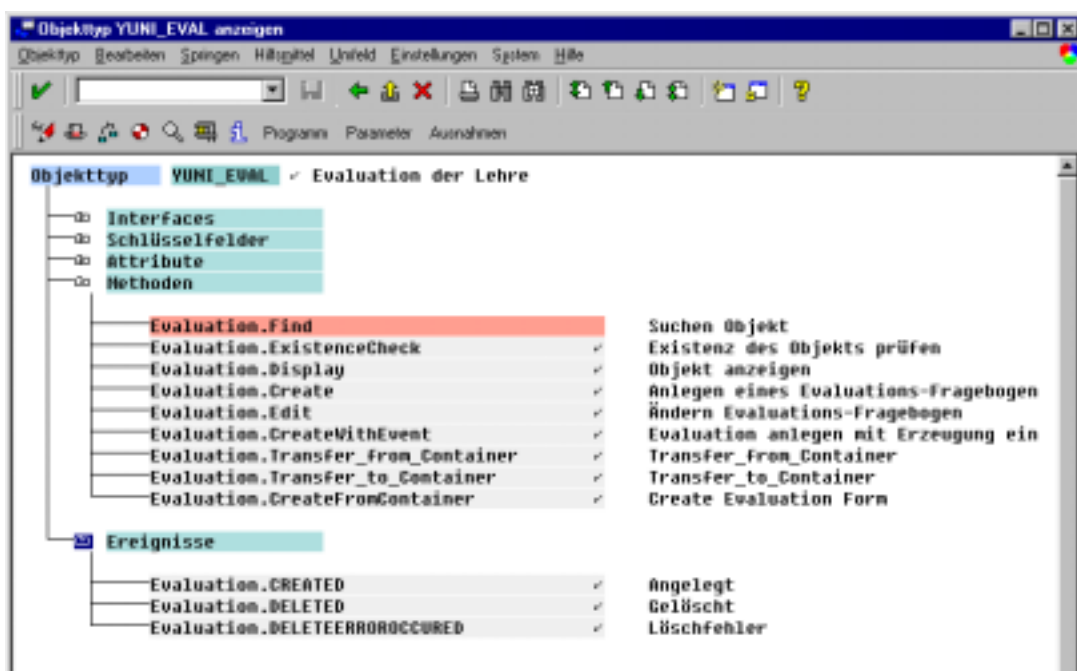


Abbildung 5-19: Objekt „Evaluation“ des Business Object Repository

Die Ereignisse der Objekte können als Startereignisse am Workflow hinterlegt werden. D.h. wird die Objektinstanz zum Evaluationsobjekt angelegt, wird das Ereignis „created“ ausgelöst und alle Workflows, die auf dieses Ereignis warten oder als Startereignis besitzen, mit der aktuellen Objektinstanz initiiert. Nach dem gleichen Prinzip arbeiten auch die anderen Services, die Prozesse initiieren, wie „Antrag auf Rechnerkennung“ oder „Interessentenregistrierung“.

Die nächste Stufe der Zielsetzung stellt die Aufgabe der durchgängigen Prozeßbearbeitung dar. In diesem Fall muß über dem Service „Integrierter Eingangskorb“ das WFMS über den ITS aufgerufen werden. Die Erweiterungen in der Anwendung „Evaluation“ müßten in den Container des Workflows „Evaluation“ zurückgreifen, um die Datenvorwahl an den ITS weiterzugeben. Damit wird die Datenflußebene nachempfunden.

## 5.2 Realisierung zum Konzept des Flexiblen und Regelbasierten Workflow-Managements

Im folgenden soll die Realisierung eines Ausschnitts des „Konzepts des Flexiblen und Regelbasierten Workflow-Managements“ beschrieben werden. Dabei lag bei der Realisierung zur anwenderorientierten Prozeßbeziehung<sup>437</sup> der Fokus auf der Operation „Prozeß initiieren“. Ziel war eine durchgehende Umsetzung von der Modellierung bis zur Laufzeit. Die Realisierung zur regelbasierten Prozeßbeziehung<sup>438</sup> wurde im Hinblick auf die Synchronisation von Prozessen durchgeführt. Dieser Mechanismus wurde dabei exemplarisch für die antragsbasierten Prozesse realisiert. Aufgrund der Implementierung innerhalb der Anwendung des Antragswesens, die unterhalb der Steuerung durch das WFMS liegt, stellt dieser Weg damit eine implizite Umsetzung der Operation „Prozeß warte“ dar. Traditionelle Prozeßbeziehungen<sup>439</sup>, beispielsweise in der Form der Ereigniskommunikation zwischen WF, sind bereits Bestandteil von gängigen WFMS.

Die Realisierung besteht aus Erweiterungen auf Basis des WFMS des SAP R/3-Systems, wie z.B. zusätzlichen Tabellen zur Modellierung des Beziehungsaspekts<sup>440</sup>, Funktionsbausteinen zur Abbildung des Beziehungsaspekts durch das WFMS, Erweiterungen der den Prozessen zugrundeliegenden Anwendungen sowie aus dem Werkzeug „VisuFlex“. Weiterhin ist die Definition der Regeln auf Basis des Werkzeugs Validations Substitutions Rules (VSR) des SAP R/3-Systems möglich. Zur strukturierten Verwaltung und Anwendungsintegration größerer Regelmengen wurde ein Rahmenwerk entworfen.

Die Beschreibung dieses Abschnitts ist in vier Unterabschnitte aufgeteilt. Im ersten sollen die grundlegenden Zusammenhänge der Prozeßinitiierung im Rahmen des Konzepts verdeutlicht werden. Dabei werden auch die Aufgaben der Prozeß-

---

<sup>437</sup> vgl. Säule drei des Modells zur Behandlung komplexer Prozeßbeziehungen in Abschnitt 4.2

<sup>438</sup> vgl. Säule zwei des Modells zur Behandlung komplexer Prozeßbeziehungen in Abschnitt 4.2

<sup>439</sup> vgl. Säule eins des Modells zur Behandlung komplexer Prozeßbeziehungen in Abschnitt 4.2

<sup>440</sup> vgl. Abschnitt 4.2.4

modellierer und ein zentrales Werkzeug, „VisuFlex“, für die Unterstützung der Verwaltung und Analyse von PBZ vorgestellt.

Daran schließt sich die Modellierung der PBZ im Detail an. Insbesondere wird die Umsetzung des exemplarischen Beziehungsaspekts dargestellt. Die Regeldefinitionen werden auf der Grundlage des Werkzeugs Validations Substitutions Rules (VSR) vorgenommen.

Auf der Modellierung der PBZ aufbauend wird in Abschnitt 5.2.3 die Realisierung der Durchführung des Konzepts des Flexiblen und Regelbasierten Workflow-Managements und der Auswertungsmöglichkeiten der PBZ dargestellt werden. Dabei wird auf die Aufgaben der Beteiligten, aber auch die Umsetzungsaspekte eingegangen.

Inhalt des letzten Unterabschnitts ist die Umsetzung des Modells, die eine Verankerung in den jeweiligen Anwendungen, wie der Bearbeitung von Anträgen auf Rechnerkennungen oder von Problembeschreibungen, voraussetzt. Zu den innerhalb der Anwendungen durchgeführten Implementierungen werden sowohl die Erweiterungen der Anwendungen, d.h. Objektmethoden des Business Object Repository, zur Abbildung der Operation „Prozeß initiieren“, als auch Implementierungsaspekte der Werkzeuge und Mechanismen beschrieben.

### 5.2.1 Prozeßinitiierung als Ausschnitt des Konzepts des Flexiblen und Regelbasierten Workflow-Managements

Anhand der Operation „Prozeß initiieren“ soll im folgenden das Konzept des Flexiblen und Regelbasierten Workflow-Managements exemplarisch realisiert werden. Zuerst soll eine Übersicht der Definition, der Durchführung und der Auswertung der Prozeßbeziehungen gegeben werden.

Prozeßinitiierung bedeutet, daß von einem Prozeß heraus andere Prozesse gestartet werden können. Dies ist eine wichtige Workflow-Operation der anwenderorientierten Prozeßbeziehung, d.h. Säule drei Modells zur Behandlung komplexer Prozeßbeziehungen (PBZ)<sup>441</sup>. Ähnliche Operationen, wie Anhalten oder Beenden anderer Prozesse, können analog umgesetzt werden.

---

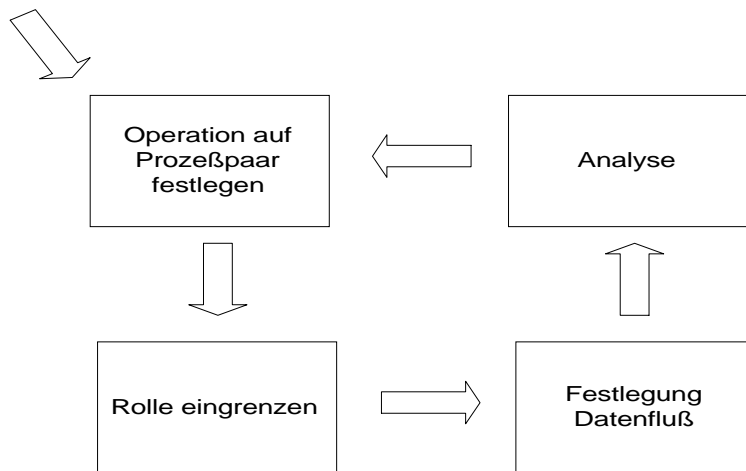
<sup>441</sup> vgl. Abschnitt 4.2

Für diese Workflow-Operation „Prozeß initiieren“ ist die Definition der PBZ, die Unterstützung der Operation und auch die Analyse der durchgeführten Operationen vorzunehmen. Bei der Definition sind also die Prozesse festzulegen, auf denen die Operationen erlaubt sind. Da die Beziehungen zwischen den Prozessen paarweise gelten, sind diese für je zwei Prozesse festzulegen, auf denen die Operationen erlaubt ist. Die den Prozessen zugrundeliegenden BOR-Objekte sind grundsätzlich auf das Konzept zu erweitern. Diese sehr allgemeinen Anwendungserweiterungen werden in der Implementierung im Detail vorgestellt. Gleiches gilt für die Integration der Protokollfunktion, die alle durchgeführten Operationen aufzuzeichnet. Auch die Einschränkung der Operation auf bestimmte Rollen wird vorgenommen. So dürfen diese Operationen nur Personen durchführen, die diese Rolle besitzen. Die Festlegung dieses organisatorischen Bestandteils des Beziehungsaspekts ist wie die Definition des Datenflusses zwischen den Prozessen durchzuführen. Bei der Definition des Datenflusses muß u.a. festgelegt werden, welche Daten bei der Durchführung der Workflow-Operation zwischen den Prozessen ausgetauscht werden sollen und in welcher Richtung die Daten fließen.

Die Protokollierung dient einer genauen Aufzeichnung der Prozeßabläufe und bildet damit die Grundlage für Auswertungen und Analyse. Bei der Protokollierung ist daher z.B. festzuhalten, wer die Workflow-Operation bei welchem Prozeßpaar, bei welchen Prozeßinstanzen und zu welcher Zeit durchgeführt hat.

Die Analyse der Prozesse muß die Definition der Operation auf den Prozessen umfassen, außerdem die protokollierten Daten aufbereiten. Anhand der aufbereiteten Protokolldaten kann ein Vergleich der modellierten Daten mit denen zur Laufzeit stattfinden. Die Laufzeitdaten sollten dabei nach verschiedenen Kriterien wie z.B. Folgen von Prozeßinstanzen oder aggregierten Prozeßpaarbeziehungen ermöglicht werden. Für die Analyse wurde das Werkzeug „VisuFlex“ entwickelt.

Die Aufgaben der Prozeßmodellierer im Rahmen der Modellierung einer PBZ, die sich am exemplarischen Beziehungsaspekt orientiert, sind in Abb. 5-20 nochmals zur Übersicht dargestellt. Nach der Festlegung der Operation auf den Prozeßpaaren ist die Rolle der Personen einzugrenzen, die die Operation ausführen dürfen. Anschließend ist der Datenfluß zu spezifizieren.



**Abbildung 5-20: Aufgaben bei der Modellierung der Prozeßbeziehungen**

Die Analyse der Laufzeitdaten können vom Modellierer selbst bzw. mit den Prozeßbeteiligten erfolgen. Diese kann wiederum zu einer Überarbeitung der Modellierung führen.



## 5.2.2 Modellierung der Prozeßbeziehungen

Die Übersicht über die modellierten PBZ soll über das Werkzeug „VisuFlex“ erfolgen<sup>442</sup>. Aus dieser Darstellung sollen die einzelnen Aspekte der Modellierung der Beziehung verdeutlicht werden.

In der folgenden Abb. 5-21 ist eine Darstellung eines Prozeßnetzes zu sehen. Das Prozeßnetz ist aus Prozeßpaaren aufgebaut, die in Beziehung bzgl. der Operation „Prozeß initiieren“ stehen. So erhält der Modellierer also einen Überblick über alle Prozeßpaare dieser Beziehungsart. Durch paarweise Beziehungen können zur Laufzeit auch Prozeßketten entstehen, indem in der modellierten Beziehung von einer Prozeßinstanz die nächste initiiert wird, und von dieser dann eine weitere. Da für die regelbasierte Prozeßbeziehung, d.h. Säule 2 im Modell zur Behandlung komplexer Prozeßbeziehungen, die Operation „Prozeß warte“ implizit umgesetzt wurde, gilt die Darstellung der expliziten Modellierung der Beziehung „Prozeß initiieren“ für die anwenderorientierte Prozeßbeziehung.

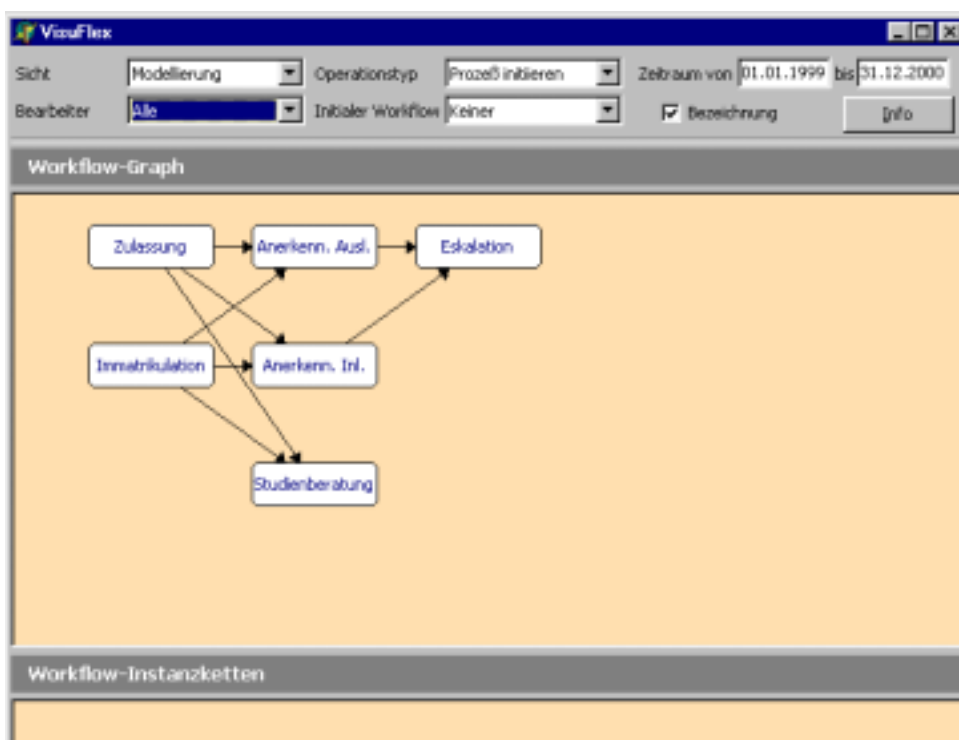


Abbildung 5-21: Prozeßnetz zu „VisuFlex“

Grundsätzlich ist die regelbasierte Nutzung der Operation „Prozeß initiieren“ aber möglich. Für das Prozeßpaar könnte dann bei der Beziehung unterschieden werden, ob diese vom Typ anwenderorientiert oder regelbasiert oder beides ist<sup>443</sup>.

Die im folgenden dargestellte Definition von PBZ orientiert sich am Beispiel der beiden Workflows „Antrag auf Rechnerkennung“ und „Interessentenregistrierung“<sup>444</sup>. Für die Definition der Beziehungen steht eine sogenannte Customizingtabelle zur Verfügung. Dort können die Prozesse und die Operationen festgelegt werden (vgl. Abb. 5-22). Damit wird ein Teil des exemplarischen Beziehungsaspekts modelliert. So wurde in der zweiten Zeile für das Workflow-Paar „WS91000068“ (Workflow zum Antrag auf Rechnerkennung) und „WS91000048“ (Workflow zur Registrierung von Interessenten) die Operation „Workflow Initi“ (Prozeß initiieren) zugelassen. Die Einträge werden wieder zur Laufzeit ausgelesen, so daß der Modellierer transparent und ohne Programmierkenntnisse die Beziehungen nur über Tabelleneinträge steuern kann.

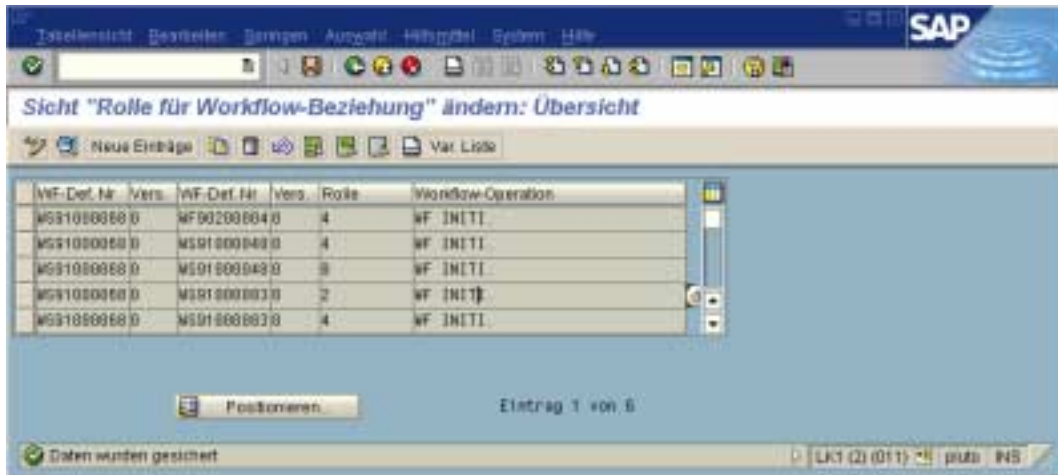
Die Definition des organisatorischen Bestandteils des Beziehungsaspekts, also die Festlegung der Rolle, kann ebenfalls über die Customizingtabelle erfolgen. Dabei kann eine Beziehung mit der Operation für verschiedene Rollen erlaubt werden. So ist im Beispiel in der zweiten Zeile die Rolle 4 (Workflow-Administrator) für die Beziehung zwischen Workflow „WS91000068“ und Workflow „WS91000048“ dargestellt. Für jeden Anwender werden seine Rollen separat festgelegt. Zur Laufzeit bekommt der Anwender dann nur die Prozeßpaare angezeigt, die für seine Rolle zulässig sind.

---

<sup>442</sup> Weitere Vorteile, wie die Auswertungsunterstützung einzelner und aggregierter PBZ im Hinblick auf die Prozeßverbesserung, sind Gegenstand von Abschnitt 5.2.3.

<sup>443</sup> Die Unterscheidung ist in VisuFlex nicht berücksichtigt.

<sup>444</sup> Die beiden Workflows wurden im Rahmen der Arbeit ebenfalls am Lehrstuhl entwickelt.



WF-Def. Nr.	Vers.	WF-Def. Nr.	Vers.	Rolle	Workflow-Operation
MS31000000	0	WF02000040	0	4	WF_INITI
MS31000000	0	MS01000040	0	4	WF_INITI
MS31000000	0	MS01000040	0	3	WF_INITI
MS31000000	0	MS01000020	0	2	WF_INITI
MS31000000	0	MS01000020	0	4	WF_INITI

**Abbildung 5-22: Customizingtabelle zur Festlegung der Prozeßbeziehung mit rollenorientierter Einschränkung der Sicht der Sachbearbeiter und zulässiger Operation auf die Prozeßpaare**

Die Definition des Datenflusses geschieht ebenfalls über eine Customizingtabelle. Dabei ist der Ausgangs- und der Zielprozeß sowie die Operation anzugeben (vgl. Abb. 5-23). In der ersten Zeile wird bei Anwendung der Operation „Workflow Initi.“ (Prozeß initiieren) das Element „Intermensch“ (Antragsteller) auf das Element „Interessent“ (Interessent im Workflow „Interessentenregistrierung“) abgebildet. Das für die Operationen zulässige Symbol \* bedeutet, daß die Beziehung für alle Operationen auf diesen Datenfluß aufsetzt.

An dieser Stelle sei auf die gemeinsame Basis der anwenderorientierten und der regelbasierten Prozeßbeziehung, d.h. von Säule drei und zwei des Modells zur Behandlung komplexer Prozeßbeziehungen, hingewiesen<sup>445</sup>. Während die Prozeßbeziehung, also der Verhaltensaspekt über Regeln abgebildet wird und damit nicht in der Customizingtabelle, bildet der Datenfluß eine gemeinsame Basis. Ein Datenfluß von zwei Prozessen, die durch Regeln in Beziehung stehen, kann ebenfalls über diese Einstellung gepflegt werden. Die Rollendefinition ist für die Regeldefinition ohne Bedeutung, da hier ebenfalls das System die Auswertung durchführt. Eine Rolleneinschränkung bzgl. des Anwenders ist dann ggf. über Regeln abzubilden.

<sup>445</sup> vgl. Abschnitt 2.2

WF-Def. Nr.	Vers.	Info	Element	WF-Def. Nr.	Vers.	Element	Workflow-Operation
MS91000068	0		INTERMENSCH	MS91000048	0	INTERESSENT	WF INIT1.
MS91000068	0		INTERMENSCH	MS91000083	0	INTERESSENT	WF INIT1.
MS91000068	0		D_FIRSTNAME	MS91000040	0	FIRSTNAME	WF INIT1.
MS91000068	0		D_FIRSTNAME	MS91000083	0	E_FIRSTNAME	WF INIT1.
MS91000068	0		D_LASTNAME	MS91000048	0	LASTNAME	WF INIT1.
MS91000068	0		D_LASTNAME	MS91000083	0	E_LASTNAME	WF INIT1.

**Abbildung 5-23: Customizingtabelle für Datenfluß, der nach Operationen unterschieden werden kann**

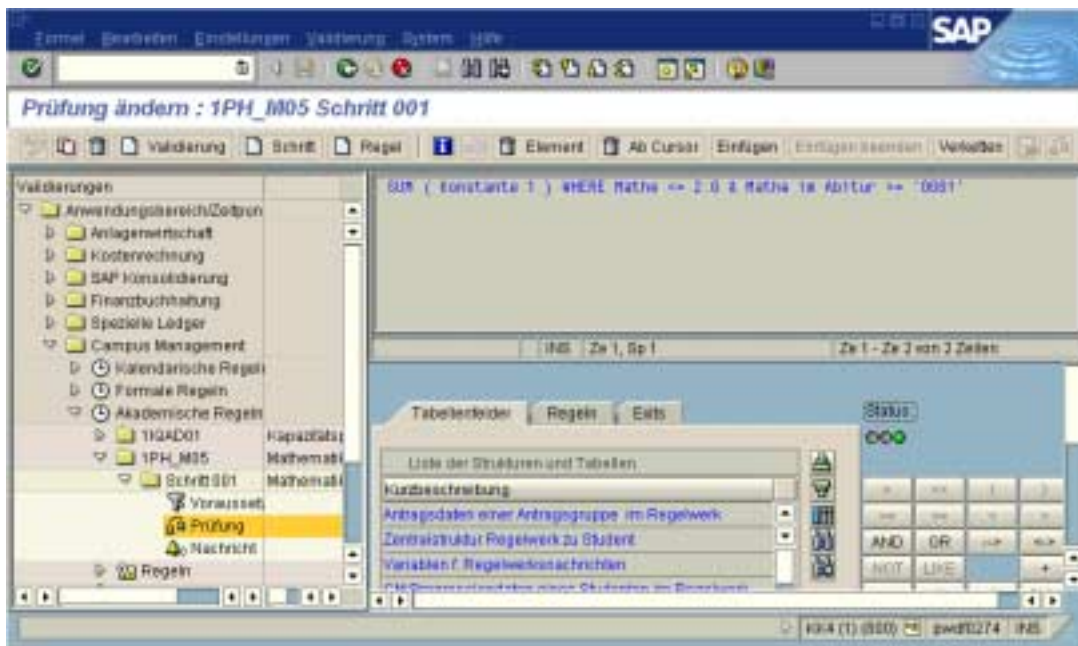
Die Realisierung für regelbasierte PBZ wird für den Bereich der Prozeßsynchronisation dargestellt. Diese wurde aus Gründen der Verdeutlichung des breiten Anwendungsspektrums vorgenommen. Eine Realisierung der Prozeßinitiation kann aber automatisiert unter Einbeziehung der Regelbasis analog zur anwenderorientierten Gestaltung erfolgen.

Die Regeln werden daher zur Synchronisation antragsbasierter Prozesse definiert und in die Auswertung integriert. Die Definition der Regeln erfolgt mit dem Werkzeug „Validations Substitutions Rules“ in einem spezifischen Editor (vgl. Abb. 5-24)<sup>446</sup>. Die Beschreibung von Regeln geschieht in Form von Booleschen Ausdrücken<sup>447</sup>. Im linken Fenster sind alle Regeln, die im Rahmen der Anwendung „Campus Management“ definiert und in die Booleschen Klassen „Kalendarrische“, „Formale“ und „Akademische“ Regeln eingeteilt wurden, systematisiert. Im rechten oberen Fenster ist die Definition der eigentlichen Regeln zu sehen. Die dargestellte Regel besagt, daß der Student über eine Qualifikation in Mathematik mit einer Note von 2.0 oder besser verfügt und diese Bestandteil der Abiturprüfung war. Die Formulierung der Regel besteht im Beispiel aus einem Sum-

<sup>446</sup> Der Regeleditor ist Bestandteil der Anwendung VSR des SAP R/3-Systems.

<sup>447</sup> vgl. Abschnitt 4.2.2

menoperator<sup>448</sup>, der alle Leistungen die der Bedingung „Mathe  $\leq 2.0$  & Mathe im Abitur“ genügen, zählt. Ist die Anzahl der gefunden Leistungen größer bzw. gleich  $\geq 1$ , ist die Prüfung erfolgreich. Die Bedingung ist ein Boolescher Ausdruck, in dem die Qualifikation in Mathematik spezifiziert werden. Im rechten unteren Fenster sind die Variablen, die bei der Regeldefinition bzw. Bedingungsdefinition zur Verfügung stehen, aufgelistet. Die „Zentralstruktur Regelwerk zu Student“ enthält alle persönlichen Daten des Studenten, wie z.B. Geburtsdatum, Adresse oder Behinderungen.



**Abbildung 5-24: Regeleditor des Werkzeugs „Validations Substitutions Rules“**

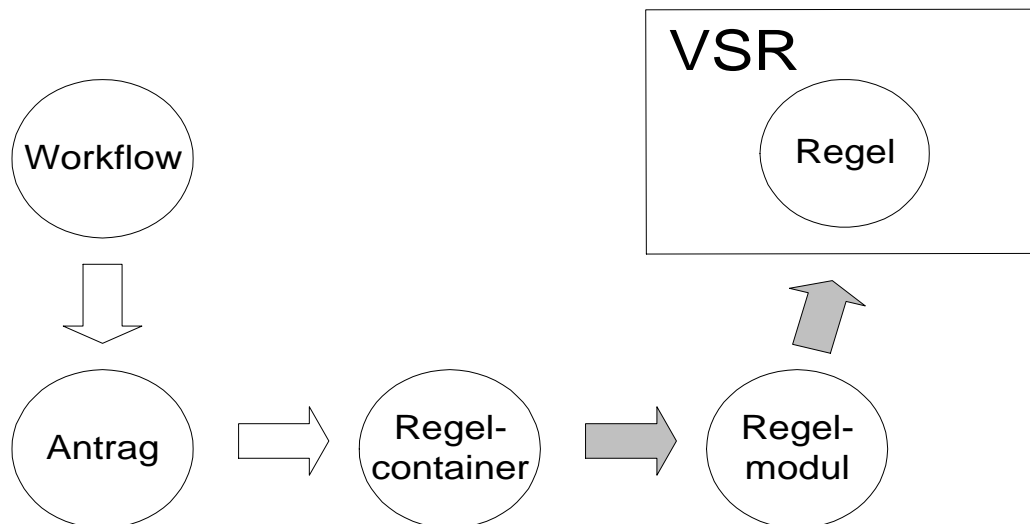
Aus Performanzgründen, d.h. um nicht alle Regeln zu prüfen, bzw. aus Vereinfachungsgründen, um die Definition der Regeln zu verschlanken, werden die Regeln zu Objekten<sup>449</sup>, den „Regelcontainern“<sup>450</sup>, zusammengefaßt. Diese „Regelcontai-

<sup>448</sup> Der Summenoperator „SUM“ ist ein Sprachelement des VSR.

<sup>449</sup> Dieser Objektbegriff bezeichnet Objekte der „Personalplanung und- entwicklung“ des SAP R/3-Systems.

<sup>450</sup> „Regelcontainer“ sind Objekte von „Campus Management“.

ner“ sind aus „Regelmodulen“<sup>451</sup> aufgebaut und können mit anderen Objekten verknüpft werden<sup>452</sup>. Die Verwendung dieser Objekte integriert damit die Regelprüfung in Workflows (vgl. Abb. 5-25). Verwendet also ein Prozeß einen Antrag, so können bei der Prüfung des Antrags die Regeln ausgewertet werden. Dieser Mechanismus wird auch zur Prozeßsynchronisation genutzt und im „Campus Management“ integriert.



**Abbildung 5-25: Einbindung der „Regelcontainer“ über Objekt „Antrag“ in den Workflow**

Zwischen antragsbasierten Prozessen ist also eine Synchronisation möglich. Die zugrundeliegende Datenstruktur, die die Daten zur Formalisierung der Regeln liefert, ist der Abb. 5-26 zu entnehmen. Die Antragsarten können zu Antragsgruppen zusammengefaßt werden. Innerhalb einer Gruppe können dann Abhängigkeiten zwischen den Status der Anträge modelliert werden. Da sich die Antragsarten in den Antragsgruppen beliebig zusammensetzen, d.h. eine unterschiedliche Zusammensetzung je nach Aufbau der Antragsgruppe vorliegt, ist die Definition der Elemente der Datenstruktur und deren Füllen „generisch“ konzipiert. Wurde eine

<sup>451</sup> „Regelmodule“ beinhalten die eigentlichen Regeln des VSR und dienen der Erhöhung der Wiederverwendbarkeit durch Modularisierung von Regeln.

<sup>452</sup> Den Objektverknüpfungen liegt die Technik der „Personalplanung und –entwicklung“ des SAP R/3-Systems zugrunde.

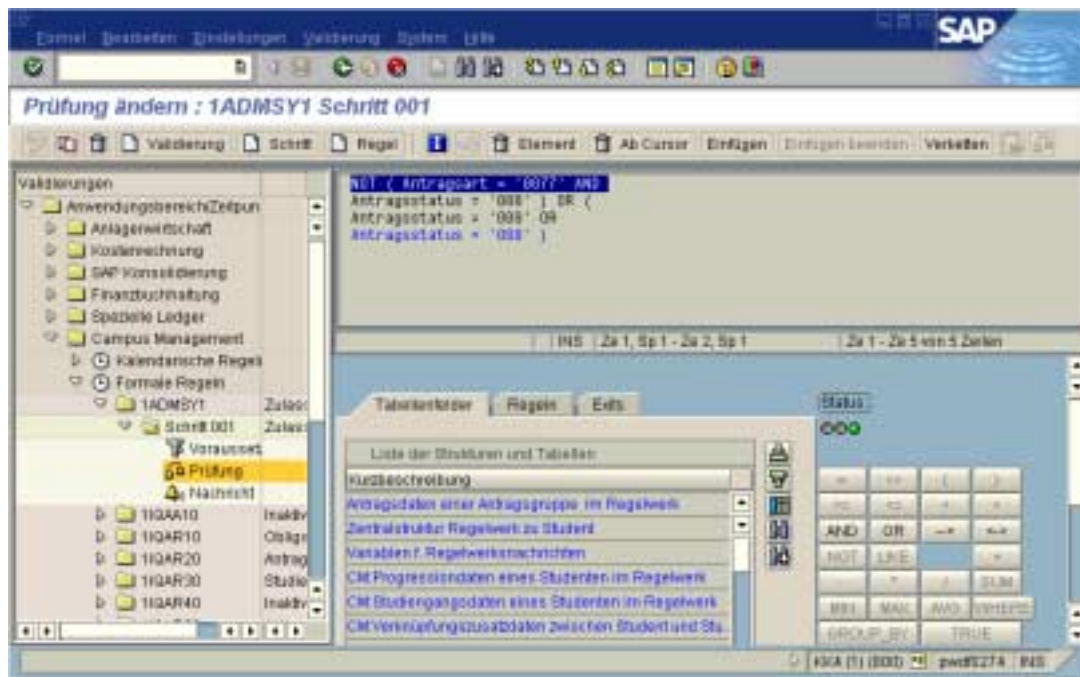
Antragsart als Hauptantrag in einer Gruppe spezifiziert, erschienen deren Daten in den Feldern H\_AN\_\*. Die Daten der aktuellen Antragsart werden in die Felder A\_AN\_\* eingetragen, ebenso wie die Daten der anderen Antragsarten der Gruppe und zwar in der Reihenfolge, in der die Antragsarten der Antragsgruppen zugeordnet wurden. Daher stehen in den Feldern N1\_AN\_\*, bis N10\_AN\_\* die restlichen Daten zu den anderen Antragsarten.

<b>Element der Datenstruktur</b>	<b>Bedeutung</b>
H_AN_APPTYPE	Art des Hauptantrags der aktuellen Antragsgruppe (z.B. „Zulassung“ in der Gruppe „Zulassung“)
H_AN_APPSTATUS	Status des Hauptantrags
A_AN_APPTYPE	Art des aktuellen Antrags (z.B. „Stipendium“)
A_AN_APPSTATUS	Status des aktuellen Antrags
N1_AN_APPSTATUS	Status des ersten anderen Antrags aus derselben Antragsgruppe (z.B. „Studienwunsch“ in „Zulassung“)
N2_AN_APPSTATUS	Status des zweiten anderen Antrags aus derselben Antragsgruppe (z.B. 2. „Studienwunsch“ in „Zulassung“)

**Abbildung 5-26: Ausschnitt der Datenstruktur „PIQRULEAPPSYNC“ zur Workflow-Synchronisation auf der Basis des für den jeweiligen Prozeß bestimmenden Antrags**

Beispiele für eine Regeldefinition auf der Basis der Datenstruktur sind für den Prozeß „Zulassung“ dargestellt (vgl. Abb. 5-27). Die Regel („Validierung“) besagt, daß der aktuelle Antrag der Art Stipendium (ID:77) nur in den Status 8 (genehmigt) gesetzt werden kann, wenn entweder der Antrag zum ersten Studienwunsch im Status 8 oder der Antrag zum zweiten Studienwunsch im Status 8 ist.

Damit ist also eine regelbasierte Prozeßsynchronisation auf der Basis prozeßtragender Antragsobjekte realisiert.



**Abbildung 5-27: Exemplarische Regel der Workflow-Synchronisation auf der Basis des für den jeweiligen Prozeß bestimmenden Antrags**

Eine alternative, direkt auf Prozeßattributen aufsetzende Lösung ist im folgenden dargestellt. Der Prozeß hat wichtige Attribute, auf die in der Regel Bezug genommen wird. Das Ergebnis ist Teil der Substitution und enthält dann den ausgewählten Prozeß sowie die durchzuführende Operation. Im Bedingungsteil der Regel zu Prozeß A und B wird formuliert, wann auf Prozeß B die Operation „Prozeß warte“ durchgeführt werden soll. Als Beispiel könnte dann folgende Regeln definiert werden:

Beispiel 1: Studienwunsch Wahl2 nur genehmigen, wenn Wahl1 abgelehnt

Regel: **Wenn** ( aktueller Prozeß = Studienwunsch AND Attribut Wunschnummer = Wahl 2 ) AND (Prozeß = Studienwunsch AND Attribut Wunschnummer = Wahl 1 AND Attribut Status <>„abgelehnt“ )

**dann** Prozeß warte.

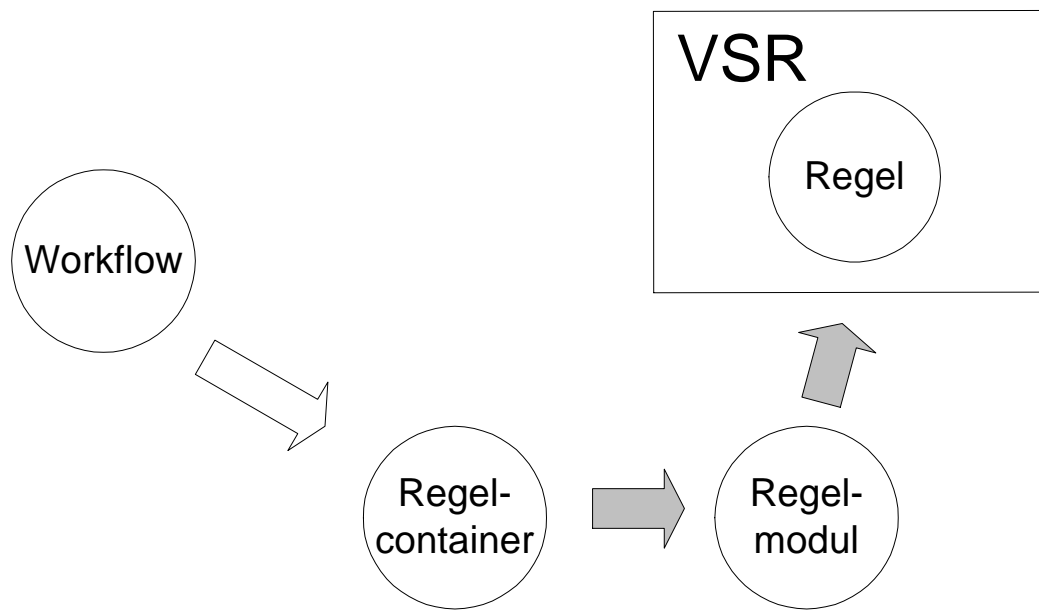
Beispiel 2: Finanzbeihilfe, nur wenn ein Studienwunsch genehmigt

Regel: **Wenn** ( Prozeß = Studienwunsch AND Attribut Wunschnummer = Wahl 1 AND Attribut Status <>„ genehmigt“ ) AND ( Prozeß = Studienwunsch AND Attribut Wunschnummer = Wahl 2 AND Attribut Status <>„ genehmigt“ )

**dann** Prozeß warte.



Die Auswertung dieser Alternative geschieht dann vor und nach Ausführung der Aufgabe. Die Auswertung der Regeln erfordert allerdings das Durcharbeiten aller Regeln. Auch hier ermöglicht das Zusammenfassen der Regeln zu Regelcontainern und deren Verbindung mit dem Prozeßobjekt (vgl. Abb. 5-28) bzw. der Workflow-Beziehung einen Performanzvorteil und eine Vereinfachung.



**Abbildung 5-28: Direkte Einbindung der „Regelcontainer“ in den Workflow**

### 5.2.3 Durchführung und Auswertung der Prozeßbeziehungen

Auch die Auswertung kann ausgehend vom Werkzeug „VisuFlex“ erfolgen. Die Darstellung der aggregierten Laufzeitdaten ist Abb. 5-30 zu sehen. Der obere Teil des Fensters enthält dabei die aggregierten PBZ für die Operation „Prozeß initiieren“. Die Verbindungslinien sind umso dicker, je häufiger die Operation auf das Paar angewandt wurde. Rote Pfeile bedeuten, daß eine Verbindung stattgefunden hat, schwarze, daß die Verbindung ausschließlich modelliert wurde. Prozeßfolgen der Einzelprozeßinstanzen sind im unteren Teilfenster dargestellt.

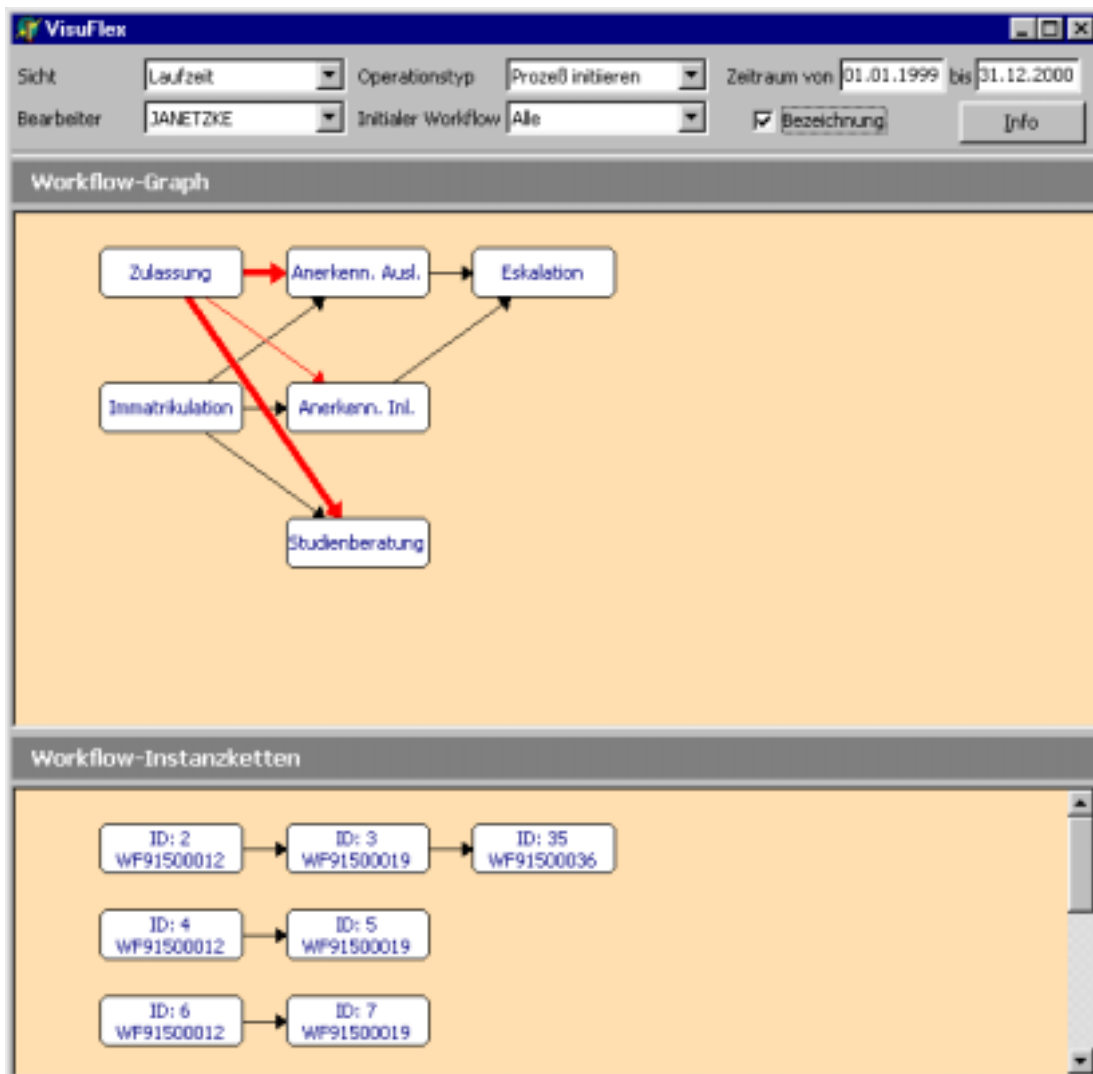


Abbildung 5-29: Darstellung Laufzeitdaten in „VisuFlex“

Entsprechend der Auswertungsziele können nun häufig genutzte Verbindungen analysiert werden. Unter Umständen kann hier automatisiert oder eine regelbasierte Lösung gefunden werden, die effizienter ist als die anwenderorientierte. Auch ein Problem in der Prozeßmodellierung könnte vorliegen und analysiert werden.

Wenn Verbindungen überhaupt nicht genutzt werden, stellt sich die Frage, ob diese redundant sind oder Fehler in der Prozeßbearbeitung vorliegen. Die Prozeßbeteiligten dieser Prozesse könnten auch zu wenig informiert worden sein.

Neben der generellen Prozeßauswertung können auch die PBZ personenspezifisch ausgewertet werden. Wer nutzt wie stark die neuen Möglichkeiten? Auch hier können Verbesserungspotentiale ermittelt werden.

Weiterhin besteht die Prozeßanalysemöglichkeit ausgehend von einem Prozeß. Von diesem können alle verbundenen Prozesse analysiert werden. Auch so können Ungleichgewichte erkannt werden. Dies ist insbesondere in Verbindung mit der Gesamtsicht zu sehen. So können die Probleme auch nur dann auftreten, wenn vorher ein anderer Prozeß zur Initiierung des aktuellen geführt hat.

Bevor die Auswertung stattfinden soll, wird ein Ablauf der Durchführung durch den Anwender gezeigt, d.h. aus Benutzerübersicht wird in der Prozeßschrittbearbeitung die Anwendung der Operation erforderlich. Als Beispiel sei hier der Workflow „Antrag auf Rechnerkennung“ gewählt. Aus dem Portal des Lehrstuhls können Studenten Anträge auf Rechnerkennung stellen, die dann von einem Lehrstuhlmitarbeiter bearbeitet werden. Der Anwender kann in der integrierten Workflow-Workbench, in der u.a. die Workflow-Operation „Start Workflow beliebig“ („Prozeß initiieren“) angeboten wird, diesen Menüpunkt selektieren (vgl. Abb. 5-30)

Starten einer Aufgabe

Workflow/Workbench System Hilfe

Start WF beliebig  
Start WF BasisObjekt  
Historie Start allg  
Historie Start User  
Akt. WF verbessern

Antragsnummer: 12

Nachname: Juraschek Matrikelnummer: 25689

Vorname: Patricia Telefonnummer: 0851/2513

Email: jurasc@10f5rzt1.rz.uni-passau.de

Antragsgrund: 2

Ersatzgrund Antrag: (empty)

R/3-System: 1

Antragsstatus: ABGELEHNT Sachbearbeiter: JAHETZKE

Datum Antragsstellen: 23.11.1998 Datum Antragsbeabtg: 06.01.2000

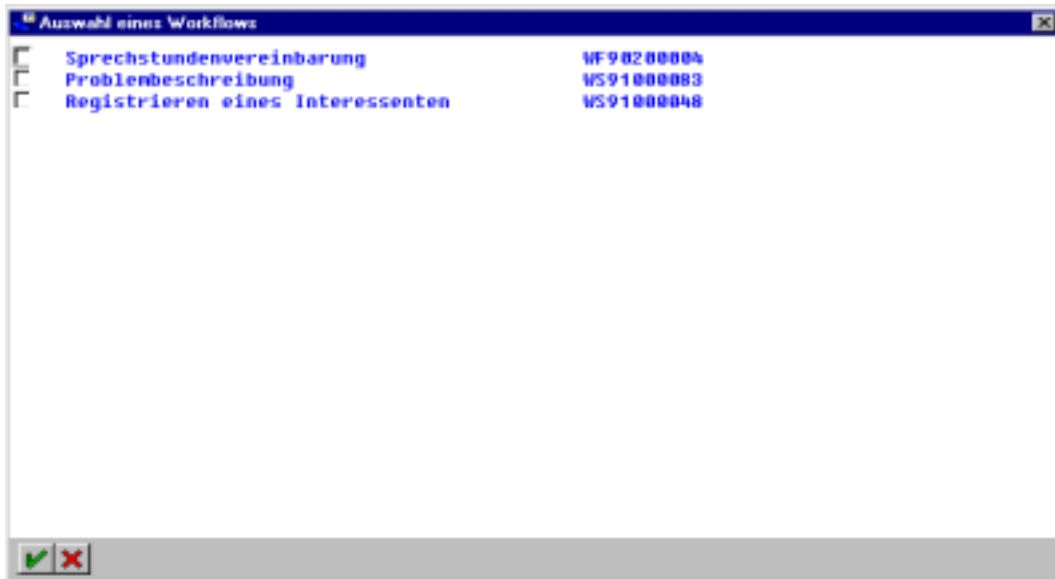
Kommentar: Test

Sichern Neuer Status Zurücklegen

LX1 (1) (011) pako INS 0.11

**Abbildung 5-30: Prozeßinitiierung durch den Anwender aus Prozeß „Rechnerkennung beantragen“**

Es erscheint dann die Auswahl der möglichen Prozesse, also die, die nach der Rolle und Beziehungsdefinition auswählbar sind (vgl. Abb. 5-31)



**Abbildung 5-31: Auswahl des Prozesses, der initiiert werden soll**

Nach Auswahl des Prozesses wird dieser automatisch initiiert. Der Datenfluß zwischen den Workflow-Containern findet im Hintergrund durch das WFMS statt. Der Sachbearbeiter kann sich vor dem Start oder danach seine bisherigen Aktionen ansehen (vgl. Abb. 5-32). Dazu hat er den Menüpunkt „Historie Start User“ in der Workflow-Workbench ausgewählt. Das Verhalten der anderen Anwender kann nach Wahl des Menüpunktes „Historie start allg“ eingesehen werden.



**Abbildung 5-32: Übersicht der durchgeführten Operationen der aktuellen Anwender**

Die Auswertung erfolgt auf Basis der Protokolldaten. Die Protokolldaten werden zur Prozeßlaufzeit bei Durchführung der Operation in eine Protokolltabelle geschrieben. Diese werden in vernetzten Strukturen nach Prozeßketten sortiert und aggregiert. Die aggregierten Prozeßketten werden dann in das statische Prozeßnetz eingetragen und können nach Startprozeß bzw. Bearbeiter gefiltert werden.

Die Auswertung sollte neben denen der Einzelprozesse stehen. So bietet z.B. das WFMS des SAP R/3-Systems die Möglichkeit, Einzelprozeßdaten in ein Informationssystem<sup>453</sup> einzuspielen und verschiedene statistische Größen abzuleiten.

Die Auswertung bezog sich auf Säule drei der PBZ. So könnten für verschiedene Rollen und Operationsinitiierung die Laufzeitdaten visualisiert werden. Ähnlich zur vorigen Aussage der regelbasierten Prozeßinitialisierung ist auch hier eine Protokollierung der regelbasierten Beziehung möglich. Im Rahmen der Realisierung der Prozeßsynchronisation antragsbasierter Prozesse stand die Auswertung nicht im Vordergrund.

#### 5.2.4 Integration der Modellimplementierung in die Anwendungen

In der Integration der Modellimplementierung in die Anwendungen soll nun die Integration der Workflow-Workbench von Seiten allgemeiner Anwendungsentwickler gezeigt werden. Weiterhin wird die Entwicklung des Laufzeitsystems zur Durchführung der Operation „Initiierung“ und schließlich die „VisuFlex“-konstituierende Algorithmik beschrieben.

Die Integration der Workflow-Workbench folgt einem generischen Ansatz. So wird eine fertige Menüleiste konzipiert, die in die eigene Anwendung von Seiten der Entwickler zu integrieren ist (vgl. Abb. 5-33).

---

<sup>453</sup> vgl. z.B. das Logistik Information System (LIS) des SAP R/3-Systems oder das Business Warehouse der SAP AG in SAP (1998c), S.8-4

andere Menüeinträge	Workflow	andere Menüeinträge
	Wizard 1	
	Wizard 2	
	Wizard 3	
	Wizard 4	
	Wizard 5	

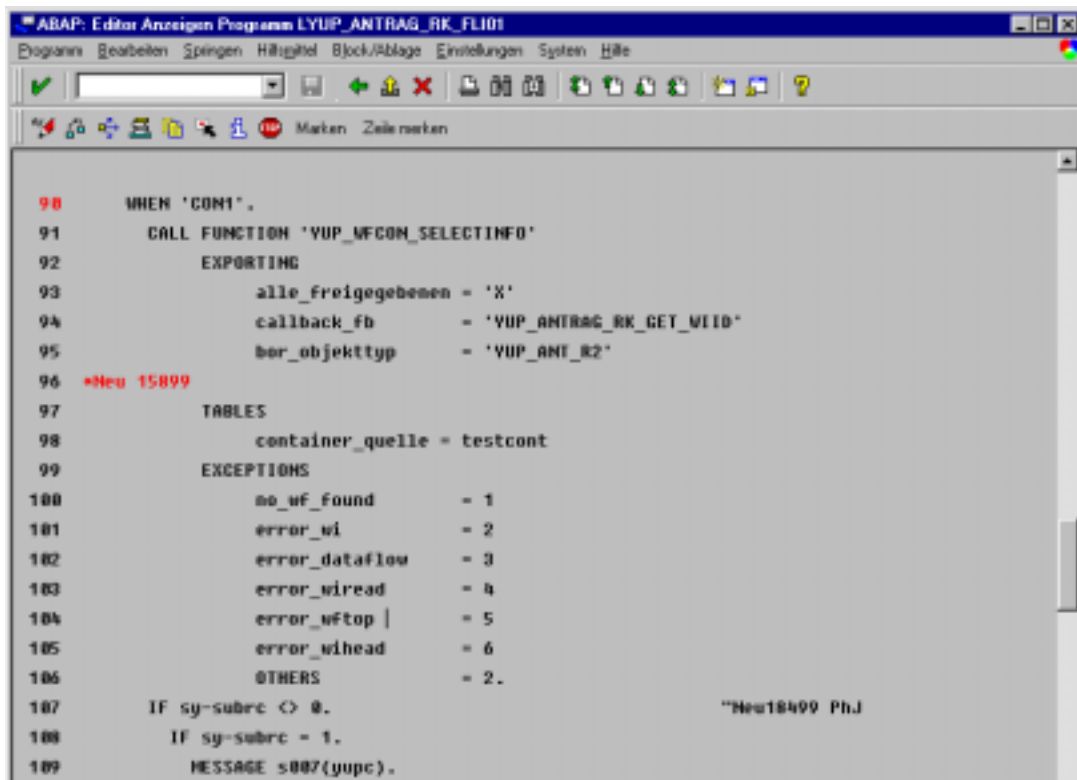
**Abbildung 5-33: Workflow-Workbench Menüleiste**

Für die Verarbeitung der einzelnen Einträge steht ein Katalog von Workflow-Workbench-Handlern<sup>454</sup> zur Verfügung. Den Handler zur Prozeßinitiiierung muß der aktuelle Prozeß sowie der Datencontainer des Ausgangsprozesses übergeben werden. Der Handler für die Workflow-Operation der Prozeßinitiiierung ist in ABAP programmiert und in Abb. 5-34 zu sehen.

Der Handler hat folgende Funktionen:

- Aufbau der Liste der Prozesse, die gemäß Rolle und Definition initiiert werden können
- Informationsbeschaffung über den ausgewählten Ziel-Workflow
- Durchführung des Datenflusses zwischen dem gewählten Prozeßpaar
- Aufruf Ziel-Workflow sowie
- Schreiben der Protokolldaten

<sup>454</sup> Als Handler werden Bausteine verstanden, die als Wizards spezifische Dienste anbieten. Wizards sind Werkzeuge zur Unterstützung der Benutzer.



```

90      WHEN 'CON1'.
91          CALL FUNCTION 'YUP_WFCOM_SELECTINFO'
92              EXPORTING
93                  alle_freigegebenen = 'X'
94                  callback_fb       = 'YUP_ANTRAG_RK_GET_WIID'
95                  ber_objektyp      = 'YUP_ANT_R2'
96      *Neu 15899
97          TABLES
98              container_quelle = testcont
99          EXCEPTIONS
100             no_wf_found      = 1
101             error_wi         = 2
102             error_dataflow   = 3
103             error_wiread     = 4
104             error_wftop      = 5
105             error_wihead     = 6
106             OTHERS          = 2.
107             IF sy-subrc <> 0.
108                 IF sy-subrc = 1.
109                     MESSAGE s007(yupc).

```

Abbildung 5-34: Workflow-Workbench Menühandler

Der Aufruf des Menühandlers erwartet also die folgenden Parameter, die kurz erläutert werden. Die Implementierung des Menühandlers ist in ABAP in Form eines Funktionsbausteins<sup>455</sup> geschehen. An Funktionsbausteine, die Prozeduren bei prozeduralen Programmiersprachen entsprechen und auch der Strukturierung und Modularisierung von Programmen dienen, werden verschiedene Parameter übergeben. Einfache Parameter werden über EXPORTING übergeben, komplexe, d.h. tabellarische, über TABLES. Als Ergebnis kann ein Funktionsbaustein Ausnahmen auslösen, auf die der Aufrufer des Funktionsbausteins reagieren kann. Auch die Rückgabe, hier nicht zu sehen, von Ergebnissen ist über IMPORTING-Parameter möglich. Die einzelnen Parameter sollen nun kurz erläutert werden.

### Parameter callback\_fb

Jeder Prozeß erhält bei der Instanziierung durch das WFMS eine individuelle ID. Diese ID ist in einem zweistufigen Verfahren mitzugeben. Zunächst wird ein weiterer Funktionsbaustein als Parameter mitgegeben. Dieser wird im aufgerufe-

<sup>455</sup> Funktionsbausteine sind Sprachelemente der Programmiersprache ABAP der SAP AG (vgl. Z.B. Kretschmer, R./Weiss, W. (1996), S.204-230



nen Funktionsbaustein dann selbst aufgerufen und liest aus dem Quellprozeß die ID. Damit soll sichergestellt werden, daß auch bei gleichzeitigem Aufruf die richtigen Prozeßinstanzen zugeordnet werden.

### **Parameter alle\_freigegeben**

Mit diesem Parameter wird festgelegt, ob alle – mit Einschränkung auf die für die Rollen des Aufrufers zulässigen - Prozeßpaare ausgewählt werden können. Mit diesem Parameter könnte das Rollenkonstrukt des Beziehungsaspekts weiter eingeschränkt werden. In der Darstellung wird aber das Standardverfahren demonstriert.

### **Parameter BOR-Objekt**

Da dieser Funktionsbaustein intern wiederum Funktionsbausteine des WFMS des SAP R/3-Systems verwendet, die das dem aktuellen Prozeßschritt zugrundeliegende BOR-Objekt erwarten, wird dieses bereits an obiger Stelle übergeben und letztlich über mehrere Aufrufe von Funktionsbausteinen weitergereicht.

### **Tabelle: tables Parameter container\_quelle**

Der Container enthält die Daten des aktuellen Prozesses. Aus diesem Container werden die Daten, die per Datenfluß in der Customizingtabelle festgelegt wurden, den Daten in dem Container des Ziel-Workflows zugeordnet.

Zur Durchführung der einzelnen Schritte stehen wiederum gekapselte Funktionsbausteine zur Verfügung, wie z.B. ein Anzeigebaustein von Prozeßlisten, ein Datenbeschaffer von Workflow-Daten aus Prozeßinstanzen, ein Startbaustein für Workflows oder ein Protokollbaustein. Eine Erläuterung der Funktionsbausteine im Detail erscheint allerdings nicht erforderlich.

Die Implementierung des Regelwerks besteht aus einer Bereitstellung und dem Füllen verschiedener Datenstrukturen, der Verwaltung der Regelcontainer, der Regelmodule und der Regelemente sowie schließlich der Entwicklung von Regelmanagern, die die Regelprüfung durchführen.

Im universitären Bereich findet eine Vielzahl von Prüfungen statt. Damit sind im folgenden allerdings Prüfungen gemeint, die im Rahmen von Prozessen wie Zu-

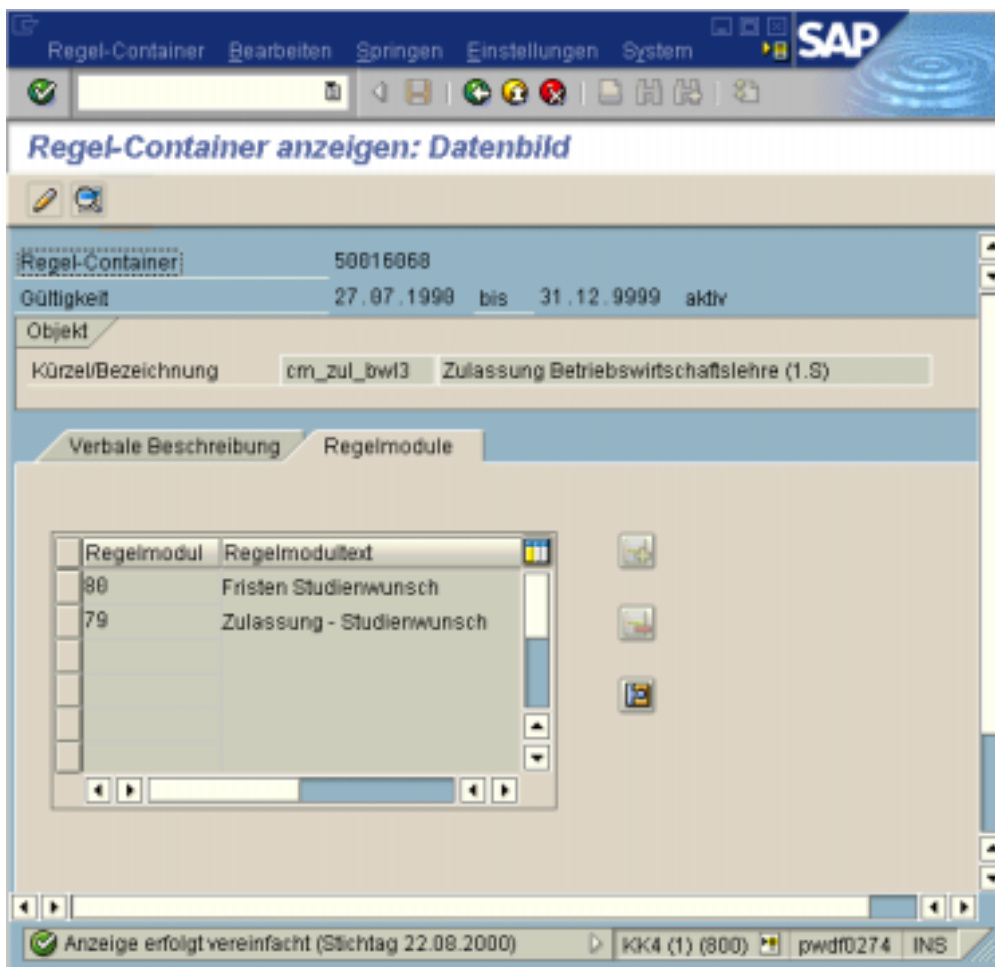
lassung, Prüfungsanmeldung, Studienfortschritt oder Graduierung stattfinden. Die meist in Form von schriftlichen oder mündlichen Prüfungen nachgewiesenen Fähigkeiten werden dann in Rahmen von akademischen Voraussetzungsprüfungen bei der Zulassung, bei der Prüfungsanmeldung oder bei der Studienfortschrittsprüfung kontrolliert. Diese in Regeln formulierbaren Voraussetzungsprüfungen wurden ebenfalls in das Regelwerk eingebettet. Daher bildet das Regelwerk eine zentrale Anwendung innerhalb des „Campus Managements“. Die folgende Darstellung zum Rahmenwerk, das eine strukturierte Verwaltung der Regeln des Regelwerks ermöglichen soll, soll daher ein Beispiel zu Voraussetzungsprüfungen im Rahmen der Zulassung aufzeigen, das Regeln zu vorausgesetzten Qualifikationen, aber auch Regeln zu definierten Fristen beinhaltet. Die Möglichkeit der Integration der Synchronisationsregel wird abschließend verdeutlicht.

Das Zusammenfassen der Regeln geschieht über eine Customizingtabelle (vgl. Abb. 5-35). Dabei werden zur besseren Modularisierung „Regelmodule“ mit „Regelementen“ gefüllt. Die „Regelemente“ entsprechen dabei den eigentlichen Regeln, die mit dem Werkzeug „Validations Substitutions Rules“ angelegt wurden. So werden zwei „Regelmodulen“ „Zulassung Studienwunsch“ sowie „Fristen Studienwunsch“ definiert und diesen Validierungsregeln aus dem VSR zugeordnet. Über eine Reihenfolge in der Sortierung kann die Reihenfolge der Prüfungen festgelegt werden.

Regelmodul	Kla..	Regelart	Regele..	Regelementtext	Kla..	Komb..	Sortierfolge
Zulassung - Stud..	1	Validierung	1PH_M05	Mathematikkenntnisse Zulassung	and	10	
Zulassung - Stud..	1	Validierung	1PH_A02	Naturwissensch.-kennt. Zulassung	and	20	
Zulassung - Stud..	(	Validierung	1PH_A03	Englischkenntnisse Zulassung	or	30	
Zulassung - Stud..	)	Validierung	1PH_A04	Englischkenntnisse TOEFL	)	40	
Fristen Studienw.	1	Validierung	1PH_K01	Frist Zulassung		10	

Abbildung 5-35: Zuordnung von „Regelementen“ zu „Regelmodulen“

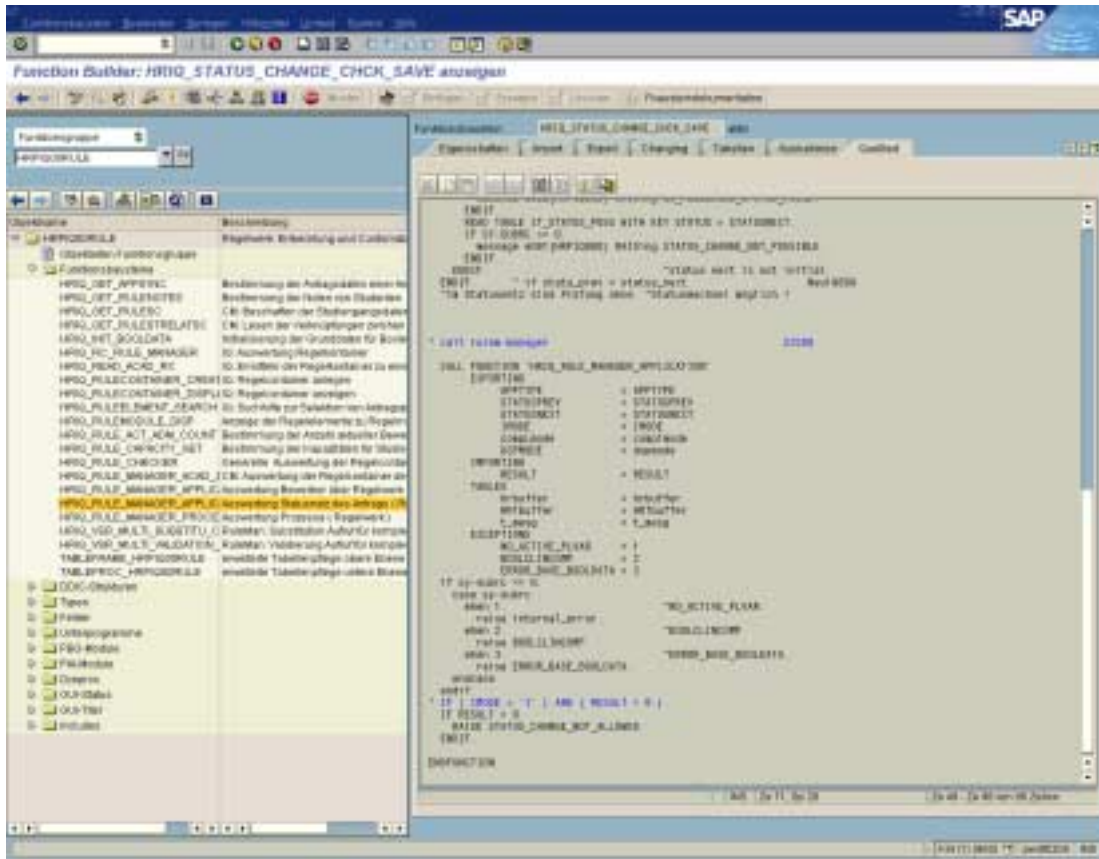
Die „Regelmodule“ können wiederum „Regelcontainer“n zugeordnet werden. Die Zuordnung ist in folgender Abb. 5-36 zu sehen. Dem „Regelcontainer“ „Zulassung Betriebswirtschaftslehre (1.S)“ mit der ID 50016068 werden nun die „Regelmodule“ „Fristen Studienwunsch“ und „Zulassung - Studienwunsch“ zugeordnet. Das mehrstufige, objektorientierte Konzept wurde bereits in Abb. 5-25 gezeigt.



**Abbildung 5-36: Zuordnung von Regelmodulen zu einem Regelcontainer**

Schließlich können die „Regelcontainer“ Antragsarten zugeordnet werden. Die Auswertung der Regeln geschieht über Regelmanagerbausteine. Für die Antragsverwaltung gibt es einen eigenen Baustein, der vorgestellt werden soll. Der Aufruf

in der Anwendung „Antragsverwaltung“<sup>456</sup> ist in folgender Abb. 5-37 zu sehen. Der Regelmanager ist ebenfalls in ABAP programmiert.



**Abbildung 5-37: Aufruf des Regelmanagers aus der Anwendung „Antragsverwaltung“**

Der Regelmanager wird in der Anwendung „Antragsverwaltung“ aufgerufen (vgl. Abb. 5-5). Der Antrag kann im neuen Status („Feld: Neuer Status“) erst gesichert werden, wenn der Sachbearbeiter durch die Drucktaste „Antrag prüfen“ den Regelmanager zur automatischen Prüfung involvierte und das Prüfergebnis erfolgreich war. An den Regelmanager, ebenfalls ein Funktionsbaustein, müssen in der Anwendung „Antragsverwaltung“ verschiedene Parameter übergeben werden. Die wichtigsten sind dabei folgende:

<sup>456</sup> Die Anwendung ist Bestandteil von „Campus Management“ der SAP AG.

**Parameter APPTYPE (Antragsart)**

Mit der Antragsart wird bekanntgegeben, um welche Antragsart es sich handelt, z.B. Studienwunsch, Immatrikulation, Rückmeldung oder Prüfungsanmeldung. Anhand der Antragsart können die richtigen Regelcontainer ermittelt werden. Schließlich sind andere Regeln für eine Immatrikulation als für eine Prüfungsanmeldung erforderlich.

**Parameter STATUSPREV bzw. STATUSNEXT (Status Antrag vorher bzw. nachher)**

Die Anträge werden im Rahmen ihrer Bearbeitung in verschiedene Status gesetzt. Dabei sind für eine Antragsart nur bestimmte Status bzw. Statusübergänge erlaubt. Anhand eines zugrundeliegenden Statusnetzes wurden bei der Definition einer Antragsart bereits mögliche Statusübergänge vordefiniert. An diesen Statusübergängen können dann die jeweiligen Regelcontainer hinterlegt werden. So ist beispielsweise beim Übergang eines Antrag von „erfaßt“ auf „vollständig“ nur die formale Vollständigkeit des Antrags zu prüfen. Also enthält dieser Regelcontainer Regelmodule mit Regelementen, die im wesentlichen formale Prüfungen beinhalten. Beim Übergang auf „genehmigt“ sind hingegen akademische Regeln (alle erforderlichen Qualifikationen und Prüfungen in ausreichender Qualität vorhanden?), aber auch Synchronisationsregeln zu überprüfen. Beispielsweise kann ein Studienwunschantrag Wahl 2 nur dann genehmigt werden, wenn der Studienwunsch Wahl 1 bereits abgelehnt wurde. Man sieht hier, daß die Regeln der Prozeßsynchronisation und die der akademischen Prüfungen im gleichen Framework<sup>457</sup> laufen und daher parallel eingesetzt werden können.

**Parameter IMODE bzw. COMPLMODE bzw. DSPMODE**

Diese Steuerungsparameter dienen zur Feinsteuerung der Nachrichtenausgabe der Regelprüfung. Ist eine Regel nicht erfüllt, können Fehler- oder Warnmeldungen ausgegeben werden. Mit IMODE werden Warnungen wie Fehlermeldungen behandelt. Mit COMPLMODE wird festgelegt, daß alle Meldungen in einer Liste gesammelt werden, d.h. also, daß nicht nach einem Fehler bereits die Regelprü-

---

<sup>457</sup> Regeln zur Prüfung der akademischen Voraussetzungen werden in „Campus Management“ ebenso im VSR formuliert wie die Regeln zur Synchronisation der Workflows.

fung abgeschlossen wird. Mit DSPMODE läßt sich festlegen, ob die Meldungen ausgegeben oder über den tables Parameter t\_mesg zurückgegeben werden.

### Tables Parameter HRBUFFER bzw. HRTBUFFER bzw. T\_MESG

Über die Parameter „hr\_buffer“ bzw. „hrtbuffer“ werden fast alle Daten des Antrags übergeben. Diese stehen in den jeweiligen Regeln als Attribute mit geringen Einschränkungen zur Verfügung. Soll keine Bildschirmausgabe der Fehlermeldungen erfolgen, können diese über den t\_mesg Parameter an den Aufrufer zurückgegeben werden.

### Parameter RESULT

Im Parameter RESULT wird bekanntgeben, ob es mindestens eine Regel geben hat, die nicht erfüllt werden konnte.

Das Ergebnis des Regelmanagers ist in Abb. 5-38 zu sehen. In diesem Fall ist ein Umsetzen des Antragsstatus auf den Status 8 („genehmigt“) nicht möglich, da der Antragsteller die notwendigen akademischen Voraussetzungen für den beabsichtigten Studiengang nicht erfüllt.

Regelname	Prüfung/Regelname/Beitrag	Prüfung/Mittel/Prüfung/Teil	Beschreibung
Fristen Studienwunsch	1PH_A01	100	Frist Zulassung
Zulassung - Studienwunsch	1PH_A01	100	Mittlermathematische Zulassung
Zulassung - Studienwunsch	1PH_A02	000	Ausreichende Leistungen in Mathematik - keine
Zulassung - Studienwunsch	1PH_A03	000	Ausreichende Leistungen in Naturwissenschaften
Zulassung - Studienwunsch	1PH_A04	000	Ausreichende Englischkenntnisse TOEFL

**Abbildung 5-38: Ausgabe des Ergebnisses der Prüfung durch das Regelwerk**

So ist der Antrag zwar fristgerecht eingegangen, das Regelmodul „Fristen Studienwunsch“ erfüllt. Die fehlenden mathematischen Vorkenntnisse führen aber,

da notwendige Voraussetzung (vgl. Regelmodul „Zulassung – Studienwunsch“), zur Ablehnung. Ein Ausgleich der Englischkenntnisse durch den TOEFL-Test wäre möglich gewesen.

In der aktuellen „Campus Management“ Realisation können neben den Regeln zur Terminprüfung oder akademischen Prüfung auch die Regeln z.B. zur Workflow-Synchronisation auf Antragsbasis geprüft werden. Diese Regeln zur Synchronisation können ebenfalls in Regelmodulen zusammengefaßt werden und diese wiederum in Regelcontainern. Die Fehlerausgabe aller Regelprüfungen erfolgt als Teil der gesamte Liste der Regelprüfungen. So hat der Anwender eine zentrale Ausgabe aller Regelresultate.

Es ist also eine weitreichende Regelprüfung an Universitäten möglich. Die Verwaltung von komplexen Regelwerken ist allerdings aufwendig und wird schnell unübersichtlich. Um daher eine strukturierte Übersicht über die Regelgesamtheit zu erhalten, wurde eine entsprechende Übersichtsdarstellung realisiert<sup>458</sup>. So kann der Verwalter der Regelwerke alle Regeln auf einen Blick verwalten (vgl. Abb. 5-39). Im linken Teilfenster wählt der Bearbeiter der Regelübersicht die Einstiegsobjekte aus. Zu sehen ist der Einstieg über Regelcontainer. Zu den Regelcontainern können dann alle zugehörigen Regelmodule und Regelelemente eingesehen werden. Diese Richtung wird als top down bezeichnet. Beginnt die Darstellung bei den Regelelementen, dann als bottom up bezeichnet, kann die Wiederverwendungshäufigkeit der einzelnen Regelelemente bzw. Regelmodule analysiert werden. Bei Änderungen einzelner Regeln ist durch diesen Verwendungsnachweis daher eine bessere Feststellung der Auswirkungen möglich.

---

<sup>458</sup> Die Verwaltung komplexer Regelwerke weist gegenüber herkömmlichen programmierten Lösungen deutliche Vorteile auf, z.B. müssen bei Änderungen von Prüfungsordnungen nur einzelne Regeln überarbeitet bzw. neue Regeln formuliert und nicht Programme geändert bzw. geschrieben werden.

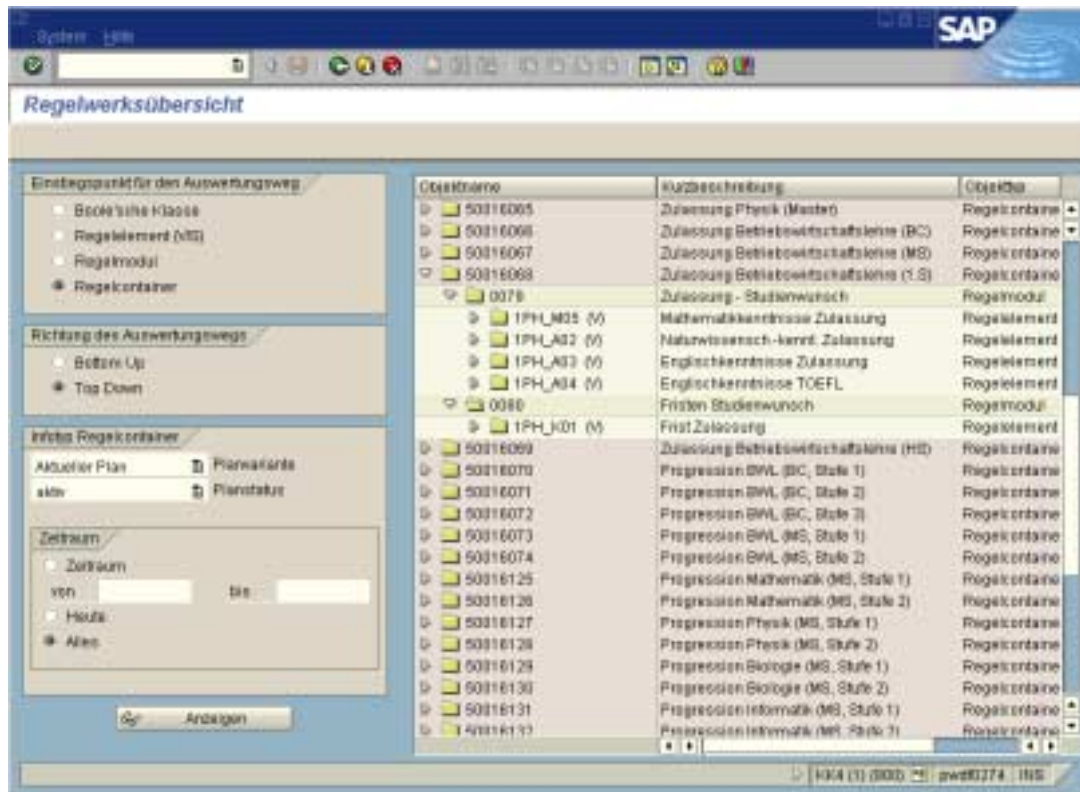


Abbildung 5-39: Regelwerksübersicht

Die Prozeßbeteiligten und Modellierer der Regeln haben im Falle der Regelpflege eine gute Kontrolle über Anweisungen bei Änderungen der Regeln. Dargestellt sind schließlich die Verwendungen der Regeln.

Bei der Beschreibung der Implementierung von „VisuFlex“ seien die Verwaltung der Prozeßfolgen und die Darstellung des Graphen umrissen.

Die Verwaltung der Prozeßketten, die aus den Protokolltabellen entwickelt wurden, geschieht über zweifach verkettete Listen. Auf der Grundlage der Protokolltabellen werden Prozeßfolgen ermittelt. Dies ist möglich, weil die ID der Prozeßinstanzen protokolliert werden. Jede Prozeßinstanz, von der eine Prozeßinstanz initiiert wurde, hat daher einen Nachfolger und umgekehrt einen Vorgänger. Daher wurde eine zweifach verkettete Liste unterstellt, wobei zwei Elemente dieser Liste jeweils zwei Workflow-Typen entsprechen und die - wieder als Liste - tabellarisch die Instanzinformationen beinhalten. Aufgrund der objektorientierten Implementierung entsprechen die Listenelemente wiederum Klassen.



Die Darstellung der aggregierten PBZ in Graphenlayout geschieht über den Sugiyamaalgorithmus<sup>459</sup> (vgl. Abb. 5-41).

Der Sugiyama-Algorithmus erzeugt den Graphen in zwei Schritten: Im ersten Schritt, dem sogenannten *Leveling*, wird die Knotenmenge in Schichten (Level) partitioniert. Ausgehend von einem initialen Knoten, der der ersten Schicht zugewiesen wird, werden die direkt mit ihm verbundenen Knoten der zweiten Schicht zugeordnet. Von jedem dieser Knoten der zweiten Schicht werden die mit ihnen inzidenten Knoten der jeweils nächsten Schicht zugeordnet.

Die Schichten werden horizontal von links nach rechts, die Knoten innerhalb der Schichten vertikal von oben nach unten angeordnet. Ziel ist es, den Graphen mit der minimalen Anzahl von Kantenüberschneidungen zu zeichnen. Um dies zu erreichen, wird ein zweiter Schritt, das sogenannte *Bubbling*, angewendet. Dabei wird innerhalb der Schichten die vertikale Anordnung der Knoten in der Art permutiert, daß möglichst wenig Überschneidungen auftreten.

#### **Abbildung 5-40: Prinzip des Sugiyama-Algorithmus**

Diese Prozedur wird für alle Schichten durchgeführt. In der Anwendung VisuFlex wird stets ein initialer Workflow-Knoten bestimmt und ausgehend von seinen Nachbarn wird die zweite Schicht aufgebaut. Dies erzeugt einen baumartigen Aufbau von links nach rechts, wobei sich der Wurzelknoten links befindet.

---

<sup>459</sup> vgl. Sugiyama, K./Tagawa, S./Toda, M. (1981)