

# Integration von Informationssystemen aus der Perspektive eines Herstellers von GUI-Entwicklungswerkzeugen

Christian Raether, Oliver Pape, Haiko Cyriaks

ISA Informationssysteme GmbH  
Azenbergstraße 35  
70174 Stuttgart  
{raetherlpapelcyriaks}@isa.de

**Abstract:** Der Beitrag beschreibt Problemstellungen und Lösungen im Zusammenhang mit der Integration von Applikationen, die aus Sicht eines Herstellers eines User Interface Management Systems (UIMS) in der Praxis von Bedeutung sind. Für eine praxisorientierte Klassifizierung von Integrationsvorhaben werden verschiedene Merkmale dargestellt, anhand derer eine Unterscheidung in leicht- und schwergewichtige Integrationsverfahren getroffen wird. Ein von mehreren Großkunden angewendetes Verfahren ist die Nutzung von GUI-Modellbibliotheken für die Integration. Zunehmend spielen auch serviceorientierte Konzepte bei einer Integration über die Benutzeroberfläche eine Rolle. Anhand zweier Kundenprojekte werden die beiden genannten Ansätze illustriert. Abschließend werden exemplarisch zukünftige Entwicklungen eines UIMS beschrieben, mit denen die Integrationsfähigkeit der damit erstellten Applikationen erhöht werden kann.

## 1 Einführung

In Kundenprojekten zeigte sich beim Einsatz des User Interface Management Systems (UIMS) ISA Dialog Manager schon sehr früh nach dessen Markteinführung, dass die Entwicklung von Benutzeroberflächen häufig einen starken Integrationsaspekt aufweist. Entsprechend bilden Funktionen zur Anbindung von Datenbeständen und Applikationen in der Produktentwicklung des ISA Dialog Managers (IDM) einen weiteren Schwerpunkt neben den Möglichkeiten für die Gestaltung von grafischen Benutzeroberflächen (Graphical User Interface, GUI). Viele GUI-Werkzeuge anderer Anbieter haben sich inzwischen ebenfalls zu Integrationswerkzeugen weiterentwickelt, da die Möglichkeiten zur Gestaltung von Benutzeroberflächen für Geschäftsanwendungen inzwischen kaum noch Spielraum für Alleinstellungsmerkmale bieten.

Ein weiterer Grund für die Entwicklung von GUI-Werkzeugen hin zu Integrationswerkzeugen liegt darin, dass eine Integration über die Benutzeroberfläche bzw. Präsentationsschicht unter bestimmten Voraussetzungen relativ einfach – verglichen mit einer Integration auf Funktions- oder Datenebene – durchführbar ist. Günstige Voraussetzungen bestehen dafür bei „vertikalen“, Datenbank-gestützten Applikationen

mit, im Hinblick auf Anzahl und Komplexität, geringen funktionalen Verbindungen zu anderen Anwendungen. Die Verwendung eines UIMS unterstützt die Entwicklung von Applikationen mit einer Model-View-Controller (MVC) Struktur. Die damit verbundene Abgrenzung von Datenhaltung, Geschäftslogik und Präsentation leistet ebenfalls einen Beitrag zur Verbesserung von Integrationsmöglichkeiten.

Eine Integration über die Präsentationsschicht wird auch in Zukunft eine Alternative darstellen, da sie mit neueren Entwicklungen (z.B. Web Services) kompatibel ist. Sie kann auch eine Zwischenstufe zu einem umfassenden Integrationskonzept wie Service-orientierte Architekturen (SOA) bilden, mit der Investitionen in Anwendungen geschützt sowie das technische und wirtschaftliche Risiko einer Integration verringert werden können.

## **2 Bisherige Entwicklungen und Erfahrungen**

Zur Zeit der Entstehung des ISA Dialog Managers Mitte der 80er Jahre war die Hauptanforderung in Bezug auf Integration, dass Anwendungen, die teilweise über kein eigenes GUI verfügten, unter einer einheitlichen grafischen Benutzungsoberfläche integriert werden konnten. Dieser Anforderung begegnet der IDM durch seine Schnittstellen für die Programmiersprachen COBOL, C/C++ und Java. Im Lauf der Zeit sind weitere Integrationsanforderungen entstanden, bei denen es um den Datenaustausch zwischen mehreren Applikationen und die gemeinsame Nutzung von Funktionen geht. In diesem Zusammenhang entstanden die DDE (Dynamic Data Exchange) und OLE (Object Linking and Embedding) Schnittstellen des IDM unter Microsoft Windows sowie seine XML-Schnittstelle.

Neben den genannten allgemeinen Schnittstellen des IDM wurden im bisherigen Verlauf seines Produktlebenszyklus bereits mehrere kundenspezifische Integrationslösungen implementiert. Ein Beispiel ist ein Konverter, zur Konvertierung alphanumerische Bildschirmmasken, die mit dem IFG (Interaktiver Formatgenerator) für BS2000 Hostanwendungen erstellt werden, in grafische Benutzeroberflächen. Damit können Anwendungen, die auf dem Fujitsu-Siemens Betriebssystem BS2000 laufen, in die modernen, grafisch-interaktiven Desktops von Sachbearbeitern integriert werden.

Die Problemstellungen in Software-Entwicklungsprojekten, mit denen sich die ISA Informationssysteme GmbH bei ihren Kunden konfrontiert sieht, ändern sich zunehmend. Der Anteil der vollständigen Neuentwicklung von Applikationen, mit denen eine Lücke in der IT-Landschaft von Unternehmen oder Behörden gefüllt wird, nimmt ab, während die Bedeutung der Weiterentwicklung, Anpassung und Integration vorhandener Anwendungen zunimmt. Das Wissen und die Erfahrung, die bei der Entwicklung und während eines oft jahrzehntelangen Betriebs durch Wartung, Pflege und Weiterentwicklungen in die Anwendungen eingeflossen sind, schließen einen Ersatz dieser Anwendungen durch vollständige Neuentwicklungen in vielen Fällen aus. Als gangbarer Weg verbleibt oft nur eine evolutionäre Weiterentwicklung von Teilbereichen der Applikationen, verbunden mit der fortwährenden Integration von alten und neuen

Komponenten. Für viele Kunden bedeutet Integration damit die Erhaltung der in der Applikationsentwicklung geschaffenen Werte.

Bedingt durch ihre lange Entwicklungs- und Einsatzzeit sind Konzept und Zuschnitt vieler Individualentwicklungen mehrere Jahre oder Jahrzehnte alt und genügen daher nicht mehr aktuellen Anforderungen aus Geschäftsmodell, Organisation oder Technik. Häufig können die Applikationen aber aus Kosten- oder Zeitgründen bzw. auf Grund des Fehlens von Ressourcen oder Know-how nicht ersetzt oder angepasst werden. Daher benötigen Mitarbeiter in Unternehmen und Behörden oft mehrere Anwendungen und Informationen aus unterschiedlichen Datenbeständen für die Erfüllung ihrer Aufgaben. Die benötigten Anwendungen bilden ein „Sammelsurium“ aus Desktop-, Server-/Host- und zunehmend Web-Anwendungen (die im Grunde ja Server-Anwendungen auf Basis von Internet-Technologien sind). Um ein produktives Arbeiten mit dieser Applikationsvielfalt zu ermöglichen, müssen durch Integration Möglichkeiten für den Datenaustausch und die Nutzung von Funktionen außerhalb des ursprünglich vorgesehenen Anwendungskontextes geschaffen werden.

### **3 Einordnung von Integrationsvorhaben**

In einer systemtheoretischen Definition versteht man unter Integration „...die Verknüpfung von Elementen bzw. Subsystemen zu einem System. Dies kann durch die Verbindung oder Verschmelzung bisher eigenständiger Elemente bzw. Subsystemen geschehen.“ ([Rau93] zitiert nach [Thr05]) Die Integration von Informationssystemen weist einen Zustands- und einem Prozessaspekt auf. Dabei beschreibt der Prozessaspekt einen Integrationsvorgang. Durch den Zustandsaspekt werden Soll- oder Ist-Zustände charakterisiert, die vor, während oder nach einem Integrationsvorgang vorliegen. (in Anlehnung an [Thr07]) Integrationsvorhaben können anhand verschiedener Dimensionen des Zustands- und Prozessaspekts klassifiziert werden (Abbildung 1).

|  | Dimensionen des Zustandsaspekts                                |  |   |
|--|--|--|---|
|  | horizontal   |  | vertikal  |
| Integrationsrichtung   |  |  |   |
| Integrationsgrad (-stärke)<br>exemplarische Bezeichnungen                        | niedrig, lose<br>Kopplung,<br>Verknüpfung                      | mittel<br>Kombination,<br>Zusammenschluss  | hoch, fest<br>Vereinigung, Ver-<br>schmelzung, Ein-<br>gliederung |
| Autonomie der Integrations-<br>objekte   | keine  | gering   | hoch  |
| Integrationsfähigkeit der<br>Integrationsobjekte                                 | nicht vorhanden  | nicht vorhanden<br>aber herstellbar  | vorhanden   |
| beteiligte Komponenten/Systeme<br>(Anzahl der Integrationsobjekte)               | 2-5/<br>1-2  | 6-12/<br>3-4   | >12/<br>>4  |
| Veränderung der Anzahl von<br>Integrationsobjekten                               | keine  | Zunahme  | Abnahme   |
| führende Integrationsobjekte   | keins  | eins   | mehrere   |
|  | Dimensionen des Prozessaspekts                                 |  |   |
|  | ex-ante/Integrationsfähigkeit                                  |  | ex-post/Integration   |
| Integrationszeitpunkt/Aufgabe  |  |  |   |
| Projektcharakter   | konkrete, spezifische<br>Lösung                                | generische Lösung und<br>erstmaliges Anwenden                                      | Anwenden einer<br>generischen Lösung                              |
| Bekanntheit von Systemen und<br>Prozessen  | unbekannt  | teilweise  | vollständig   |
| Anzahl/Verteilung von beteiligten<br>und betroffenen Organisations-<br>einheiten | 1-3/gering,<br>intern bei<br>kleiner/mittlerer<br>Organisation | 4-8/mittel<br>intern bei großer<br>Organisation,<br>organisationsüber-<br>greifend | >8/hoch<br>organisations-<br>übergreifend                         |

Abbildung 1: Dimensionen des Zustands- und Prozessaspekts von Integrationsvorhaben  
(eigene Darstellung in Anlehnung an [Thr05, Thr07])

Thranert führt – neben weiteren – die „Invasivität“ in Bezug auf die Integrationsobjekte als Dimension des Zustandsaspekts einer Integration an [Thr07]. Sie gibt an, ob im Rahmen der Integration Änderungen an den Integrationsobjekten vorgenommen werden. Obwohl dies ein wichtiges Kriterium bei der Durchführung von Integrationsvorhaben ist, verzichten wir darauf, die Invasivität gesondert aufzuführen, da sie in der Praxis sehr eng mit dem Integrationsgrad und der vorhandenen Integrationsfähigkeit der Integrationsobjekte korreliert ist. In der deutschen Wikipedia werden Kopplung und Integration gegeneinander abgegrenzt. Bei einer Kopplung werden Schnittstellen von Systemen oder Komponenten für deren Verbindung genutzt während bei einer Integration Schnittstellen abgebaut oder vermieden werden [Wik08a, Wik08b]. Wir folgen dieser Abgrenzung nicht, sondern verstehen unter Kopplung eine Integrationsform mit niedrigem Integrationsgrad. Bei dieser Integrationsform werden Instrumente entwickelt und eingesetzt, mit denen die Überbrückung von System- oder Komponentengrenzen automatisiert und sie z.B. aus Sicht der Benutzer transparent gemacht werden.

In einer groben Unterteilung lassen sich Integrationsprobleme zwei Bereichen zuordnen. Der übergeordnete Bereich umfasst die IT-Architektur und betrachtet die Informations- und Kommunikationssystem-Landschaft von Unternehmen und Behörden als Ganzes. Auf diesen Bereich beziehen sich Konzepte wie das Zachman-Framework ([SoZa92]), das einen methodischen Rahmen für die Gestaltung der IT-Architektur beschreibt, und Ansätze wie Enterprise Application Integration (EAI) und Serviceorientierte Architekturen. Bezogen auf die in Abbildung 1 genannten Dimensionen geht es in diesem Bereich häufig um die Entwicklung generischer, von einzelnen Applikationen und Datenbeständen unabhängige Lösungen und eine ex-ante Integration mit der Integrationsfähigkeit geschaffen werden soll. Die Integrationsobjekte dieses Bereichs

weisen zum Teil einen höheren Abstraktionsgrad auf (z.B. Modelle). Der zweite Integrationsbereich ist der Bereich der konkreten Anwendungen und Datenbestände. Auch in diesem Bereich wird der Begriff Architektur verwendet, bezieht sich hier aber auf die interne Struktur von Applikationen und Daten. Bei Integrationsvorhaben werden überwiegend konkrete, spezifische Lösungen entwickelt oder vorhandene, generische Lösungen angewendet. Ziele von Integrationsvorhaben in diesem Bereich sind z.B. die Nutzung des Datenbestands einer Anwendung mit einer zweiten Anwendung oder eine aufgabenorientierte Kombination von Funktionen verschiedener Applikationen.

In beiden genannten Integrationsbereichen kann sich die Integration auf die Daten-, Funktions- und Präsentationsebene beziehen [StTh08]. Als weitere Ebene wird teilweise die Geschäftsprozessebene genannt [Wik08b], die schwerpunktmäßig im Bereich der IT-Architektur angesiedelt ist. Die Zuordnung von Integrationsproblemen zu verschiedenen Integrationsebenen ist im Hinblick auf die einzusetzenden Methoden und Werkzeuge von Bedeutung. Reale Integrationsvorhaben betreffen allerdings eher selten ausschließlich eine Ebene. Auch wenn die Zielsetzung eines Integrationsvorhabens nur eine der Ebenen betrifft, führt es in vielen Fällen zu Anpassungen auf den anderen Ebenen.

Mit den agilen Konzepten für die Software-Entwicklung hat sich die Unterscheidung zwischen leichtgewichtigen und schwergewichtigen Vorgehensweisen etabliert, die auch auf Integrationsverfahren angewendet werden kann. Bei Vorgehensweisen der Software-Entwicklung wird die Unterscheidung unter anderem an Hand der Merkmale Anzahl bzw. Umfang und Detailliertheit von Regeln, Dokumentation und Planung getroffen. Allerdings hat sich noch keine gefestigte Definition von leicht- und schwergewichtig herausgebildet. [StTh08] In Hinblick auf die Unterscheidung zwischen leichtgewichtigen und schwergewichtigen Integrationsverfahren heben wir in diesem Beitrag nicht auf die eben genannten Merkmale ab, sondern beziehen uns auf drei Dimensionen des Zustandsaspekts von Integrationsvorhaben. Dabei geht es uns nicht um die Definition einer exakten Trennlinie, sondern um tendenzielle Aussagen:

- Ein Integrationsverfahren wird schwer-/leichtgewichtiger, je höher/niedriger die Integrationsstärke ist, die mit ihm erreicht werden soll. Dies hängt damit zusammen, dass nach unserer Erfahrung für einen hohen Integrationsgrad stärkere Eingriffe in die Integrationsobjekte erforderlich sind (siehe oben).
- Ein Integrationsverfahren wird schwer-/leichtgewichtiger, je schlechter/besser die Integrationsfähigkeit der Integrationsobjekte ist. Integrationsfähigkeit umfasst dabei sowohl den Anteil an Integrationsobjekten, die nutzbare Schnittstellen aufweisen, als auch die Eignung der Schnittstellen für die Ziele der Integration.
- Ein Integrationsverfahren wird schwer-/leichtgewichtiger, je mehr die Anzahl der Integrationsobjekte durch die Integration zu-/abnimmt. Dieser Punkt wirkt in gegensätzlicher Richtung zum ersten Punkt. Welchem Merkmal bei einer Integrationslösung die größere Bedeutung zukommt, hängt von der konkreten Situation ab und kann nicht allgemein beantwortet werden.

Bei den Aussagen ist zu beachten, dass sie sich auf die gesamte Lebensdauer einer Integrationslösung beziehen, also sowohl die Entwicklung als auch den Betrieb der

Lösung. Deshalb kann eine Serviceorientierte Integration, unabhängig davon wie leicht- oder schwergewichtig das Verfahren zur Herstellung der Servicefähigkeit vorhandener Applikationen ist, durch den Betrieb einer Serviceorientierten Architektur (SOA) mit einer komplexen Middleware-Schicht zu einem schwergewichtigen Ansatz werden.

## **4 Ein UIMS als Integrationswerkzeug**

### **4.1 Integration mit Hilfe von Modellbibliotheken**

Ein UIMS als Integrationswerkzeug für Applikationen wird überwiegend im Bereich der konkreten Anwendungen und Datenbestände zum Tragen kommen. Allerdings haben einige große Anwender des IDM sein Einsatzgebiet durch selbst entwickelte Bausteinbibliotheken in Richtung der Architekturebene erweitert. Der ISA Dialog Manager bietet die Möglichkeit, Komponenten einer Benutzeroberfläche in Form von Modellen zu entwickeln, die dann als Basis für die Benutzeroberflächenentwicklung eingesetzt werden.

Die von Anwendern entwickelten Modelle haben unterschiedliche Ausprägungen. Zum Teil handelt es sich einfach um Vorlagen für immer wieder benötigte GUI-Elemente (Widgets), zum Beispiel ein Eingabefeld mit einer bestimmten Gültigkeitsprüfung für die eingegebenen Werte. Andere Modelle haben abstrakteren Charakter und bieten generische Problemlösungen. Als Beispiel seien GUI-Komponenten genannt, die zur Entwicklungs- oder Laufzeit für verschiedene Datentypen konfigurierbar sind, die passenden Widgets auswählen und in der Benutzeroberfläche anzeigen. Dabei werden Konfigurationen meist in XML-Formaten definiert, zur Auswertung der Konfigurationen werden die XML-Schnittstelle des IDM und seine XML-Verarbeitungsfunktionen verwendet.

Von Kunden mit dem IDM entwickelte Modelle reichen von einzelnen Widgets mit einer bestimmten Funktionalität oder einem bestimmten Aussehen bis zu vollständigen Eingabefeldern für komplexe Datenstrukturen mit Abhängigkeiten zwischen den einzelnen Elementen des Formulars. Mit dem IDM sind mehrstufige Modellhierarchien möglich, so dass verschiedene fachliche Aspekte auf unterschiedlichen Modellebenen abgebildet werden können. Im Unterschied zu den objektorientierten Vererbungsmechanismen anderer GUI-Toolkits werden auch dynamische Änderungen der Modelle auf die daraus abgeleiteten GUI-Elemente übertragen. Das heißt, die Änderung eines Modells zur Laufzeit bewirkt dieselbe Änderung bei allen abgeleiteten Instanzen.

Die Nutzung von Modellbibliotheken führt zu einer erheblich höheren Produktivität bei der Entwicklung von Benutzeroberflächen, unabhängig davon ob es sich um die Entwicklung einer neuen oder die Anpassung einer vorhandenen Benutzeroberfläche handelt. Außerdem erleichtern es die Modellbibliotheken, Style Guides und Standards im Hinblick auf die Barrierefreiheit durchzusetzen, ein Aspekt, der in Integrationsprojekten regelmäßig von Bedeutung ist. Ausführliche Informationen zu Modellbibliotheken und Dialogbausteinen mit Beispielen finden sich in [Fae00].

## 4.2 Serviceorientierte Integration

In den letzten Jahren haben immer mehr Anwender des IDM damit begonnen, bei der Gestaltung ihrer IT-Landschaften und -Systeme serviceorientierte Konzepte einzusetzen. Einige Kunden nutzen dabei die Web Services Standards überwiegend als Referenz-Schnittstellen und -Protokolle. Diese Form der Nutzung bezeichnen wir als Serviceorientierte Integration. Prinzipiell ist Serviceorientierte Integration ein Ansatz auf Funktionsebene, mit dem Vorteil der hohen Verbreitung und Akzeptanz der Schnittstellen- und Protokolldefinitionen. Bisher verfolgen nur wenige Kunden der ISA die Zielsetzung, eine umfassende Serviceorientierte Architektur aufzubauen, da bei ihnen zumindest aktuell noch kein Bedarf an typischen Komponenten der Middleware-Schicht einer SOA (z.B. Service Repository, Service Bus) besteht.

Ob sich Serviceorientierte Integration im Sinne der oben getroffenen Unterscheidung als leicht- oder schwergewichtiges Integrationsverfahren erweist, kann in allgemeiner Form nicht beantwortet werden. Integration durch Nutzung von Web Services ist ein Konzept mit niedriger Integrationsstärke (lose Kopplung). Wie leicht- oder schwergewichtig dieses Konzept ist, hängt davon ab, welche Anpassungen an Applikationen notwendig sind, um sie servicefähig zu machen. Serviceorientierte Integration wird schwergewichtiger, je mehr sie sich in Richtung einer ausgebauten Serviceorientierten Architektur entwickelt. Dies ist durch die zunehmende Komponentenzahl und Komplexität der mit einer SOA verbundenen Middleware bedingt, deren Komponenten als Integrationsobjekte zu berücksichtigen sind.

Die Aufgabe, die Leistungen einer SOA auf den Desktop der Benutzer zu bringen, wird inzwischen teilweise als „Last Mile of SOA“ Problem bezeichnet [Buf07, McK07]. Diese Bezeichnung enthält viel Marketing, allerdings ist der darin enthaltene Vergleich mit einem Telekommunikations- oder Energieversorgungsnetz ein treffendes Bild für die Aufgabe. Im Zusammenhang mit dieser Last Mile of SOA wird nach unserer Erfahrung häufig explizit empfohlen oder implizit davon ausgegangen, dass neue Desktop-Anwendungen und neue, meist Browser-gestützte Benutzeroberflächen entwickelt werden. Allerdings hat sich in Kundenprojekten der ISA immer wieder gezeigt, dass – im Gegensatz zu einer Neuentwicklung des GUI – ein mit einem UIMS entwickeltes GUI einen geeigneten Ausgangspunkt dafür bietet, um eine Desktop-Applikation servicefähig zu machen. Der Einsatz eines UIMS fördert die Entwicklung von Anwendungen mit einer Schichtenstruktur, im Beispiel des IDM mit einer Model-View-Controller Struktur. Die dadurch bereits definierte Schnittstelle zwischen View und Controller bietet sich in vielen Fällen als Basis für die Definition von Services an. Dies kann zunächst erfolgen, um Funktionen einer bestehenden Anwendung in einer SOA zu nutzen. Es kann aber auch mit dem Ziel verbunden sein, eine Anwendung schrittweise abzulösen, indem ihre Funktionen nach und nach stillgelegt und ersetzt werden.

Bei der Definition von Services sollten fachliche über technische Aspekte gestellt werden und jeder Service eine oder mehrere fachliche Funktionen erfüllen. Für eine Service-Definition ausgehend von der View-Controller-Schnittstelle erweist es sich als Vorteil, dass auch grafische Benutzeroberflächen überwiegend nach fachlichen Gesichtspunkten strukturiert sind. Dialoge dienen in der Regel zur Anzeige und Be-

arbeitung von Geschäftsobjekten (Business Objects) und führen die entsprechenden Informationen und Funktionen zusammen. In tieferen Anwendungsschichten kommen zunehmend technische Kriterien für die Strukturierung hinzu, so dass Geschäftsobjekte und zugehörige Bearbeitungsfunktionen in diesen Schichten schwieriger zu identifizieren sind.

## **5 Zwei Beispiele aus aktuellen Projekten**

### **5.1 Integration mit Hilfe einer Modellbibliothek**

Aus Sicht der Benutzeroberflächenentwicklung ist Standardisierung ein wichtiger Aspekt der Integration. Bei einem Kunden der ISA Informationssysteme GmbH – einer großen Behörde – ist die Entwicklung von Anwendungen bundesweit auf mehrere Standorte verteilt. In der Regel setzt dabei jede Anwendung ein bestimmtes Fachverfahren um. An den Arbeitsplätzen der Behörde müssen die Anwendungen für die verschiedenen Fachverfahren in eine übergreifende Arbeitsplatzanwendung integrierbar sein. Der dabei bestehende Integrationsbedarf auf funktionaler Ebene stellt dabei in den meisten Fällen keine besondere Herausforderung dar. Die Fachverfahren sind relativ abgeschlossen und die entsprechenden Anwendungen sind vertikale, Datenbankgestützte Anwendungen mit wenigen Funktionen, die in anderen Verfahren sinnvoll eingesetzt werden können. Höherer Integrationsbedarf besteht auf der Datenebene, der sich allerdings vorrangig auf Stammdaten (z.B. von Bürgern) bezieht, weniger auf Bewegungsdaten, die bei der Bearbeitung der Fachverfahren entstehen und verarbeitet werden. Ein laufendes Projekt der Behörde besteht dementsprechend in der Schaffung einer durchgängigen, redundanzfreien Grunddatenbasis für alle Fachverfahren.

Bei der Klassifizierung dieses Kundenbeispiels anhand der in Abschnitt 3 dargestellten Dimensionen müssen im Hinblick auf Projektcharakter und Integrationszeitpunkt die Perspektiven der Modellbibliotheksentwicklung und der Fachverfahrensentwicklung unterschieden werden. Aus Sicht der Bibliotheksentwicklung handelt es sich um eine ex-ante Integration bei der generische Lösungen entwickelt werden. Aus der Perspektive der Entwicklung von Fachanwendungen geht es um eine ex-post Integration und das Anwenden der generischen Lösungen, die von der Modellentwicklung geschaffen wurden. Die übergreifende Arbeitsplatzanwendung ist führendes Integrationsobjekt, über die eine mittlere Integrationsstärke (Kombination, Zusammenschluss) erreicht werden soll. Autonomie und Anzahl der Integrationsobjekte sind hoch. Insgesamt müssen ca. 20 Fachverfahren integriert werden, dazu kommen übergreifende Anwendungen wie Stammdatenverwaltung und Dokumentenarchivierung. Anzahl und Verteilung der betroffenen Organisationseinheiten sind ebenfalls hoch.

Bei diesem Integrationsvorhaben spielt die Integrationsfähigkeit der entwickelten Fachanwendungen in die verfahrensübergreifende Arbeitsplatzanwendung der Sachbearbeiter eine sehr wichtige Rolle. Einheitliche Bedienbarkeit und durchgängiges Look-and-Feel sind wesentliche Einflussfaktoren auf die Produktivität der Sachbearbeiter. Zudem verringern sie den Schulungs- und Einarbeitungsaufwand bei jeder neuen oder



geänderten Fachanwendung, was bei weit über 100.000 Arbeitsplätzen bei allen Einrichtungen der Behörde im Bundesgebiet zu maßgeblich niedrigeren Einführungskosten führt.

Die Behörde nutzt eine selbst entwickelte Bibliothek aus ISA Dialog Manager Modellen, um die Integrationsfähigkeit der mit der Modellbibliothek entwickelten Anwendungen zu verbessern. Die Modellbibliothek wird zentral an einem Standort entwickelt und den Entwicklern an anderen Standorten als Basis für die Umsetzung von Fachverfahren bereitgestellt. Die Dialogmodelle werden plattformübergreifend für Microsoft Windows und Linux mit dem Fenstersystem Motif entwickelt, da die Bundesländer die Wahl ihrer Client- und Server-Plattformen unabhängig treffen und die genannten Betriebssysteme auf den Behördenarbeitsplätzen verschiedener Bundesländer eingesetzt werden.

Die Entwicklung von Fachanwendungen erfolgt geografisch verteilt in mehreren Bundesländern. Die Benutzerschnittstellen zu diesen Fachanwendungen werden auf Basis der Bibliothek aus Dialogmodellen entwickelt. Dies führt einerseits zu einem geringeren Aufwand für die Entwicklung der Benutzeroberfläche, andererseits ist über die Modelle eine Definition von Standards mit einer Detailgenauigkeit möglich, die mit herkömmlichen Style Guides auch dann nicht erreichbar wäre, wenn sie von allen Entwicklern berücksichtigt und exakt eingehalten würden.

#### **4.2 Serviceorientierte Integration als Vorstufe einer funktionalen Eingliederung**

Als zweites Beispiel soll im Folgenden die Situation in einem laufenden Integrationsprojekt bei einem Kunden der ISA aus der Versicherungsbranche beschrieben werden. Nach der Fusion zweier Unternehmen existieren für die Schadensregulierung zwei von den Unternehmen jeweils selbst entwickelte Anwendungen, die wegen ihrer gemeinsamen fachlichen Domäne eine große Schnittmenge an Funktionen aufweisen.

Die Schadenanwendungen umfassen alle Funktionen für die Aufnahme und Verwaltung der von Versicherten gemeldeten Schäden. Das sind Funktionen für die Erstellung, Archivierung und Bereitstellung der Korrespondenz. Außerdem ermöglichen die Anwendungen Sachbearbeitern den Zugriff auf die Bestandssysteme mit verschiedenen Informationen zu Verträgen und einer Historie der Schäden, die einem Vertrag zugeordnet sind. In den Applikationen zur Schadensregulierung können anschließende Prozessschritte wie Auszahlungen und Betrugsmanagement ausgelöst werden.

Organisatorisch wurde die Schadensbearbeitung bereits zusammengelegt, d.h. die Sachbearbeiter bearbeiten Kundenvorgänge beider ehemaligen Unternehmen. Dafür müssen sie derzeit noch mit zwei Anwendungen arbeiten. Ziel der Integration ist es, alle benötigten Funktionen und den Zugriff auf alle für die Schadensregulierung relevanten Daten in einer Anwendung zusammenzuführen. Dafür wurden drei Lösungsansätze evaluiert:

- Integration aller benötigten Funktionen in eine dritte, in vielen Teilen neu zu entwickelnde Anwendung, auf Basis der Java Enterprise Plattform.

- Ablösung der Eigenentwicklungen durch Standardsoftware (SAP).
- Integration aller Funktionen in eine der Anwendungen zur Schadensbearbeitung.

Die erste Alternative wurde verworfen, da durch die Integration weiterer Funktionen in die Java-Anwendung erhebliche Performance-Probleme erwartet wurden. Beim Einsatz von SAP wären die Abbildung der Bearbeitungsprozesse und die Verbindung der Standardsoftware mit allen Schnittstellen, die beide Schadensanwendungen aufwiesen, sehr aufwendig gewesen. Außerdem war nicht auszuschließen, dass Änderungen der bewährten Bearbeitungsprozesse für ihre Abbildung in der Standardsoftware notwendig geworden wären. Unter dem Strich erwies sich daher die Anpassung einer der beiden Schadensanwendungen als wirtschaftlichere Lösung. Diese zukünftig führende Anwendung verwendet für ihre Benutzerschnittstelle den ISA Dialog Manager.

Im aktuellen Integrationsprojekt geht es um eine ex-post Integration, bei der eine konkrete, spezifische Lösung entwickelt wird. Mit der beizubehaltenden Schadensanwendung gibt es ein führendes Integrationsobjekt. Die Integrationsfähigkeit der beteiligten Applikationen ist zunächst nur bedingt vorhanden und wird im Wesentlichen im Verlauf des Vorhabens hergestellt. Die Anzahl der Integrationsobjekte ist mit zwei Systemen niedrig, allerdings sind in die Applikationen bereits weitere Systeme integriert (z.B. Dokumentenmanagement, Büroprogramme) bzw. sollen in weiteren Projekten integriert werden. Die Integration erfolgt in zwei Stufen. In einer ersten Stufe wird zunächst ein niedriger Integrationsgrad angestrebt (Kopplung). Ziel der zweiten Stufe ist eine Eingliederung mit hoher Integrationsstärke (siehe Abbildung 2).

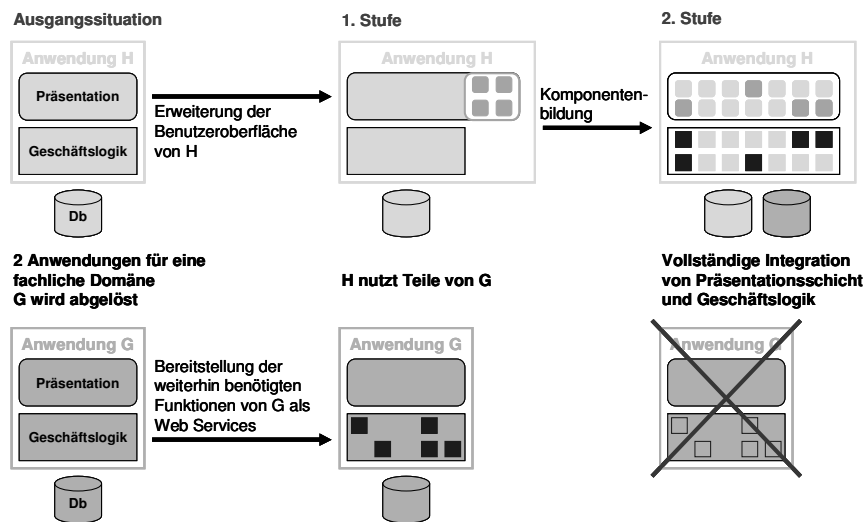


Abbildung 2: Vorgehensweise in einem Vorhaben zur Eingliederung einer Applikation

Im Sinne der oben getroffenen Unterscheidung zwischen leicht- und schwergewichtigen Verfahren wurde ein schwergewichtiges Integrationsverfahren gewählt. Dabei sind umfangreiche Änderungen der beizubehaltenden Schadensanwendung vorgesehen. Für die

Benutzeroberfläche bedeutet dies ebenfalls eine Anpassung und Erweiterung, zum einen um Zugriffsmöglichkeiten auf die neu integrierten Funktionen zu schaffen, zum anderen um Daten darstellen zu können, die für die abzulösende Schadensanwendung konzipiert wurden. Die angepasste Anwendung nutzt die Datenbestände beider ursprünglichen Anwendungen. Eine Integration der Datenbestände der beiden ehemaligen Unternehmen in diesem Bereich ist Gegenstand eines separaten Vorhabens.

Als Zwischenstufe der Integration wurde ein in diesem Fall eher leichtgewichtiges Integrationsverfahren gewählt, bei dem die weiterhin benötigten Funktionen der abzulösenden Schadensanwendung über Web Service Schnittstellen zugänglich gemacht wurden. Die Benutzeroberfläche der weiter zu nutzenden Anwendung wurde so erweitert, dass diese Web Services über sie konsumiert werden können. Bei dieser Zwischenstufe ging es also nur darum, den Zugriff auf bestimmte Funktionen der abzulösenden Anwendung zu ermöglichen. Eine weitergehende Integration, beispielsweise dass Daten aus beiden Anwendungen an der Benutzeroberfläche zu einer einheitlichen, integrierten Darstellung zusammengeführt werden, war in dieser Stufe nicht beabsichtigt.

Das Vorgehen in zwei Stufen macht deutlich, dass mit dem Integrationsvorhaben möglichst schnell Vorteile für die Sachbearbeiter der Schadensbearbeitung erzielt werden sollen. Vor diesem Hintergrund wurde eine agile Vorgehensweise für das Integrationsprojekt als Ganzes gewählt, mit starker Einbeziehung der betroffenen Fachabteilungen während des gesamten Vorhabens und der Möglichkeit, bereits umgesetzte Lösungen möglichst schnell in den produktiven Betrieb übernehmen zu können.

Die Kommunikation mit den Web Services erfolgt über das SOAP-Protokoll. Dafür wird die XML-Schnittstelle des IDM in Verbindung mit einer C-Funktion für die eigentlichen Service-Aufrufe eingesetzt. Die Verbindung zwischen der Funktion für die Service-Aufrufe und den Objekten der Benutzeroberfläche erfolgt über die C/C++ Schnittstelle des IDM. Die XML-Schnittstelle des IDM ist als allgemeine Schnittstelle für die Verarbeitung von XML-Daten konzipiert. Dies hat den Vorteil, dass sie sehr flexibel eingesetzt werden und unterschiedlichste Anforderungen erfüllen kann. Ein Nachteil liegt darin, dass es für Entwickler relativ aufwändig ist, die Schnittstelle für einen XML-Dialekt wie das SOAP-Protokoll zu nutzen.

## **5 Ausblick**

Das Beispiel aus dem Versicherungsunternehmen war ein Anlass zur Weiterentwicklung des IDM und zur Schaffung neuer Schnittstellen. Eine Web Service Schnittstelle auf Basis der vorhandenen XML-Schnittstelle und einer neuen SOAP-Schnittstelle erleichtert die Realisierung von Benutzeroberflächen, die Web Services konsumieren. Außerdem wird mit dem „Service Integrator“ ein Entwicklungswerkzeug ergänzt, der die Erstellung der notwendigen Funktionsdefinitionen und Ereignistypen für das Applikationsobjekt einer IDM-Anwendung unterstützt und teilweise automatisiert. Die Funktionsdefinitionen und Ereignistypen bilden die Schnittstelle zwischen Dialogobjekten und Web Services.

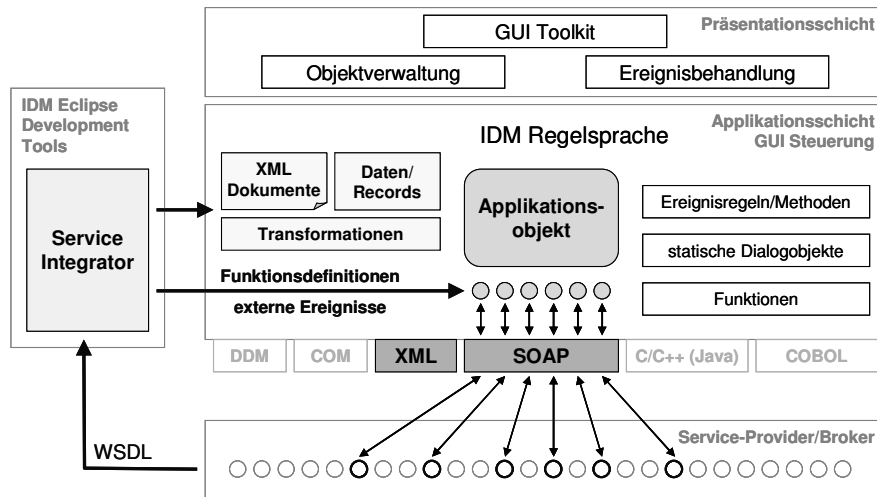


Abbildung 3: Web Service Architektur des ISA Dialog Managers

Wenn eine Benutzeroberfläche selbst in Komponenten (Services) zerlegt wird, benötigt man eine Sprache um diese Komponenten zu orchestrieren. Diese Aufgabe übernimmt derzeit die Skriptsprache des IDM (Regelsprache). Für die Zukunft sind für diesen Zweck Ansätze wie WS-BPEL Extension for People (BPEL4People) interessant, mit denen eine explizite Berücksichtigung von Benutzeraktionen in ausführbaren Beschreibungen von Geschäftsprozessen ermöglicht wird.<sup>1</sup> Die Entwicklung geht dahin, dass für die Orchestrierung der Komponenten zukünftig seltener Spezialisten (Entwickler) erforderlich sind und sie – mit entsprechenden grafischen Editoren – in größerem Umfang durch die jeweiligen Fachanwender durchgeführt werden kann.

Derzeit entstehen Ansätze mit denen vorausschauend günstige Voraussetzungen für eine spätere Integrationsfähigkeit der mit diesen Ansätzen entwickelten Anwendungen geschaffen werden. Um die Integrationsfähigkeit von IDM-Anwendungen zu verbessern, konzipiert die ISA für zukünftige Versionen des IDM eine Interceptor-Schnittstelle in der Laufzeitumgebung, über die andere Anwendungen und Werkzeuge dynamisch auf IDM-Anwendungen zugreifen und Informationen über ihren aktuellen Zustand ermitteln können. Weiterhin wird es über die Interceptor-Schnittstelle möglich sein, interne Ereignisse einer IDM-Anwendung an andere Applikationen weiterzuleiten. Diese Entwicklungen haben das Ziel, leichtgewichtige, minimal-invasive Integrationsverfahren [StTh08] mit dem ISA Dialog Manager zukünftig noch besser zu unterstützen. Im Mittelpunkt des Interesses stehen dabei Integrationsverfahren, für die – im Gegensatz zu den in diesem Beitrag geschilderten Beispielen – keine oder nur stark eingeschränkte Möglichkeiten zum Eingriff in den Quellcode von Applikationen vorhanden sein müssen.

<sup>1</sup> Wir danken Wolfgang Beinbauer (Fraunhofer-Institut für Arbeitswirtschaft und Organisation), der die Anregung zu dieser Überlegung gab.

## Literaturverzeichnis

- [Buf07] Buffone, B.: *Enterprise Web 2.0 Solves the Last Mile of SOA* (posted on June 2<sup>nd</sup>, 2007). Online: [soa.sys-con.com/node/366878](http://soa.sys-con.com/node/366878), abgerufen am 18.06.2009.
- [Fae00] Faehnrich, K.-P.: *Methoden und Werkzeuge zur softwareergonomischen Entwicklung von Informationssystemen* (Habilitationsschrift). Heimsheim; 2000.
- [McK07] McKendrick, J.: *AJAX and Enterprise 2.0 for that 'last mile' of SOA* (posted on May 4<sup>th</sup>, 2007). Online: [blogs.zdnet.com/service-oriented/?p=870](http://blogs.zdnet.com/service-oriented/?p=870), abgerufen am 18.06.2009.
- [Rau93] Rautenstrauch, C.: *Integration Engineering – Konzeption, Entwicklung und Einsatz integrierter Softwaresysteme*. Bonn; 1993.
- [SoZa92] Sowa, J. F.; Zachman, J. A.: *Extending and formalizing the framework for information systems architecture*. IBM Systems Journal 31(3); pp. 590–616; 1992.
- [StTh08] Stefan, F.; Thränert, M.: *Minimal-invasive Integration von Anwendungssystemen*. In: Fähnrich, K.-P.; Kühne, S.; Thränert, M. (Hrsg.): *Model-Driven Integration Engineering*; S.263–275. Leipzig; 2008.
- [Thr05] Thränert, M.: *Integration – Eine Begriffsbestimmung*. In: Fähnrich, K.-P.; Thränert, M.; Wetzel, P. (Hrsg.): *Umsetzung von kooperativen Geschäftsprozessen auf eine internet-basierte IT-Struktur*; S. 11–22. Leipzig; 2005.
- [Thr07] Thränert, M.: *Integration und Integration Engineering – Begriffsbestimmungen*. In: Fähnrich, K.-P.; Thränert, M.; Wetzel, P. (Hrsg.): *Integration Engineering; Motivation – Begriffe – Methoden – Anwendungsfälle*; S. 9–20. Leipzig; 2007.
- [Wik08a] Wikipedia-Autoren: *Integration (Software)*. Stand: 14.12.2008. In: Wikipedia, Die freie Enzyklopädie (Hrsg.). Online: [de.wikipedia.org/w/index.php?title=Integration\\_\(Software\)&oldid=54125754](http://de.wikipedia.org/w/index.php?title=Integration_(Software)&oldid=54125754), abgerufen am 19.06.2009.
- [Wik08b] Wikipedia-Autoren: *Kopplung (Softwareentwicklung)*. Stand: 31.10.2008. In: Wikipedia, Die freie Enzyklopädie (Hrsg.). Online: [de.wikipedia.org/w/index.php?title=Kopplung\\_\(Softwareentwicklung\)&oldid=52447153](http://de.wikipedia.org/w/index.php?title=Kopplung_(Softwareentwicklung)&oldid=52447153), abgerufen am 19.06.2009.