

Zweitveröffentlichung



Sinz, Elmar J.

Anwendungssystem-Architektur der Universität

Datum der Zweitveröffentlichung: 22.10.2024

Verlagsversion (Version of Record), Beitrag in Sammelwerk

Persistenter Identifikator: urn:nbn:de:bvb:473-irb-1040013

Erstveröffentlichung

Sinz, Elmar J. (1998): Anwendungssystem-Architektur der Universität, in: Hans-Ulrich Küpper, Elmar J. Sinz (Hrsg.), Gestaltungskonzepte für Hochschulen : Effizienz, Effektivität, Evolution, Stuttgart: Schäffer-Poeschel, S. 58–69.

Rechtehinweis

Dieses Werk ist durch das Urheberrecht und/oder die Angabe einer Lizenz geschützt. Es steht Ihnen frei, dieses Werk auf jede Art und Weise zu nutzen, die durch die für Sie geltende Gesetzgebung zum Urheberrecht und/oder durch die Lizenz erlaubt ist. Für andere Verwendungszwecke müssen Sie die Erlaubnis der Rechteinhaberinnen und Rechteinhaber einholen.

Für dieses Dokument gilt eine Creative-Commons-Lizenz.



Die Lizenzinformationen sind online verfügbar:

<https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>

2 Anwendungssystem-Architektur der Universität

Elmar J. Sinz¹

2.1 Kartierung der Anwendungssystem-Architektur in den Geschäftsprozeßmodellen der Universität	59
2.2 Entwicklung prozeßorientierter Anwendungssysteme	63
Literatur	69

¹ Prof. Dr. Elmar J. Sinz, Universität Bamberg, Lehrstuhl für Wirtschaftsinformatik, insbes. Systementwicklung und Datenbankanwendung, Feldkirchenstraße 21, D-96045 Bamberg. Tel. ++49 951 863-2512, Fax ++49 951 863-2513, E-Mail: elmar.sinz@sowi.uni-bamberg.de, <http://www.seda.sowi.uni-bamberg.de>

Ebenso wie in der Wirtschaft ist auch an Universitäten eine effektive und effiziente Leistungserbringung ohne den bestmöglichen Einsatz computergestützter Anwendungssysteme - im folgenden kurz als Anwendungssysteme bezeichnet - nicht möglich. Im Gegensatz zur Wirtschaft, die das strategische Potential des Einsatzes von Informationstechnologie bereits frühzeitig erkannt hat, bestehen hier an Universitäten noch erhebliche Defizite.

Die typische Anwendungssystemlandschaft an Universitäten ist historisch gewachsen. Sie besteht aus einzelnen, vielfach nur unzureichend integrierten Anwendungssystemen. Die Palette verfügbarer Anwendungssysteme ist zudem begrenzt. Hochschulsoftware wurde bislang insbesondere nur von der HIS GmbH in Hannover, einer Einrichtung des Bundes und der Länder, angeboten. Erst in letzter Zeit interessieren sich auch andere Softwarehersteller für Universitäten als Nachfrager von Software.

Lehrstühle und Institute verwenden häufig andere Systemplattformen als die Hochschulverwaltung. Letztere ist zudem an zum Teil technisch überholte Richtlinien für Verwaltungsdatenverarbeitung gebunden, deren Anpassung an die technische Entwicklung durch lange Entscheidungsdauern der zuständigen Gremien behindert wird.

Darüber hinaus bestehen zum Teil noch erhebliche Lücken in der Versorgung mit Informationstechnologie. Diese existieren zumindest solange, als nicht jede Mitarbeiterin und jeder Mitarbeiter bei Bedarf in gleicher Weise über einen vernetzten Personal Computer verfügt wie über ein Telefon. Die Analogie zum Telefon verdeutlicht noch einen weiteren Aspekt. Aufgrund der Kostenentwicklung im Bereich der Informationstechnologie steht nicht die Auslastung von Geräten im Vordergrund, sondern die breite Verfügbarkeit. Erst diese eröffnet die volle Palette der Gestaltungspotentiale für teilautomatisierte Universitätsprozesse. Die Informationstechnologie wirkt hier als *enabling technology*.

Für die Gestaltung und Automatisierung von Universitätsprozessen ist auch die Verfügbarkeit von Informationstechnologie bei den Kunden der Universität von Interesse. Sind ausreichend Rechnerarbeitsplätze für Studierende verfügbar? Sind die Studierenden mit privaten Rechnern ausgestattet? Welche Dienste und Nutzungsformen des Internet sollen berücksichtigt werden? Wie soll die Kommunikation mit den Forschungspartnern gestaltet werden? Gestaltungsoptionen sind z.B. Forschungsberichte im WorldWideWeb und die Nutzung von Videoverbindungen (ProShare) für die Kommunikation zwischen entfernten Forschungspartnern.

2.1 Kartierung der Anwendungssystem-Architektur in den Geschäftsprozeßmodellen der Universität

Unter dem Begriff Anwendungssystem-Architektur werden der „Bauplan,, und die „Konstruktionsregeln,, verstanden, die den Anwendungssystemen eines betrieblichen Systems, in diesem Falle der Universität, zugrundeliegen. Der Bauplan umfaßt die Beschreibung der Komponenten und Beziehungen von Anwendungssystemen. Die Konstruktionsregeln für die Bildung von Komponenten und Beziehungen werden in Form von Metamodellen vorgegeben. Diese enthalten die verfügbaren Arten von Komponenten und Beziehungen sowie Konsistenzbedingungen für deren Verwendung².

Entsprechend dem in Teil I eingeführten Gestaltungsrahmen werden Anwendungssysteme als maschinelle Aufgabenträger für die Durchführung der automatisierten Aufgaben von Geschäftspro-

² SINZ Architektur betrieblicher Informationssysteme 1997.

zessen verstanden. Diese Sichtweise ermöglicht es, die Anwendungssysteme der Universität in direktem Bezug zu den Universitätsprozessen zu analysieren und zu gestalten³. Die vorhandenen oder geplanten Anwendungssysteme werden dabei in den Geschäftsprozeßmodellen „kartiert“,⁴.

Im Zusammenhang mit der Analyse und Gestaltung der Anwendungssystem-Architektur sind insbesondere folgende Fragen von Interesse:

- Welche Aufgaben und Transaktionen von Geschäftsprozessen werden durch Anwendungssysteme automatisiert?

Dabei wird zwischen *Automatisierbarkeit* und *Automatisierung* unterschieden. Eine Aufgabe ist automatisierbar, wenn sie formal spezifiziert und ein für die Durchführung der Spezifikation geeigneter maschineller Aufgabenträger angegeben werden kann. Anderenfalls ist die Aufgabe nicht automatisierbar und von einem personellen Aufgabenträger durchzuführen. Automatisierbare Aufgaben können dann bezüglich ihrer tatsächlichen Automatisierung nicht-, teil- oder vollautomatisiert sein.

Analog ist eine Transaktion automatisierbar, wenn sie formal spezifiziert und ein geeigneter elektronischer Übertragungskanal angegeben werden kann. Nicht automatisierbare Transaktionen werden z.B. papiergestützt durchgeführt. Automatisierbare Transaktionen können nicht- oder vollautomatisiert sein (zur Darstellung siehe die Legende in Abbildung 2-1).

Die Bestimmung der Automatisierbarkeit und die Festlegung der Automatisierung der Aufgaben und Transaktionen eines Geschäftsprozesses beschreibt dessen Automatisierungsgrad.

- Sind die Anwendungssysteme „prozeßgerecht“, gestaltet?

Im Hinblick auf eine flexible, evolutionäre Weiterentwicklung von Geschäftsprozessen und den sie unterstützenden Anwendungssystemen ist es notwendig, die Anwendungssystem-Architektur an der Struktur der Geschäftsprozesse auszurichten. Zum Beispiel sollte eine Anwendungssystemkomponente die Aufgaben von genau einem betrieblichen Objekt unterstützen, Funktionsüberlappungen von Anwendungssystemkomponenten sowie Automatisierungslücken sollten vermieden werden.

Auf diese Weise werden die Voraussetzungen dafür geschaffen, Anwendungssysteme schritthaltend mit den Geschäftsprozeßmodellen weiterentwickeln zu können⁵. Im Gegensatz zu großen, monolithischen Anwendungssystemen, die sich häufig der Prozeßentwicklung sperren, führt dieser Ansatz zu kleineren, kooperierenden Anwendungssystemkomponenten. Ohne Kartierung in den Geschäftsprozeßmodellen ist diese Form von Anwendungssystemen nicht beherrschbar.

- Welche Integrationsformen werden für die Verknüpfung der einzelnen Anwendungssysteme verwendet?

Integration von Anwendungssystemen bedeutet das Zusammenführen von Teil-Anwendungssystemen zu einem im Ganzen wirksamen Anwendungssystem. Ausgangspunkt der Integration können isolierte Teil-Anwendungssysteme oder ein (postuliertes) Gesamt-Anwendungssystem sein, welches in Teil-Anwendungssysteme strukturiert werden soll. Ziele

³ KRUMBIEGEL *Integrale Gestaltung* 1997.

⁴ KRUMBIEGEL *Anwendungssystem-Kartierung* 1996.

⁵ FERSTL/SINZ *Flexible Organizations* 1997.

der Integration sind die Kontrolle der Redundanz und der Verknüpfung von Systemkomponenten sowie die Erreichung von globaler Konsistenz und Aufgabenerfüllung.

Der Zielerreichungsgrad ist abhängig von der verwendeten Integrationsform. Folgende Integrationsformen für Teil-Anwendungssysteme werden unterschieden ⁶:

- a) **Personelle Funktionsintegration:** Funktionen unterschiedlicher Teil-Anwendungssysteme werden an einem Arbeitsplatz durch einen personellen Aufgabenträger integriert. Dieser aktiviert die einzelnen Funktionen und übergibt Daten von einer Funktion zur nächsten. Über die Erreichung von Integrationszielen ist bei dieser Integrationsform keine Aussage möglich.
 - b) **Datenflußorientierte Funktionsintegration:** Teil-Anwendungssysteme mit eigenen Datenbanken tauschen auf Kommunikationskanälen Daten aus. Diese Form der Integration zielt insbesondere auf die Verknüpfung von Systemkomponenten.
 - c) **Datenintegration:** Teil-Anwendungssysteme sind über eine gemeinsame Datenbasis integriert. Diese Form erlaubt vor allem eine Kontrolle von Datenredundanz und Konsistenz. Datenintegration ist die derzeit in der Praxis dominierende Integrationsform.
 - d) **Objektintegration:** Teil-Anwendungssysteme werden in Form von gekapselten Objekten und Objektklassen realisiert, die jeweils über eine lokale Datenbasis verfügen und untereinander durch Nachrichten interagieren. Durch die Verfügbarkeit von aktiven Objekten und speziellen Objekten zur Vorgangssteuerung erlaubt diese Integrationsform die Kontrolle aller genannten Integrationsziele. Mit der zunehmenden Verbreitung objektorientierter Konzepte setzt sich diese Integrationsform in der Praxis durch.
- Wie sind die Anwendungssysteme verteilt?

Monolithische Anwendungssysteme werden zunehmend durch verteilte Anwendungssysteme auf der Grundlage des Client/Server-Prinzips abgelöst. Ein verteiltes Anwendungssystem stellt aus Außensicht *ein* System dar, welches eine bestimmte Aufgabenstellung erfüllt. Das Anwendungssystem besteht aus einer Menge autonomer Komponenten, die zur Erfüllung dieser Aufgabenstellung kooperieren. Die Verteilung der Aufgaben auf die einzelnen Komponenten sowie die Struktur der Verteilung ist aus Außensicht nicht notwendigerweise bekannt. Bei einer Verteilung nach dem Client/Server-Prinzip werden i.a. Komponenten für Präsentation, für Applikation und für Datenverwaltung gebildet. Die Datenbasis kann dabei global oder selbst wiederum verteilt sein.

Die genannten Merkmale der Automatisierung von Geschäftsprozessen durch Anwendungssysteme, sowie der prozeßgerechten Gestaltung, der Integration und der Verteilung von Anwendungssystemen werden durch die Kartierung von Anwendungssystemen in Geschäftsprozeßmodellen erfaßt und explizit dargestellt. Die Kartierung leistet einen Beitrag zum Management von Anwendungssystemen, zur Beurteilung der Eignung von Anwendungssystemen sowie zur korrespondierenden Weiterentwicklung von Geschäftsprozeßmodellen und Anwendungssystemen.

Die Landschaft der Anwendungssysteme an Universitäten ist außerdem durch zunehmende Heterogenität gekennzeichnet. Standard-Softwarekomponenten (z.B. von HIS und SAP) werden mit eigenentwickelten Systemen kombiniert. Workflow-Managementsysteme, Groupware-

⁶ FERSTL/SINZ Grundlagen der Wirtschaftsinformatik 1994, S. 200 ff.

Systeme, Electronic Mail und Word Wide Web (WWW) sind mit diesen Systemen zu integrieren. Voraussetzung hierfür ist eine Kartierung der Anwendungssysteme in der beschriebenen Form.

Auf die Universitäten kommen in diesem Zusammenhang zusätzliche Anforderungen zu: Wie kann die organisatorische und systemtechnische Betreuung der Anwendungssysteme sichergestellt werden? Während letztere i.a. von den Rechenzentren übernommen wird, muß die organisatorische Betreuung unter Beteiligung der Fachabteilungen geleistet werden.

Als Beispiel zur Kartierung von Anwendungssystemen in Universitätsprozeßmodellen ⁷ wird das bereits in Teil II, Kapitel 1, Abschnitt 1.3 eingeführte Rückmeldesystem verwendet. Ausgehend vom Sollkonzept des Universitätsprozeßmodells (Abbildung 1-7) zeigt Abbildung 2-1 die Automatisierbarkeit und die Automatisierung von Aufgaben sowie die Kombination automatisierter Aufgaben bzw. Teilaufgaben zu Anwendungssystemen.

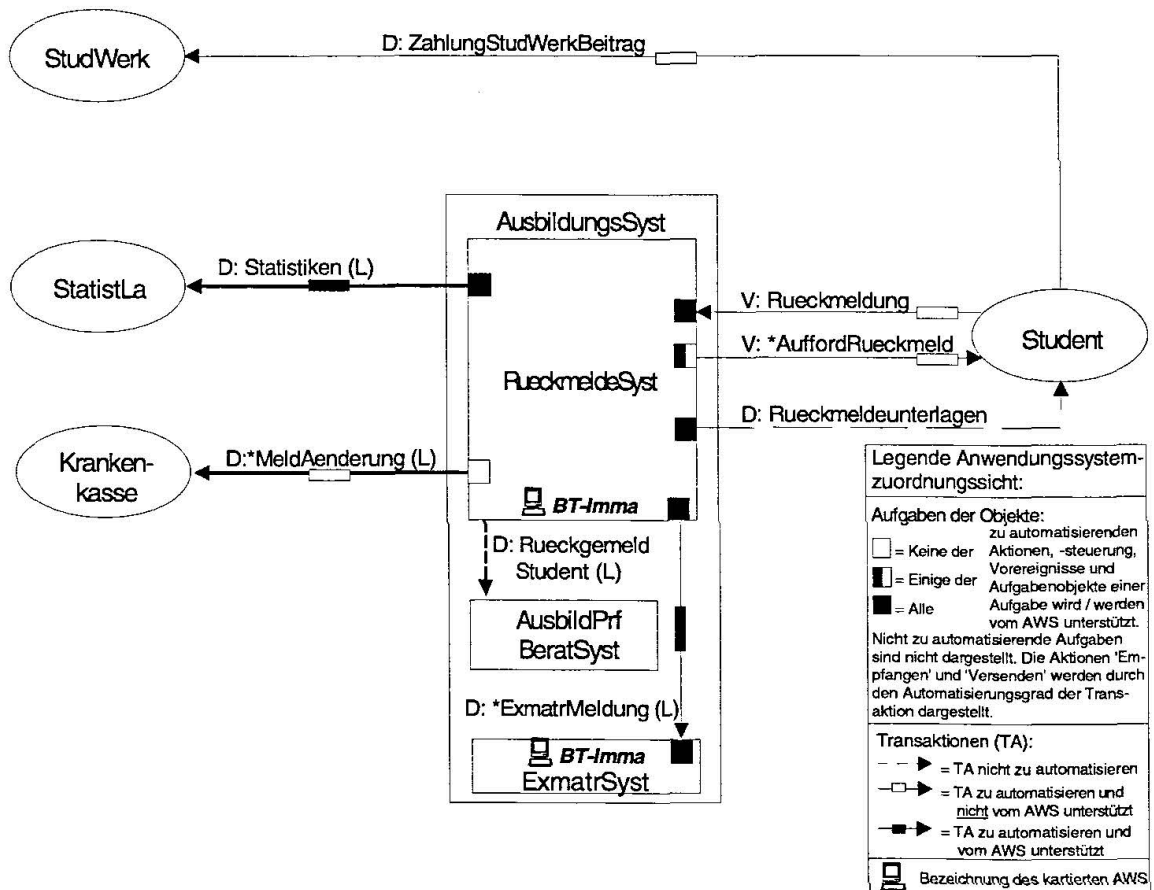


Abbildung 2-1: Automatisierung und Anwendungssystem-Zuordnung im Ausschnitt Rückmeldung des Universitätsprozesses Studium und Lehre

Zur Erläuterung der Darstellung sollen folgende Beispiele dienen:

⁷ KRUMBIEGEL Integrale Gestaltung 1997.

- Die Aufgabe in *RückmeldeSyst*, welche Statistiken für *StatistLa* erstellt, ist vollautomatisiert. Ebenfalls automatisiert ist die Transaktion, welche die Statistiken übergibt.
- Änderungsmeldungen an *Krankenkasse* werden sowohl manuell erstellt (nichtautomatisierte Aufgabe) als auch in Papierform versandt (nichtautomatisierte Transaktion).
- Rückmeldeunterlagen werden automatisiert erstellt, aber in Papierform übermittelt.
- Die Aufgabe zur Generierung von *AuffordRueckmeld* (Aufforderung zur Rückmeldung) wäre vollautomatisierbar, ist aber lediglich teilautomatisiert. Die Aufforderungen selbst werden in Papierform erstellt und versandt.

Die automatisierten Aufgaben und Teilaufgaben von *RückmeldeSyst* und *ExmatrSyst* sind zu einem Anwendungssystem (*BT-Imma*) zusammengefaßt. Teil-Anwendungssysteme von *BT-Imma* und deren Integrationsform werden nicht betrachtet.

Abbildung 2-2 zeigt einen Ausschnitt aus der Integrationssicht der Anwendungssysteme zur Unterstützung des Universitätsprozesses *Studium und Lehre*. Die automatisierten Aufgaben und Teilaufgaben des Objekts *VergabeSystem* werden durch das Anwendungssystem *HISZUL-UNIX*, die der Objekte *Immatrikulationssystem* und *Rückmeldesystem* durch *HISSOS-GX* und die des Objekts *PrüfungsSyst* durch *FlexNow* durchgeführt.

Die Anwendungssysteme *HISZUL-UNIX* und *HISSOS-GX* sind datenintegriert, d.h. sie greifen auf eine gemeinsame Datenbasis zu. Diese Integrationsform vermeidet Datenredundanzen und dadurch verursachte Dateninkonsistenzen, ist jedoch nicht geeignet, die globale Aufgabenerfüllung zu unterstützen.

HISSOS-GX und *FlexNow* sind datenflußintegriert, d.h. sie tauschen zu bestimmten Zeitpunkten Datenflüsse aus, die zur Aktualisierung der jeweils anderen Datenbasis genutzt werden. Temporäre Inkonsistenzen sowie Datenredundanzen sind dabei unvermeidlich. Diese Integrationsform ist somit unbefriedigend.

2.2 Entwicklung prozeßorientierter Anwendungssysteme

In Abschnitt 2.1 wurde die Kartierung von Anwendungssystemen in Geschäftsprozeßmodellen vorgestellt. Darüber hinaus stellen Geschäftsprozeßmodelle eine geeignete Grundlage für die systematische Entwicklung des Fachkonzepts prozeßorientierter Anwendungssysteme dar. Dies wird im folgenden am Beispiel der Entwicklung eines Anwendungssystems zur Unterstützung der Abwicklung von Prüfungen aufgezeigt⁸. Das Beispiel beschreibt in vereinfachter Form die Konzeption des Anwendungssystems *FlexNow*, welches an der Universität Bamberg in vier wirtschaftswissenschaftlichen Studiengängen zur Unterstützung eines studienbegleitenden, flexiblen Prüfungssystems verwendet wird. Es werden ausschließlich schriftliche Prüfungen betrachtet.

⁸ SINZ Service-Orientierung der Hochschulverwaltung 1995.

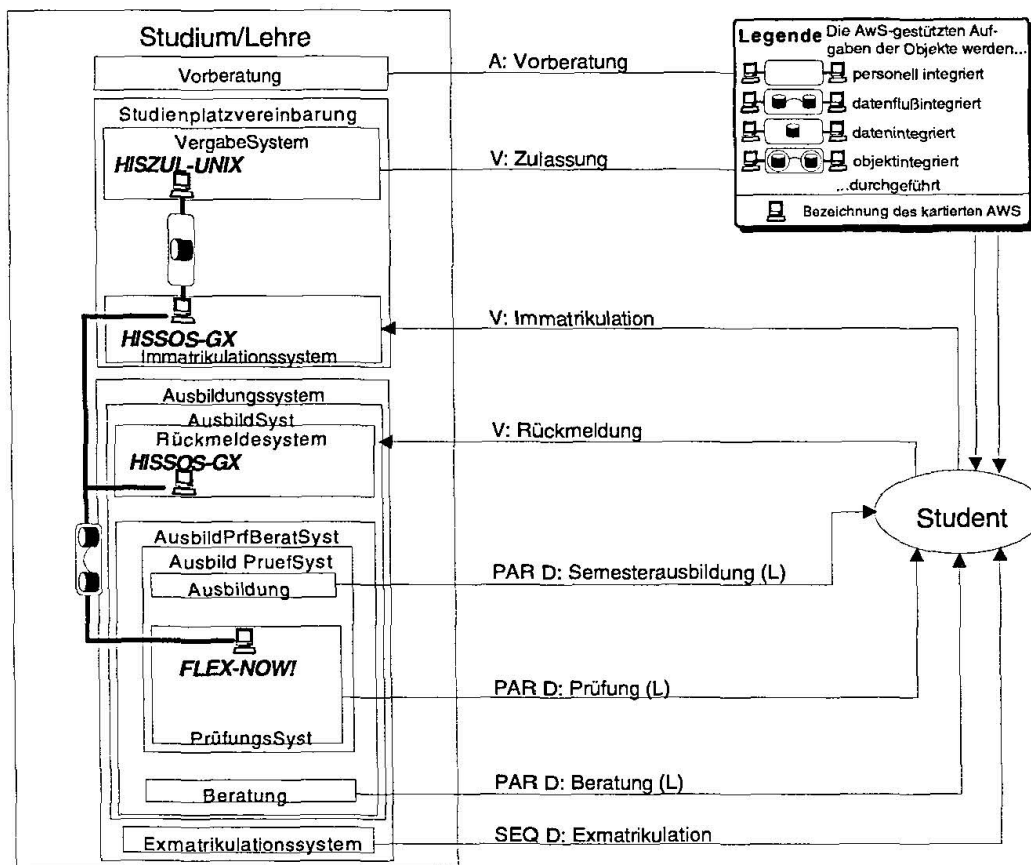


Abbildung 2-2: Integrationssicht der Anwendungssysteme des Universitätsprozesses Studium und Lehre (Ausschnitt)

Ausgangspunkt ist der durch die Transaktion *Prüfung* zwischen *Prüfungssystem* und *Student* abgegrenzte Ausschnitt des Universitätsprozeßmodells des Hauptprozesses *Studium und Lehre* (Abbildung 2-2). Die Entwicklung des Fachkonzepts erfolgt dann in mehreren Schritten:

1. Zunächst wird durch Zerlegung der Transaktion *Prüfung* in Teiltransaktionen das fachliche Protokoll für die Übergabe dieser Leistung aufgedeckt.
2. Anschließend wird für die einzelnen Teiltransaktionen geprüft, welche Aufgaben von den Prüfern selbst durchgeführt werden müssen und welche an die Prüfungsabwicklung delegierbar sind.
3. Im dritten Schritt werden entsprechende Serviceprozesse abgegrenzt, die vom verbleibenden Hauptprozeßanteil des Prüfungssystems beauftragt werden.
4. Zuletzt wird die Automatisierbarkeit der Aufgaben und Transaktionen geprüft, ihre tatsächliche Automatisierung festgelegt und das Fachkonzept für das Anwendungssystem aufgestellt.

Die einzelnen Schritte werden im folgenden näher ausgeführt.

Ad 1: Die Verfeinerung der Transaktion *Prüfung* führt zu folgender Zerlegungsstruktur (Abbildung 2-3):

- Prüfung
 - A: Prüfungsangebot
 - A.A: Veröffentlichung Prüfungsangebot
 - A.V: Anforderung Anmeldeformular für Prüfung
 - A.D: Anmeldeformular für Prüfung
 - V: Prüfungszulassung
 - V.A: Aufforderung zur Prüfungsanmeldung
 - V.V1(PAR): Prüfungsanmeldung
 - V.V2(PAR): Allgemeine Nachweise
 - V.D: Zulassung zur Prüfung
 - D: Prüfungsabnahme
 - D.V1(SEQ): Klausuraufgabenstellung
 - D.V2(SEQ): Bearbeitete Klausur
 - D.D: Prüfungsergebnis

Abbildung 2-3: Zerlegung der Transaktion Prüfung

Jede *Prüfung* wird durch die Veröffentlichung eines geplanten *Prüfungsangebots* angebahnt, durch eine *Prüfungszulassung* vereinbart und in Form einer *Prüfungsabnahme* durchgeführt. Die einzelnen Teiltransaktionen werden unter Nutzung des Verhandlungsprinzips weiter verfeinert. Zum Beispiel wird in der Transaktion *Prüfungsabnahme* das *Prüfungsergebnis* in zwei sequentiellen Schritten vereinbart: seitens des Prüfers durch die *Klausuraufgabenstellung*, seitens des Studenten durch die *Bearbeitete Klausur*. Die Leistungsübergabe erfolgt anschließend in der Durchführungsphase *Prüfungsergebnis*.

Die Zerlegung der Transaktion *Prüfung* zeigt das fachliche Protokoll, welches zwischen dem Objekt *Prüfungssystem* und dem Umweltobjekt *Student* im Rahmen von Prüfungen abzuwickeln ist. Jede Transaktion wird durch je eine Aufgabe der beiden beteiligten Objekte durchgeführt. Zum Beispiel wird die Transaktion *Zulassung zur Prüfung* von der Aufgabe *Erzeugen und Versenden Zulassung zur Prüfung* des Objekts *Prüfung* und der Aufgabe *Empfangen Zulassung zur Prüfung* des Objekts *Student* durchgeführt. Da die Aufgabenbezeichnungen aus den Transaktionsbezeichnungen ableitbar sind, wird im folgenden auf ihre Nennung verzichtet.

Ad 2: Bei allen durch Zerlegung der Transaktion *Prüfung* entstandenen Teiltransaktionen ist eine Unterstützung durch Services der Prüfungsabwicklung möglich und wünschenswert. So können alle Teiltransaktionen der Anbahnungs- und Vereinbarungsphase von der Prüfungsabwicklung durchgeführt werden (Unterstützung der Lenkung). In der Durchführungsphase kann die Prüfungslogistik - vom Ausgeben der Klausuraufgabenstellung bis zum Einsammeln der bearbeiteten Klausuren - von der Prüfungsabwicklung übernommen werden. Eine weitere Unterstützung ist bei der Bekanntgabe der Prüfungsergebnisse möglich (Unterstützung der eigentlichen Leistungserstellung).

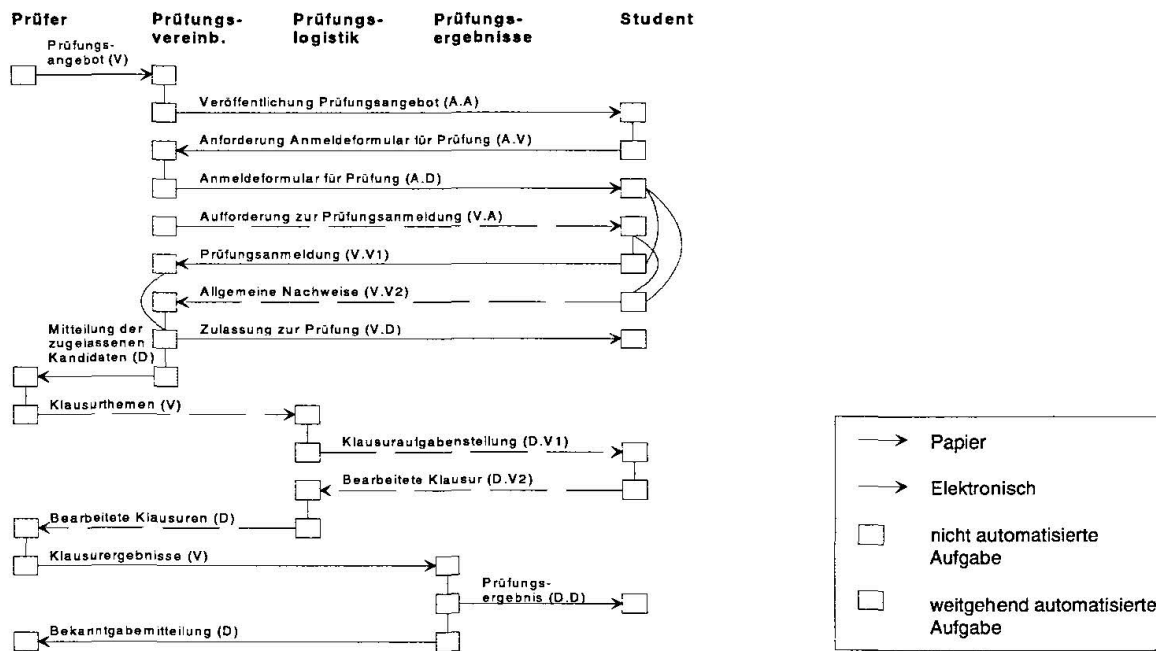


Abbildung 2-5: Prozeßmodell des Prüfungswesens (Ablaufsicht)

Durch die Leistungs-, die Lenkungs- und die Ablaufsicht ist ein Geschäftsprozeßmodell im SOM-Ansatz vollständig spezifiziert. Die drei Sichten sind über einem integrierten Meta-Modell⁹ definiert, anhand dessen die Konsistenz der Sichten in sich und zueinander überprüft werden kann.

Ad 4: Der in den vorausgehenden Schritten entwickelte Ausschnitt *Prüfungswesen* des Geschäftsprozeßmodells der Universität wird nun als Grundlage für die Spezifikation der Architektur eines workflow-orientierten Anwendungssystems zur Unterstützung des Prüfungswesens verwendet. Die einzelnen Schritte der Spezifikation sind im folgenden kurz dargestellt:

- *Abgrenzung des Anwendungssystems:* Das workflow-orientierte Anwendungssystem soll die Prüfungsabwicklung bei den im Rahmen des Prüfungswesens zu erbringenden Services unterstützen. Das bedeutet, daß das Anwendungssystem die Objekte *Prüfungsvereinbarung*, *Prüfungslogistik* und *Prüfungsergebnisse* unterstützen soll. Genauer: das Anwendungssystem übernimmt die automatisierbaren Anteile der im Rahmen der Transaktionen dieser Objekte durchzuführenden Aufgaben¹⁰.
- *Ableitung des Workflow innerhalb des Anwendungssystems:* Der Workflow ergibt sich aus einer Projektion der Ablaufsicht des Geschäftsprozesses auf die vom Anwendungssystem zu unterstützenden Objekte, erweitert um eine interne Kontrolle externer Ereignisbeziehungen. Zum Beispiel ist durch einen internen Workflow zu kontrollieren, daß vom *Prüfer* Klausurthemen für alle zugelassenen Kandidaten bereitgestellt werden (Abbildung 1-5).
- *Analyse der Automatisierbarkeit von Aufgaben und der in Transaktionen genutzten Übertragungsmedien:* In Abbildung 2-5 sind nicht-automatisierbare Aufgaben der Objekte *Prüfungsvereinbarung*, *Prüfungslogistik* und *Prüfungsergebnisse* ohne Schattierung, (weitgehend) automatisierbare Aufgaben mit grauer Schattierung dargestellt. Außerdem sind den Transaktio-

⁹ FERSTL/SINZ Der Ansatz des Semantischen Objektmodells 1995, S. 216.

¹⁰ Vgl. auch FERSTL/SINZ Multi-Layered Development 1996.

nen Übertragungsmedien zugeordnet. Durchgezogene Kanten symbolisieren eine elektronische, gestrichelte Kanten eine papiergestützte Übertragung.

- *Festlegung der Systemplattformen:* Im vorliegenden Beispiel wird davon ausgegangen, daß das Anwendungssystem auf der hybriden Plattform eines Datenbanksystems (PRF-DBS) und von WorldWideWeb (WWW) realisiert werden soll. Das auf dem Internet basierende WWW und die zugehörige Internet-Mail dienen zur Kommunikation mit dem Umweltobjekt *Student*. Das Objekt *Prüfer* erhält direkten Zugriff auf das Datenbanksystem.
- *Spezifikation der Integrationsform für die automatisierten Aufgaben des Anwendungssystems:* Hier werden die Integrationsformen der Datenintegration und der Objektintegration genutzt¹¹. Datenintegration wird für die Aufgaben des PRF-DBS, Objektintegration zur Kopplung von PRF-DBS und WWW verwendet. Auf eine automatisierte Vorgangsauslösung und Vorgangsteuerung durch ein Workflow-Managementsystem wird verzichtet.

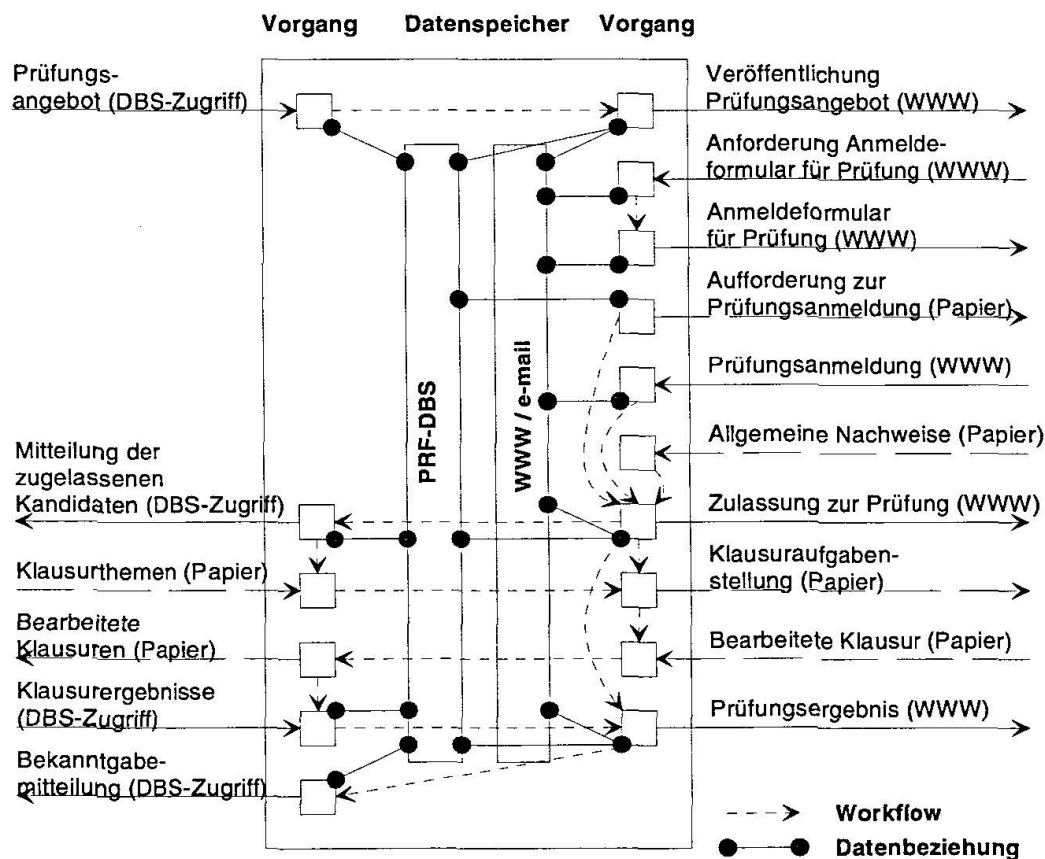


Abbildung 2-6: Workflow-orientiertes Anwendungssystem zur Unterstützung des Prüfungswesens

Abbildung 2-6 zeigt die Architektur des gesamten Anwendungssystems im Überblick. Die einzelnen Aufgaben sind hier aufgeteilt in die Aufgabenverrichtung (Vorgang) und das Aufgabenobjekt, auf dem die Verrichtung durchgeführt wird (Datenspeicher). Die Beziehungen zwischen Vorgang

¹¹ FERSTL/SINZ Grundlagen der Wirtschaftsinformatik 1994, S. 200 ff.

und Datenspeicher sind als Datenbeziehungen, Beziehungen zwischen Vorgängen als Workflowbeziehungen dargestellt.

Literatur

- FERSTL O.K./SINZ E.J.: Grundlagen der Wirtschaftsinformatik. 2. Auflage, Oldenbourg, München 1994.
- FERSTL O.K./SINZ E.J.: Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen. In: Wirtschaftsinformatik 37 (1995) 3, S. 209-220.
- FERSTL O.K./SINZ E.J.: Multi-Layered Development of Business Process Models and Distributed Business Application Systems. An Object-Oriented Approach. In: König W., Kurbel K., Mertens P., Preßmar D. (ed.): Distributed Information Systems in Business. Springer, Berlin 1996, p. 159 - 179.
- FERSTL O.K./SINZ E.J.: Flexible Organizations Through Object-oriented and Transaction-oriented Information Systems. In: Krallmann H. (Hrsg.): Wirtschaftsinformatik '97. Internationale Geschäftstätigkeit auf der Basis flexibler Organisationsstrukturen und leistungsfähiger Informationssysteme. Physica-Verlag, Heidelberg 1997, S. 393 - 411.
- KRUMBIEGEL J.: Anwendungssystem-Kartierung in den Soll-Modellen der Universitätsprozesse 'Studium und Lehre' und 'Personal'. Diskussionsbeiträge zur Optimierung von Universitätsprozessen, Bamberg, März 1996.
- KRUMBIEGEL J.: Integrale Gestaltung von Geschäftsprozessen und Anwendungssystemen in Dienstleistungsbetrieben. Dissertation, Deutscher Universitätsverlag, Heidelberg 1997.
- SINZ E.J.: Service-Orientierung der Hochschulverwaltung und ihre Unterstützung durch workflow-orientierte Anwendungssysteme. In: Knop J. (ed.): Trends in Academic Information Systems in Europe. European University Information Systems. Proc. EUNIS'95 Congress, 6. - 8. November 1995, Düsseldorf 1995, S. 225 - 237.
- SINZ E.J.: Architektur betrieblicher Informationssysteme. In: Rechenberg P., Pomberger G. (Hrsg.): Informatik-Handbuch. Hanser-Verlag, München 1997, S. 875 - 887.