

LDVZ – Nachrichten

Herausgeber:
Landesamt
für Datenverarbeitung und Statistik
Nordrhein-Westfalen

Redaktion:
Bianca Klose,
Alfons Koegel

Kontakt:
Landesamt für Datenverarbeitung
und Statistik NRW
Postfach 10 11 05
40002 Düsseldorf,
Mauerstraße 51
40476 Düsseldorf

Telefon:
0211 9449-01
Telefax:
0211 442006
Internet:
<http://www.lds.nrw.de>
E-Mail:
poststelle@lds.nrw.de

Auflage:
1 200

Druck:
Wölfer Druck Verlag Werbeservice,
Haan

© Landesamt
für Datenverarbeitung
und Statistik NRW,
Düsseldorf, 2003
Vervielfältigung und Verbreitung,
auch auszugsweise, mit Quellen-
angabe gestattet.

Bestell-Nr. Z 09 1 2003 51

ISSN 1616-377X

... in Kürze

ISO-9001-Zertifizierung
des LDS NRW verlängert **2**

Verwendung von Glossaren **2**

Lehrer-Einstellung Online
Einzug des E-Governments
in ein Schlüsselverfahren
der Verwaltung **4**

Entwicklung einer Anwendung
zur Fehlerbehandlung bei
Abfallbegleitscheinen **5**

Weiterentwicklung
des V-Modells **6**

Schwerpunktthemen

GeoServer
der Landesverwaltung NRW
nimmt Produktion auf **9**

Angebot
zentraler GIS-Dienste
mit dem GeoServer **12**

Landtags-
informationssystem NRW **17**

Microsofts .Net-Technologie
– wohin geht die
Programmierung? **20**

Programmierung mit dem
VisualAge Generator
im Projekt OBELIX **26**

Webhosting-Dienstleistungen
des LDS NRW **35**

ISO-9001-Zertifizierung des LDS NRW verlängert

Nachdem das Qualitätsmanagementsystem (QM-System) im Landesamt für Datenverarbeitung und Statistik (LDS NRW) für bestimmte Bereiche der Datenverarbeitung bereits vor drei Jahren eingeführt worden war, erfolgte nun die erfolgreiche Verlängerung des Zertifikates mit gleichzeitiger Ausweitung des Geltungsbereiches.

Qualitätsmanagement (QM) ist durch die Normenfamilie ISO 9000 in Wirtschaft und Verwaltung zu einem bekannten Begriff geworden. Die Umsetzung der Forderungen dieser Normenfamilie soll die Wettbewerbsfähigkeit durch eine hohe Qualität der Produkte und ein günstiges Preis-Leistungs-Verhältnis sichern helfen. Viele Unternehmen stellen sich heute diesen Normen und lassen sich von Zertifizierungsunternehmen bestätigen, dass sie die Anforderungen eines Qualitätsmanagements erfüllen.

Um ihren Aufgaben als IT-Kompetenz- und Dienstleistungszentrum der Landesverwaltung NRW auch in Zukunft gerecht zu werden, hatte sich auch die Landesdatenverarbeitungszentrale (LDVZ) im LDS NRW bereits 1999 dazu entschlossen, ihr QM-System zu implementieren und nach DIN EN ISO 9001 zertifizieren zu lassen. Der Geltungsbereich des QM-Systems sollte zunächst den Geschäftsprozess „Anwendungsbereitstellung“ umfassen und danach sukzessive auf die weiteren Geschäftsprozesse der LDVZ erweitert werden.

Angestrebtes Ziel des Aufbaus und der Weiterentwicklung des QM-Systems sowie der Zertifizierung ist insbesondere die Entwicklung von Produkten in professioneller Qualität und die

kontinuierliche und konsequente Verbesserung der Produktions- und Dienstleistungsprozesse im Hinblick auf

- Funktionalität,
- Zuverlässigkeit,
- Benutzbarkeit,
- Änderbarkeit,
- Wirtschaftlichkeit.

Nachdem das QM-System eingeführt worden war, wurde im Sommer 1999 durch das erfolgreiche dreitägige Zertifikataudit die DIN EN ISO 9001-Konformität des QM-Systems der LDVZ bestätigt und das Zertifikat ausgestellt. Seither gehört auch das LDS NRW zu den Institutionen der Landesverwaltung, die ein nach der DIN EN ISO 9001 zertifiziertes QM-System betreiben.

Als nächster Geschäftsprozess wurde die „IT-Aus- und Fortbildung“ in das QM-System eingebunden und im Jahr 2000 ebenfalls zertifiziert.

Im Jahr 2002 erfolgte die Ausweitung des QM-Systems auf die Geschäftsprozesse „IT-Produktion“ und „IT-Beschaffung“. Im Rahmen der wegen Ablauf des nur jeweils für drei Jahre gültigen Zertifikates notwendig gewordenen erfolgreichen Neu-Zertifizierung wurden die beiden genannten Geschäftsprozesse ebenfalls zertifiziert.

In einem weiteren Schritt ist vorgesehen, den Geschäftsprozess „IT-Infrastruktur“ in das QM-System einzubeziehen und im Jahr 2003 zertifizieren zu lassen.

Eine unabdingbare Voraussetzung sowohl bei der Einführung als auch bei der Weiterentwicklung eines QM-Systems ist eine angemessene Beteiligung der Beschäftigten durch:

- Transparente Darstellung der Managemententscheidung
- Mitwirkung der Beschäftigten bei der Entwicklung und Festlegung der Maßnahmen
- Offene Informationspolitik
- Vertrauensvolle Zusammenarbeit in Teams

Nur dadurch sind eine entsprechende Identifikation, Akzeptanz und Motivation bei den Beschäftigten zu erreichen. Der Erfolg dieser Maßnahmen ist bereits heute in der LDVZ sichtbar.

Dr. Joachim Möhring
Qualitätsmanagementbeauftragter
der LDVZ im LDS NRW
Telefon: 0211 9449-3492
E-Mail: moehring@lds.nrw.de

Verwendung von Glossaren

Immer wieder erlebt man, dass in Projekten Probleme dadurch entstehen, dass Unklarheiten hinsichtlich der verwendeten Begriffswelt bestehen und nicht ausgeräumt werden:

Schließt der Begriff „Produkt“ in einem Marketing-System nun den Begriff „Dienstleistung“ mit ein oder sind „Produkt“ und „Dienstleistung“ zwei Begriffe, die auf der selben Ebene nebeneinander stehen?

Betrachten wir als „Kunde“ nur diejenigen, die etwas bei uns kaufen oder gehören hierzu z. B. auch Personen, die kostenlose Informationen anfordern?

Gehört zu einer „Adresse“ auch die Fax-Nummer und die E-Mail-Angabe?

Bezieht „Name“ auch Adelstitel und akademische Grade mit ein?

Verstehen wir unter „Achsentfernung“ den Ausbau der Achsen oder den Abstand zwischen den Achsen?

Auf solche Begriffe wird z. B. bei der Definition von Anforderungen an ein zu entwickelndes IT-System zurückgegriffen. Da findet man dann Aussagen, wie „Für alle Kunden sind Name, Adresse, ... festzuhalten“. Eine solche Anforderung macht natürlich nur dann Sinn, wenn die Beteiligten – also insbesondere Auftraggeber und Auftragnehmer – ein gemeinsames Verständnis über die verwendeten Begriffe haben.

Hier ist ein Glossar, in dem alle vorkommenden Begriffe mit entsprechenden Begriffsbestimmungen eingetragen werden, von großem Nutzen. Mit dem Führen eines solchen Glossars sollte möglichst frühzeitig in einem Projekt begonnen werden, um einen ‚Begriffswirrwarr‘ gar nicht erst entstehen zu lassen und eine Stelle zu haben, an der neue Begriffe und deren Erklärung niedergelegt werden.

Beispiel-Glossar für eine Bankanwendung (Ausschnitt)

Kunde

Eine Person, die ein Konto [→Konto] oder mehrere Konten bei einer Mitgliedsbank [→ Mitgliedsbank] des XY-Bankenverbunds hat.

Kundenbank

Diejenige Mitgliedsbank [→ Mitgliedsbank], die die Bankkarte [→Bankkarte] an den Kunden [→Kunde] herausgegeben hat und bei der der Kunde ein Konto [→ Konto] oder mehrere Konten hat.

Konto

Dienst, der von einer Mitgliedsbank [→ Mitgliedsbank] bereitgestellt wird, um die Geldgeschäfte eines Kunden [→ Kunde] abzuwickeln. Zu jedem Konto gehört eine eindeutige Kontonummer. ...

Bankkarte

Ein physisches Identifikationsmedium ...

Ziel ist es also, durch ein Glossar die Begriffswelt zumindest projektweit konsistent zu halten. Es soll dazu beitragen, dass Begriffe eindeutig verwendet werden und Missverständnisse vermieden werden. Insofern ist es besonders wichtig, solche Begriffe aufzunehmen, die potentiell Anlass für Mehrdeutigkeiten bieten können. So hat sich immer wieder gezeigt, dass ein und dieselbe Bezeichnungen von den Software-Entwicklern für etwas ganz anderes verwendet wird als von den am Projekt beteiligten Personen aus der Fachabteilung. So versteht etwa ein Software-Entwickler unter „Produkt“ die Ergebnisse des Anwendungsbereitstellungsprozesses (also irgendwelche Entwurfsdokumente oder Software-Bausteine), während der Fachabteilungs-Vertreter damit die vermarkteten Statistiken bezeichnet. Auch zeigt die Erfahrungen, dass bei der Beteiligung mehrerer Personen/Institutionen auf Seiten der Auftraggeber unter diesen keine einheitlichen Begriffsvorstellungen vorliegen, selbst dann, wenn man meinen könnte, die Fachaufgabe wäre durch gesetzliche Regelungen relativ klar vorstrukturiert. Dies liegt u. a. daran, dass sich über die Jahre bei den einzelnen Institutionen unterschiedliche Vorgehensweise mit jeweils eigenen Begriffswelten herausgebildet haben. Auch beim Reengineering existierender IT-Verfahren bietet sich der Einsatz eines Glossars an, um die verwendete Terminologie zu durchdringen.

Man kann das Glossar als Sammlung der für das IT-System relevanten Schlüsselkonzepte (key concepts) ansehen und es dann dazu verwenden, zu überprüfen, ob alle darin aufgeführten Schlüsselkonzepte bei der Spezifikation des zu entwickelnden Systems entsprechend umgesetzt wurden. Dies bedeutet bei einer objekt-orientierten Vorgehensweise typischerweise, dass die modellierten Anwendungsfälle (use cases) alle vorgesehenen Schlüsselkonzepte abdecken.

Die Pflege des Glossar sollte durch den Analytiker erfolgen, wobei allen Projektbeteiligten die Möglichkeit gegeben sein muss, Beiträge zum Glossar beizusteuern. Bevor Begriffe und deren Definitionen in das Glossar aufgenommen wird, sollte allerdings das Einverständnis aller Beteiligten vorliegen.

Heutzutage wird wohl jeder ein solches Glossar elektronisch führen. Zunächst könnte man hier an ein einfaches Word-Dokument denken. Bei näherem Hinsehen zeigt sich jedoch, dass ein gutes Glossar über Querverweise zwischen den einzelnen Begriffen verfügen sollte (siehe „→“ im Beispiel). Es bietet sich daher an, dass man solchen Verweisen ‚auf Knopfdruck‘ nachgehen kann. Die eine Möglichkeit hierzu ist die Ausgestaltung des Glossars als Hypertext, d. h. über einen Hyperlink gelangt man zu dem Begriff, auf den verwiesen wird. Hierbei sollte allerdings beachtet werden, dass sich bei größeren Glossaren die Pflege nur mit Werkzeugunterstützung gewährleisten lässt.

Eine andere Möglichkeit besteht darin, für die Verwaltung der Glossareinträge ein Werkzeug zu verwenden, dass die Einträge in einer Datenbank ablegt. Hierfür kommt z. B. das Anforderungsmanagement-Werkzeug „RequisitePro“ der Fa. Rational in Betracht, in dem bereits ein Anforderungstyp für den Aufbau von Glossaren vordefiniert ist.

Dadurch, dass Glossareinträge untereinander in Beziehung gesetzt werden können, sind Verweise zwischen den einzelnen Einträgen realisierbar. Es ist dann auch erkennbar, ob sich an einem der beteiligten Begriffe etwas geändert hat und somit der Verweis dahingehend zu überprüfen ist, ob er noch weiterhin stimmig ist. Ferner können auch Beziehungen zwischen den Glossareinträgen und anderen im Werkzeug abgelegten Anforderungen hergestellt werden.

Zusammenfassend lässt sich sagen:

- Glossare können einen wichtigen Beitrag zum Gelingen von Projekten leisten;
- Glossare lassen sich mit vertretbarem Aufwand realisieren;
- zur Realisierung von Glossaren existieren hilfreiche Werkzeuge.

Ulrich von Hagen

Telefon: 0211 9449-6706

E-Mail: ulrich.von-hagen@lds.nrw.de

Lehrer-Einstellung Online

Einzug des E-Governments in ein Schlüsselverfahren der Verwaltung

Alljährlich werden in NRW mehrere tausend Einstellungen in den Schuldienst vorgenommen. Bereits recht früh gab es Bestrebungen, dies über das Internet abzubilden. Den Anfang machten statische Seiten mit den schulscharfen Ausschreibungen je Bezirksregierung und den zugehörigen Informationstexten und Bewerbungsformularen als Download. Die Veröffentlichung der Stellenausschreibungen fand in den Internetauftritten der jeweiligen Bezirksregierungen statt, dabei wurden jeweils nur die Stellen veröffentlicht, die sich im entsprechenden Regierungsbezirk befanden. Eine landesweite Suche war somit nicht möglich, was immer dann problematisch war, wenn ein Bewerber eine Stelle im Grenzgebiet von Regierungsbezirken suchte. Dieser Bewerber musste die Internetauftritte aller beteiligten Bezirksregierungen durchsuchen. Dabei konnte er nur eine

„Textsuche“ ausführen, eine elektronische Bewerbung war zu dieser Zeit noch nicht möglich. Es gab zwar den Versuch, die Änderung der Bewerberdaten elektronisch zu ermöglichen, dieser setzte aber eine elektronische Signatur voraus. Da der Aufwand, um eine solche Signatur zu erhalten, für den Bewerber erheblich war, gab es keine nennenswerte Nutzung.

Im Rahmen der Arbeiten zum Bildungsportal wurde das LDS NRW Anfang 2002 damit beauftragt eine E-Governmentanwendung zu erstellen, die sowohl eine landesweite parameterisierbare Suche nach ausgeschriebenen Stellen als auch eine Onlinebewerbung ermöglichen sollte. Quelle der ausgeschriebenen Stellen und Ziel für die Bewerberdaten sollte das existierende DV-Verfahren zur Lehrereinstellung (LEV) sein. Zusammen mit dem Ministerium für Schule, Jugend und Kinder (damals MSWF, heute MSJK) und allen Bezirksregierungen wurden die erforderlichen Funktionalitäten definiert:

- Informationen und Gesetzestexte als strukturiertes, statisches Angebot
- Suchmaschinen für die verschiedenen Bewerbertypen (vollausgebildete Lehrkräfte, Seiteneinsteiger, ...)
- „Warenkorb“ für die Vormerkung von Bewerbungen
- Onlinebewerbung
- Ergänzung von Bewerbungen
- Übersicht über den Buchungsstand einer Bewerbung

Sehr schnell wurde auch klar, dass es ein abgegrenztes Onlineverfahren für die Schulen geben musste, mit dem diese die Ausschreibungstexte der jeweiligen Stellen erfassen konnten. Dies wurde als INES (Internetbasierte Eingabe von Stellen) durch das LDS NRW eigenständig realisiert.

Die Realisierung der strukturierten statischen Informationen wurde mittels Contentmanagementsystem vom MSJK eigenständig vorgenommen.

Die Suchmaschinen wurden durch das LDS NRW so realisiert, dass sowohl nach fachlichen (Fächerkombinationen oder Schulform) als auch nach geographischen Kriterien gesucht werden kann. Die Datenbasis für die Suchmaschinen ist eine Datenbank im Internet, die automatisch einmal pro Nacht aus dem Landesverwaltungsnetz (LVN) heraus aktualisiert wird.

Die Erfahrungen mit dem signaturbasierten Änderungsverfahren waren ausschlaggebend dafür, dass bei der Realisierung der Onlinebewerbung (und bei der Ergänzung der Bewerbung) auf die elektronische Unterschrift des Bewerbers verzichtet wurde. Es hatte sich klar gezeigt, dass kaum ein Nutzer bereit war, sich für ein einzelnes Verfahren eine elektronische Unterschrift zu beschaffen. Dennoch musste ein Weg gefunden werden, der eine Authentifizierung des Bewerbers ermöglichte. Hierzu wurde ausgenutzt, dass ein Bewerber auf jeden Fall Unterlagen auf Papier einsenden muss (Zeugnis kopien, ...). Somit gilt im Lehrereinstellungsverfahren, dass eine Erstbewerbung aus der Lehrereinstellung Online (LEO) erst dann übernommen wird, wenn die Papierunterlagen des Bewerbers eingetroffen sind und diese auch einer Überprüfung standhalten.

Damit die Daten des Bewerbers selbst geschützt sind, erfolgt der Austausch von Daten in LEO zwischen Bewerber und Server ausschließlich unter der SSL (Secure Socket Layer). Auf dem Server werden die Bewerberdaten umgehend mit einem asynchronen Verschlüsselungsverfahren mittels eines öffentlichen Schlüssels des LDS NRW verschlüsselt. Die Übertragung der verschlüsselten Bewerberdaten vom LEO-Server im Internet in das LVN erfolgt mittels E-Mail mit Anhang.

Bei der Erstbewerbung erhält der Bewerber eine Bewerbernummer und muss ein Kennwort wählen. Mit diesen Zugangsdaten kann sich der Bewerber den Stand seiner Buchungen ansehen und Änderungen seiner Daten durchführen (z. B. neue Wohnanschrift). Da ein echter Schutz von Daten im Internet nicht garantiert werden kann, werden die Bewerberdaten unter LEO nicht vorgehalten. Bewerber, die ihre bisherigen Daten ändern wollen, müssen die Änderung also „blind“ vornehmen.

Zum Zeitpunkt der Erstellung dieses Artikels wurden bereits zwei Einstellungsverfahren über LEO abgewickelt. Zusammen erfolgten in beiden Verfahren ca. 10 000 Buchungen (Erstbewerbungen, weitere Bewerbungen und Änderungen). In den jeweiligen Ausschreibungszeiträumen dominierten die Seiten von LEO klar die Aufrufstatistiken des Bildungsportals (in der Spitze bis zu 3 000 000 Hits gegenüber 300 000 Hits).

Für die Zukunft ergibt sich zum einen die Möglichkeit, artgleiche Verfahren ebenso zu lösen. Zum anderen wird auch LEO selbst weiterentwickelt werden. Hierbei wird ein besonderes Gewicht darauf gelegt werden, die Wünsche von Nutzern zu integrieren. Zusätzlich muss sorgfältig beobachtet werden, ob und wann eine höhere Verbreitung von elektronischen Signaturen gegeben ist, um diese ggfs. auch in LEO nutzen zu können.

Markus Guhl
Telefon: 0211 9449-2578
E-Mail: markus.guhl@lds.nrw.de

Entwicklung einer Anwendung zur Fehlerbehandlung bei Abfallbegleitscheinen

AIDA – AbfallInformationsDaten-drehscheibe – ist ein Projekt des Landesumweltamtes Nordrhein-Westfalen (LUA) zur Zusammenführung von abfallwirtschaftlichen Daten verschiedener Quellsysteme und deren Präsentation im Intranet. Im Rahmen dieses Projektes wurde im vergangenen Jahr bereits das Modul „Mengenrecherche“ realisiert, eine Weboberfläche zur Abfrage von Abfallentsorgungsmengen aus unterschiedlichen Datenbankquellen.

Die Akzeptanz von Informationssystemen dieser Art steht und fällt mit der Datenqualität und -aktualität. Zur Verbesserung der Qualität und Aktualität einer der Quelldatenbanken von AIDA – des Abfallnachweissystems ASYS – wurde im LDS NRW jetzt mit der Neuentwicklung der Fehlerbehandlung bei Abfallbegleitscheinen begonnen.

In ASYS werden so genannte Abfallbegleitscheine, die bei der Entsorgung bestimmter nachweispflichtiger Abfallarten auszustellen sind, zentral im LUA gesammelt und ausgewertet. Wegen der großen Anzahl der in NRW anfallenden Abfallbegleitscheine werden diese nicht im LUA selbst erfasst, sondern es wird das LDS NRW als Dienstleister in Anspruch genommen. Dazu werden die Daten im LDS NRW erfasst, Begleitscheine eingescannt und dem LUA als .TIFF-Datei zur Verfügung gestellt.

Nach der Erfassung müssen die Daten im LUA nachbearbeitet werden, da sie neben Erfassungsfehlern vor allem Fehler aufgrund falsch oder unvollständig ausgefüllter Begleitscheine enthalten. Die Nachbearbeitung erfolgt bisher mehrstufig: Zunächst werden die Daten stapelweise – ca. 2 000 Datensätze gleichzeitig – in eine Ac-

cess-Datenbank geladen und dort durch vorher auszuwählende Fehler-routinen auf das Vorkommen so genannter Massenfehler geprüft und soweit möglich verbessert. Anschließend werden die Daten nach ASYS zurückgeschrieben und verbliebene Fehler anhand ausgedruckter Fehlerlisten durch den zuständigen Sachbearbeiter mit Hilfe der ASYS-Anwendung nachkorrigiert. Die nun vollständig berichtigten Daten werden wiederum in die Access-Datenbank geladen und mit den ursprünglich erfassten Daten verglichen. Daraufhin werden Fehlerberichte für die zuständigen Behörden erstellt.

Diese Vorgehensweise hat erkennbar diverse Nachteile: Das Hin- und Herschauen großer Datenmengen zwischen zwei verschiedenen Datenbanksystemen ist aufwändig und eine zusätzliche Fehlerquelle. Wegen der Verwendung einer lokalen Datenbank kann die so genannte „Massenfehlerbehandlung“ nicht arbeitsteilig durchgeführt werden. Und schließlich sind insgesamt vier Anwendungsprogramme für den gesamten Vorgang nötig – neben den bereits erwähnten Datenbankanwendungen gibt es noch eine Stapelverwaltung, in der der jeweilige Stand der Arbeiten festgehalten wird, sowie ein Programm zur Anzeige der gescannten Belege.

Eine völlig neu zu entwickelnde Anwendung „Fehlerbehandlung Abfallbegleitscheine“ (FebA) soll deshalb die bestehenden Programme in diesem Arbeitsprozess schrittweise ersetzen. Die Anwendung wird in Java entwickelt und greift über JDBC direkt auf die Ingres-Datenbank ASYS zu. Um einen Zwischenstand der in Bearbeitung befindlichen Begleitscheindaten sichern zu können, werden in der ASYS-Datenbank zusätzliche Tabellen für FebA angelegt. Ein Statusflag kennzeichnet fehlerhafte, in Bearbeitung befindliche sowie korrigierte Daten.

Eine besondere Schwierigkeit besteht in der vom LUA vorgegebenen Reihenfolge der Entwicklung und der stufenweisen Inbetriebnahme der neuen Anwendung. Der Arbeitsschritt „Nachkorrektur in ASYS“ sollte wegen schlechter Performance und Arbeitsstau zuerst durch die neue Anwendung FebA ersetzt werden. Deshalb wurde FebA in der ersten Ausbaustufe zwischen die eigentliche Fehlersuche und das Erstellen der Fehlerberichte in der Access-Anwendung „Massenfehlerbehandlung“ gesetzt. Während in ASYS die fehlerhaften Begleitscheine anhand ausgedruckter Listen einzeln aufgerufen werden mussten, stellt FebA eine elektronische Liste zur Verfügung. Zum gleichen Entsorgungsnachweis gehörende andere Begleitscheine können zum Vergleich mit dem fehlerhaften Begleitschein herangezogen werden. Im Unterschied zur ASYS-Anwendung ist es in FebA möglich, auch die Scan-Bilder der Begleitscheine direkt anzuzeigen, was die Bearbeitung schneller und komfortabler macht.

Um gravierende Änderungen an der Altanwendung zu vermeiden, wurde die Schnittstelle zur „Massenfehlerbehandlung“ zunächst mit Hilfe einer Datenbankprozedur realisiert, die beim Zurückschreiben der Daten nach ASYS die FebA-Tabellen füllt. Wegen der unbefriedigenden Performance dieser Lösung ist allerdings geplant, die Anwendung „Massenfehlerbehandlung“ so anzupassen, dass direkt in die FebA-Tabellen geschrieben wird.

In der zweiten Ausbaustufe sollen die Anwendungen „Massenfehlerbehandlung“ und „Stapelverwaltung“ in FebA integriert werden, so dass der gesamte Arbeitsprozess dann in einer Anwendung realisiert ist. Schließlich sollen in einer dritten Ausbaustufe die Informati-

onen aus den Fehlerberichten den zuständigen unteren Abfallwirtschaftsbehörden über eine Web-Anwendung zur Verfügung gestellt werden.

Ilona Lungershausen

Telefon: 0211 9449-2602

E-Mail: ilona.lungershausen@lds.nrw.de

Weiterentwicklung des V-Modells

Das V-Modell 97 (auch bekannt unter dem offiziellen Namen „Entwicklungsstandard für IT-Systeme des Bundes“ (EStdIT)) ist für viele Unternehmen und Behörden mittlerweile eine entscheidende Grundlage für die Organisation und Durchführung von IT-Vorhaben. Es verbessert zum einen die Qualität der erstellten Systeme und unterstützt zum anderen die Zusammenarbeit zwischen Auftraggebern und Auftragnehmern. Für die Landesverwaltung Nordrhein-Westfalen ist in den IT-Richtlinien NRW festgelegt, dass bei der Durchführung von IT-Vorhaben das V-Modell zugrunde gelegt werden soll.

Nach der Fertigstellung des V-Modells im Jahre 1997 hat es allerdings keine Fortschreibung gegeben, so dass das V-Modell nicht in allen Bereichen den aktuellen Stand der Informationstechnologie widerspiegelt. Auch die beim Einsatz des V-Modell 97 zwischenzeitlich gesammelten umfangreichen Erfahrungen sollten im Rahmen einer Weiterentwicklung zur Verbesserung des Vorgehensmodells genutzt werden. Um eine grundlegendere Überarbeitung des V-Modells zu ermöglichen, wurde die TU München im letzten Jahr mit der Weiterentwicklung des V-Modells beauftragt. Am V-Modell-Weiterentwick-

lungsprojekt WEIT, das von Prof. Dr. Manfred Broy vom Lehrstuhl für Software & Systems Engineering geleitet wird, nehmen als Industriepartner die Firmen EADS, IABG und Siemens teil.

Ausgehend von den erkannten Verbesserungspotenzialen und Weiterentwicklungsmöglichkeiten wurden folgende Ziele für ein „V-Modell 200x“ formuliert¹⁾:

- bessere Unterstützung der Anpassbarkeit, Anwendbarkeit, Skalierbarkeit sowie Änder- und Erweiterbarkeit des V-Modells
- Erweiterung des Anwendungsbereiches und Ausweitung zum allgemeinen Systementwicklungsstandard
- Aktualisierung sowie technologische, methodische Weiterentwicklung (Inhalte auf den neuesten Stand der Technologie bringen)
- Anpassung an aktuelle (Quasi-)Standards, Vorschriften und Normen

Bei der Ausgestaltung des V-Modell 200x sind im Projekt WEIT verschiedene Vorgaben bzw. Rahmenbedingungen zu berücksichtigen:

- Die bisherige Philosophie und Grundstruktur des V-Modell 97 soll im V-Modell 200x erkennbar bleiben.
- Das V-Modell 200x soll bei den referenzierten Verfahren und Methoden den aktuellen Stand der Technik berücksichtigen, in Bezug auf die Werkzeugunterstützung aber herstellerunabhängig sein. Eine Fortschreibung der bisherigen eigenständigen Methoden- und Werkzeugstandards soll hingegen nicht erfolgen.
- Die Komplexität des V-Modell 200x soll so gering wie möglich gehalten werden. Die Qualitätseigenschaften, wie Anpassbarkeit, Anwendbarkeit, Skalierbarkeit sowie

1) Die Darstellung der Ziele sowie der Vorgaben bzw. Rahmenbedingungen orientiert sich an der Präsentation zum Thema „Weiterentwicklung des EStdIT auf der Basis des V-Modell 97“, die Herrn Saas (BWB) auf dem Erfahrungsaustausch 2002 des V-Modell-Anwender-Vereins ANSSTAND e. V. gehalten hat (<http://www.ansstand.de/aendkonf/e2002v4.pdf>)

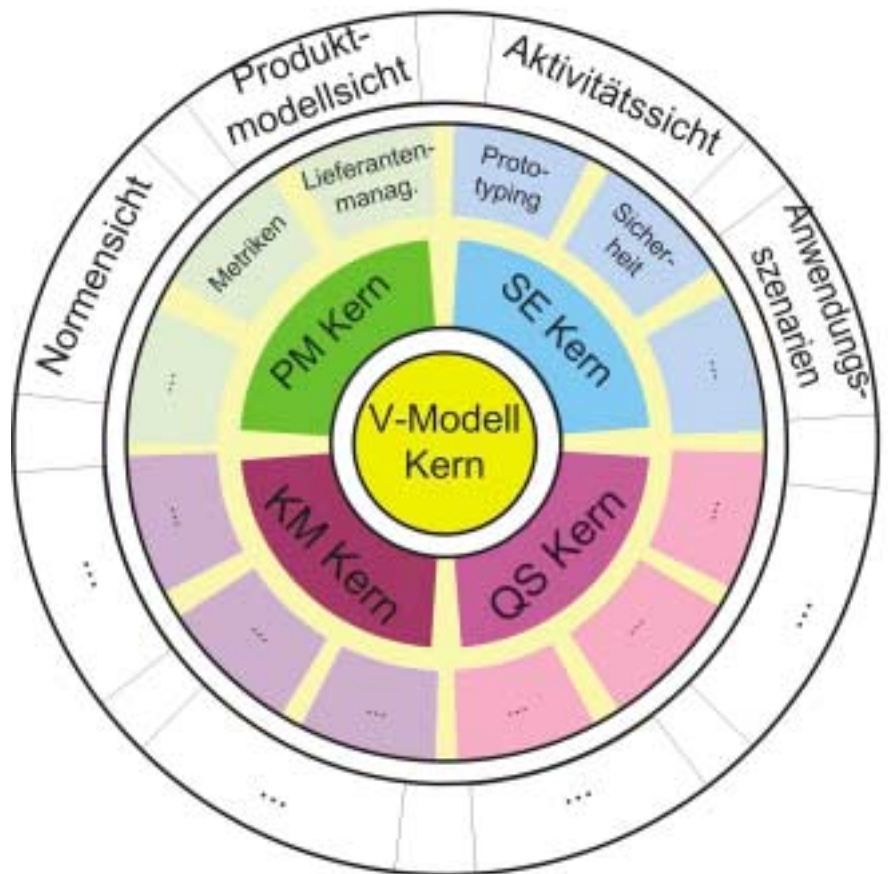
Änder- und Erweiterbarkeit des V-Modell 200x sollen verbessert werden, um insbesondere eine größere Akzeptanz auch bei kleinen und mittleren IT-Vorhaben zu erreichen.

- Das V-Modell 200x soll eine modulare Struktur haben und die Trennung unabhängiger Aspekte beinhalten.
- Das V-Modell 200x soll eine Werkzeug- und Workflow-Unterstützung prinzipiell ermöglichen.
- Die Bereitstellung einer Plattform für die Veröffentlichung im Internet, Anwendung, Pflege und Wartung sowie Erweiterung des V-Modell 200x soll unterstützt werden.
- Eine deutsche und eine englische Version des V-Modell 200x sollen bereitgestellt werden.

Einen guten Überblick über inhaltliche Aspekte der geplanten Weiterentwicklung gibt die Präsentation "Anforderungen an das V-Modell 200x", die Prof. Broy (TU München) auf dem Erfahrungsaustausch 2002 des V-Modell-Anwenderverei ANSSTAND e. V. gehalten hat (siehe <http://www.ansstand.de/aendkonf/e2002v5.pdf>).

Um die Anwendbarkeit des V-Modells zu erhöhen, soll das V-Modell 200x nach einem „Baukastenprinzip“ aufgebaut sein, d. h. basierend auf einfachen, modularen Grundstrukturen mit entsprechenden Beschreibungselementen, wie zum Beispiel Aktivitäten, Produkten und Rollen wird ein V-Modell-Kern zur Verfügung gestellt, der je nach Organisation bzw. Projekt um zusätzliche vordefinierte V-Modell-Vorgehensbausteine, die ebenfalls aus Beschreibungen der Aktivitäten, Produkte und Rollen bestehen, erweitert werden kann.

Darüber hinaus sollen verschieden strukturierte Sichten zur Verfügung gestellt werden, die den Einstieg in das V-Modell 200x erleichtern. Durch dieses Sichtenkonzept ist das V-Modell im Hinblick auf verschiedene Entwicklungsansätze (z. B. traditionelles Wasserfallmodell, Evolutionäre Entwicklung) anwendbar und es werden unterschiedliche Herange-



hensweisen (also z. B. ausgehend von den einzelnen Rollen (Projektleiter, Systemanalytiker usw.)) ermöglicht.

Diese Lösungsvision wird in der nebenstehenden Abbildung verdeutlicht, die der Präsentation von Prof. Broy entnommen wurde.

Die vorgesehene Projektabwicklung gliedert sich in drei Phasen. In der ersten Phase werden die inhaltlichen Verbesserungspotenziale des V-Modells analysiert und eine verbesserte Struktur des V-Modell 200x erarbeitet und anhand einer Fallstudie evaluiert. Diese Ergebnisse fließen in ein Strategiepapier und in eine Vorhabensbeschreibung für die zweite Phase ein. Diese zweite Phase befasst sich mit den konkreten Überarbeitungen und Erweiterungen des V-Modells. In der abschließenden dritten Phase wird das neue V-Modell publiziert, verbreitet und der Öffentlichkeit zugänglich gemacht. Hierbei sollen verstärkt moderne Darstellungs- und Verbreitungsformen (insbesondere mittels Internet) zur Anwendung kommen.

Das Projekt hat Ende letzten Jahres begonnen und befindet sich derzeit in der Phase 1. Das inhaltliche Verbesserungspotenzial des V-Modells wurde inzwischen analysiert und die sich daraus ergebenden Anforderungen an die V-Modell-Weiterentwicklung wurden in einem Anforderungsanalysedokument zusammengetragen. Z. Zt. erfolgt die Priorisierung der Anforderungen sowie die Erarbeitung einer modularen Grobstruktur, die letztendlich dazu beitragen soll, dass das V-Modell deutlich leichter anzuwenden und anpassbar ist.

Zusätzliche Informationen zur Weiterentwicklung des V-Modells erhalten Sie beim Autor des Artikels, der sowohl als Mitglied der V-Modell-Änderungskonferenz als auch als Reviewer beim V-Modell-Weiterentwicklungsprojekt aktiv an der Weiterentwicklung dieses Vorgehensmodells mitwirkt.

Ulrich von Hagen
Telefon: 0211 9449-6706
E-Mail: ulrich.von-hagen@lds.nrw.de

**Windows
Adress-,
Veranstaltungs- und
Einladungsverwaltung**

WAVE

*Die Software
zur Planung und
Durchführung von
Großveranstaltungen*

Landesamt für
Datenverarbeitung
und Statistik NRW



Was sind die Ziele
von **WAVE**?

Wave unterstützt den gesamten Prozess der Veranstaltungsplanung von der Einladung bis zur Einlasskontrolle.

WAVE bietet deshalb:

- Zeitersparnis bei der Planung großer Veranstaltungen
- Unterstützung bei der Ermittlung von Teilnehmern
- Vermeidung doppelter Einladungen
- Einlasskontrolle
- Zentrale Datenhaltung

Systemumgebung

Windows2000/NT/XP
Access2000-Applikation mit
Jet-Engine 4.0 oder SQL-Server 2000
Word2000



Ihre **Ansprechpartner** sind:

Frank Walter

☎ 0211 9449 - 6729

✉ frank.walter@lds.nrw.de

Jürgen Sperling

☎ 0211 9449 - 6729

✉ juergen.sperling@lds.nrw.de

NRW.

Mit Installation der GeoServer-Vers. 4.2 wurde die Bereitstellung von Geodaten im Intranet entgeltlich aufgenommen. Da über die Systemarchitektur und die Funktionsweise des GeoServers bereits in den LDVZ-Nachrichten 2/2001 berichtet wurde, liegt der Schwerpunkt dieses Artikels nunmehr auf dem Bereitstellungsverfahren. Abschließend werden die weiteren Entwicklungsmaßnahmen und die Einbindung des GeoServers in die im Aufbau befindliche Geodateninfrastruktur des Landes NRW aufgezeigt.

Geo-/Geobasisdaten-Angebot

Mit dem GeoServer werden die Kartenwerke der Landesverwaltung über einen WEB-Client im Intranet visualisiert. Sie können nach thematischen Gesichtspunkten vorselektiert und miteinander kombiniert werden. Das Angebot des GeoServers erstreckt sich sowohl auf Geobasis- als auch auf Geodaten.

Geodaten

Das Geodaten-Angebot beinhaltet überwiegend Umwelt- und Planungskarten aus den Geschäftsbereichen des Ministeriums für Umwelt und Naturschutz, Landwirtschaft und Verbraucherschutz (MUNLV) und des Ministeriums für Verkehr, Energie und Landesplanung (MVEL). Als beispielhafte Karteninhalte sind hier zu nennen die Natur- und Landschaftsschutzgebiete der Landesanstalt für Ökologie, Bodenordnung und Forsten (LÖBF), die Wasserschutzgebiete und das Gewässernetz des Landesumweltamtes (LUA), der Gebietsentwicklungsplan des MVEL und der Bezirksregierungen sowie die digitale Bodenkarte und die Lagerstättenkarte des Geologischen Dienstes (GD). Diese Kartenwerke werden im Shape-, GIAP-, DXF- und C60-Format abgegeben.

Geobasisdaten

Der Bereich der Geobasisdaten umfasst im Wesentlichen die von der Landesvermessung gepflegten topographischen Kartenwerke TK 25/50/100, ÜK 250/500 in Form von Rasterkarten, die über den GeoServer in unterschiedlichen Formaten bereitgestellt werden. Mit im Angebot ist das vektor-

basierte ATKIS (DLM 25) sowie die daraus abgeleitete DTK 10, deren Übernahme in den GeoServer kurz vor dem Abschluss steht. Die DGK 5 Grundriss und Luftbild ergänzen die Produktpalette.

Bestellworkflow im WEB-Client des INTRANETS

Um die o. g. Geo-/Geobasisdaten über den GeoServer bestellen zu können, bedarf es einer Anmeldung an das Bestellmodul des WEB-Clients. In Bezug auf die Bereitstellung von Rasterkarten wird seitens der GeoServer-Administration angestrebt, dass pro Behörde nur wenige Mitarbeiter/-innen in die Kundendatenbank aufgenommen werden. Hiermit soll der Aufbau mehrerer speicherplatzintensiver Sekundärdatenbestände verhindert werden.

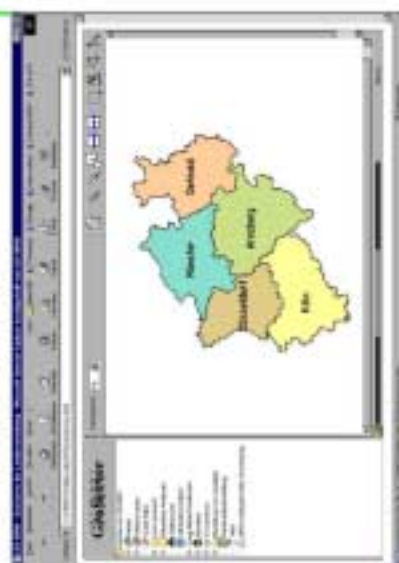
Für den freigeschalteten Personenkreis öffnet sich nach vorheriger Autorisierung (User-ID und Passwort) ein Dialog zur Auswahl von Geo-/Geobasisdaten. Der Anwender kann sowohl Raster- als auch Vektordaten nach räumlichen und fachlichen Kriterien selektieren und sie unter Angabe des gewünschten Ausgabeformats (z. B. Shape, MrSid etc.) und Koordinatensystems (GK 2/3er Streifen) in den Warenkorb geben. Dieser Warenkorb kann weiter ergänzt und für vektorbasierte Grundrissdaten als Bestellvorlage für spätere Aktualisierungsbestellungen genutzt werden.

Die tatsächliche Bestellung der Geo-/Geobasisdaten erfolgt aus dem Warenkorb. In einem weiteren Dialog kann der WEB-Anwender in Abhängigkeit des Datenvolumens die Versandart bzw. das Speichermedium, auf dem ausgeliefert werden soll, auswählen. Ist die Bestellung abgeschickt, so erhält der Nutzer kurzfristig per E-Mail eine Auftragsbestätigung und schließlich mit der Auslieferung der Daten einen entsprechenden Lieferschein. Abb.1 zeigt zusammenfassend den gesamten Workflow für das Bestellverfahren im GeoServer.

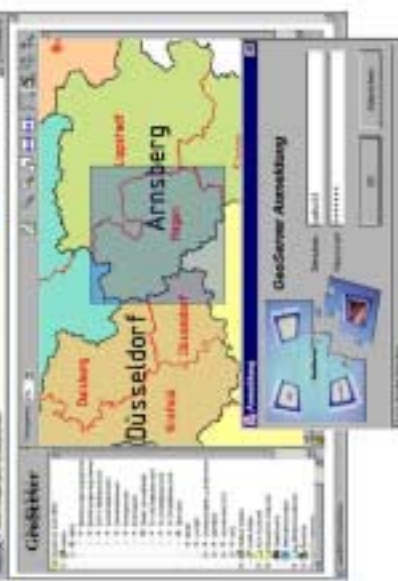
Die Bereitstellung der Geo-/Geobasisdaten erfolgt mit dem GeoServer im Intranet (<http://geoserver.lids.nrw.de>) der Landesverwaltung für Landesbehörden gemäß § 3, § 6 Abs. 2, § 7 Abs. 2 u. § 9 Abs. 2 LOG NRW als auch für Einrichtungen des Landes gemäß § 14 Abs. 2 LOG NRW entgeltfrei.

1. GeoServer Start

GeoServer der Landesverwaltung



2. Gebietsauswahl



3. Produktauswahl



4. Warenkorb - Bestellung



Abb. 1 Bestellworkflow im GeoServer

5. Auftragsbestätigung

LDZ NEW
Grafikzentrum
Rath Chausse
Münster 11
4079 Düsseldorf
Düsseldorf 11.11.2002 14:45:03
Der Auftrag vom 11.11.2002 14:44:56, Der Auftrag-Nr.: 1000, Unser Auftrag-Nr.: 20
AUFTRAGS BESTÄTIGUNG
Ihre Bestellung ist bei uns eingegangen und wird bearbeitet.
Ihre Bestellung wird bei uns folgende Bearbeitung:
Produkt: Produkt 1
Produkt: Produkt 2
Produkt: Produkt 3
Produkt: Produkt 4
Produkt: Produkt 5
Produkt: Produkt 6
Produkt: Produkt 7
Produkt: Produkt 8
Produkt: Produkt 9
Produkt: Produkt 10
Produkt: Produkt 11
Produkt: Produkt 12
Produkt: Produkt 13
Produkt: Produkt 14
Produkt: Produkt 15
Produkt: Produkt 16
Produkt: Produkt 17
Produkt: Produkt 18
Produkt: Produkt 19
Produkt: Produkt 20
Produkt: Produkt 21
Produkt: Produkt 22
Produkt: Produkt 23
Produkt: Produkt 24
Produkt: Produkt 25
Produkt: Produkt 26
Produkt: Produkt 27
Produkt: Produkt 28
Produkt: Produkt 29
Produkt: Produkt 30
Produkt: Produkt 31
Produkt: Produkt 32
Produkt: Produkt 33
Produkt: Produkt 34
Produkt: Produkt 35
Produkt: Produkt 36
Produkt: Produkt 37
Produkt: Produkt 38
Produkt: Produkt 39
Produkt: Produkt 40
Produkt: Produkt 41
Produkt: Produkt 42
Produkt: Produkt 43
Produkt: Produkt 44
Produkt: Produkt 45
Produkt: Produkt 46
Produkt: Produkt 47
Produkt: Produkt 48
Produkt: Produkt 49
Produkt: Produkt 50
Produkt: Produkt 51
Produkt: Produkt 52
Produkt: Produkt 53
Produkt: Produkt 54
Produkt: Produkt 55
Produkt: Produkt 56
Produkt: Produkt 57
Produkt: Produkt 58
Produkt: Produkt 59
Produkt: Produkt 60
Produkt: Produkt 61
Produkt: Produkt 62
Produkt: Produkt 63
Produkt: Produkt 64
Produkt: Produkt 65
Produkt: Produkt 66
Produkt: Produkt 67
Produkt: Produkt 68
Produkt: Produkt 69
Produkt: Produkt 70
Produkt: Produkt 71
Produkt: Produkt 72
Produkt: Produkt 73
Produkt: Produkt 74
Produkt: Produkt 75
Produkt: Produkt 76
Produkt: Produkt 77
Produkt: Produkt 78
Produkt: Produkt 79
Produkt: Produkt 80
Produkt: Produkt 81
Produkt: Produkt 82
Produkt: Produkt 83
Produkt: Produkt 84
Produkt: Produkt 85
Produkt: Produkt 86
Produkt: Produkt 87
Produkt: Produkt 88
Produkt: Produkt 89
Produkt: Produkt 90
Produkt: Produkt 91
Produkt: Produkt 92
Produkt: Produkt 93
Produkt: Produkt 94
Produkt: Produkt 95
Produkt: Produkt 96
Produkt: Produkt 97
Produkt: Produkt 98
Produkt: Produkt 99
Produkt: Produkt 100

6. Auslieferung

LDZ NEW
Grafikzentrum
Rath Chausse
Münster 11
4079 Düsseldorf
Düsseldorf 11.11.2002 14:45:03
Der Auftrag vom 11.11.2002 14:44:56, Der Auftrag-Nr.: 1000, Unser Auftrag-Nr.: 20
AUFTRAGS BESTÄTIGUNG
Ihre Bestellung ist bei uns eingegangen und wird bearbeitet.
Ihre Bestellung wird bei uns folgende Bearbeitung:
Produkt: Produkt 1
Produkt: Produkt 2
Produkt: Produkt 3
Produkt: Produkt 4
Produkt: Produkt 5
Produkt: Produkt 6
Produkt: Produkt 7
Produkt: Produkt 8
Produkt: Produkt 9
Produkt: Produkt 10
Produkt: Produkt 11
Produkt: Produkt 12
Produkt: Produkt 13
Produkt: Produkt 14
Produkt: Produkt 15
Produkt: Produkt 16
Produkt: Produkt 17
Produkt: Produkt 18
Produkt: Produkt 19
Produkt: Produkt 20
Produkt: Produkt 21
Produkt: Produkt 22
Produkt: Produkt 23
Produkt: Produkt 24
Produkt: Produkt 25
Produkt: Produkt 26
Produkt: Produkt 27
Produkt: Produkt 28
Produkt: Produkt 29
Produkt: Produkt 30
Produkt: Produkt 31
Produkt: Produkt 32
Produkt: Produkt 33
Produkt: Produkt 34
Produkt: Produkt 35
Produkt: Produkt 36
Produkt: Produkt 37
Produkt: Produkt 38
Produkt: Produkt 39
Produkt: Produkt 40
Produkt: Produkt 41
Produkt: Produkt 42
Produkt: Produkt 43
Produkt: Produkt 44
Produkt: Produkt 45
Produkt: Produkt 46
Produkt: Produkt 47
Produkt: Produkt 48
Produkt: Produkt 49
Produkt: Produkt 50
Produkt: Produkt 51
Produkt: Produkt 52
Produkt: Produkt 53
Produkt: Produkt 54
Produkt: Produkt 55
Produkt: Produkt 56
Produkt: Produkt 57
Produkt: Produkt 58
Produkt: Produkt 59
Produkt: Produkt 60
Produkt: Produkt 61
Produkt: Produkt 62
Produkt: Produkt 63
Produkt: Produkt 64
Produkt: Produkt 65
Produkt: Produkt 66
Produkt: Produkt 67
Produkt: Produkt 68
Produkt: Produkt 69
Produkt: Produkt 70
Produkt: Produkt 71
Produkt: Produkt 72
Produkt: Produkt 73
Produkt: Produkt 74
Produkt: Produkt 75
Produkt: Produkt 76
Produkt: Produkt 77
Produkt: Produkt 78
Produkt: Produkt 79
Produkt: Produkt 80
Produkt: Produkt 81
Produkt: Produkt 82
Produkt: Produkt 83
Produkt: Produkt 84
Produkt: Produkt 85
Produkt: Produkt 86
Produkt: Produkt 87
Produkt: Produkt 88
Produkt: Produkt 89
Produkt: Produkt 90
Produkt: Produkt 91
Produkt: Produkt 92
Produkt: Produkt 93
Produkt: Produkt 94
Produkt: Produkt 95
Produkt: Produkt 96
Produkt: Produkt 97
Produkt: Produkt 98
Produkt: Produkt 99
Produkt: Produkt 100

Geo-/Geobasisdaten-Vertrieb im INTERNET

Mit Erlass des Innenministeriums vom 11. 12. 2002 wurde das LDS NRW beauftragt, in Ergänzung zu den klassischen Vertriebswegen den Zugang zu den Geobasisdaten der Landesvermessung über den GeoServer im Internet zu ermöglichen. Hierzu werden die noch zu dem bereits existierenden MapServer im Internet fehlenden Selektions- und Bestellkomponenten kurzfristig nachlizensiert und eingerichtet. Um Kartenwerke bepreisen zu können, muss das Bestellmodul um einen WEB-Pricing&Ordering-Service

erweitert werden. Mit diesem Modul sollen verschiedene Nutzungs- und Entgeltmodelle für unterschiedliche Kartenwerke hinterlegt werden können. Perspektivisch ist auch daran gedacht, das Bestellmodul um eine E-Payment-Komponente zu erweitern. Dann wird der Vertriebsweg mittels digitaler Signatur bargeldlos im Online-Betrieb abgewickelt werden können.

Geodateninfrastruktur NRW

Der GeoServer im LDS NRW soll gemäß o. g. Erlass als zentrale Datendrehscheibe Kernelement der Geoda-

teninfrastruktur des Landes sein. Das LDS NRW wurde beauftragt, im Rahmen dieser Initiative des Landes und konform zu den hierin vereinbarten Regelungen den GeoServer kurzfristig und zielgerichtet hinsichtlich eines internet-basierten Vertriebssystems (s. o.) weiterzuentwickeln und im Einvernehmen mit den Datenproduzenten des Landes ein Geo-/Geobasisdatenportal einschließlich der zugehörigen Dienste zur Nutzung der Daten einzurichten sowie den Produktionsbetrieb sicherzustellen.

Stefan Küpper

Telefon: 0211 9449-6317

E-Mail: stefan.kuepper@lds.nrw.de



- Sind Sie Mitarbeiter der Landesverwaltung NRW und haben Aufgaben zu erfüllen, die mit hohem Besprechungsaufwand und häufig auch Dienstreisen verbunden sind?
- Sind an diesen Besprechungen zwei oder auch mehr Teilnehmer aus verschiedenen Behörden – ggf. auch aus anderen Bundesländern – beteiligt?
- Sollen während einer laufenden Besprechung weitere Teilnehmer, beispielsweise per Handy, in diese Besprechung hineingerufen werden können?
- Müssen Dokumente gemeinsam bearbeitet werden?

Wenn Sie diese Fragen mit „ja“ beantworten können, dann sollten Sie über den Einsatz von Videokonferenzen nachdenken, um die beschriebenen Aufgaben von Ihrem Arbeitsplatz aus zu erledigen!

Das LDS NRW stellt für Mitarbeiter der Landesverwaltung seit Mai 2002 die erforderliche zentrale technische Infrastruktur zur Verfügung. Über das Landesverwaltungsnetz sind Videokonferenzen mit zwei oder mehr Konferenzteilnehmern möglich. Weitere Gesprächspartner können über ISDN zugeschaltet werden. Das zentrale System, die Multipoint Control Unit, ermöglicht hierbei die Teilnahme von unterschiedlich leistungsfähigen Videokonferenzendgeräten an einer Videokonferenz und verhindert so Kompatibilitätsprobleme zwischen den unterschiedlichen Systemen.

Wie kann ich Videokonferenzen durchführen?

Nach dem Start der Videokonferenzsoftware haben Sie drei Möglichkeiten eine Konferenz zu gestalten:

- Punkt zu Punkt-Verbindung zwischen zwei Teilnehmern: Über die aus dem Sukzessivleistungsvertrag beschafften Komponenten tragen Sie sich in ein zentrales Teilnehmerverzeichnis des Videokonferenzsystems NRW ein. Darüber können Sie nun mit einem Klick Anwender per Videokonferenz anrufen und eine Verbindung aufbauen.
- Ad-hoc-Konferenz mit zwei oder mehr Teilnehmern: Im Teilnehmerverzeichnis sind ebenfalls sogenannte offene Videokonferenzräume sichtbar, in die Sie sich wiederum mit einem Klick einwählen können. Alle Gesprächspartner wählen sich selbstständig in denselben Raum ein. Sie können mit mehreren Teilnehmern konferieren.
- Videokonferenzen mit Vorplanung und zentraler Steuerung: Mit einer separaten webbasierten Anwendung, dem WebCommander™, können Sie völlig eigenständig auf dem zentralen Videokonferenzsystem im LDS Videokonferenzen mit zwei oder mehr Teilnehmern konfigurieren. Über eine intuitiv zu bedienende Oberfläche haben Sie die Möglichkeit, Videokonferenzen zu planen und zu steuern. Alle Teilnehmer werden zu dem von Ihnen bestimmten Zeitpunkt automatisiert angerufen und in die gemeinsame Videokonferenz geschaltet. Die Teilnehmer werden vorab per E-Mail über die Konferenz informiert. Während der Konferenz können weitere Teilnehmer von Ihnen hinzugeschaltet werden. Sie entscheiden auch über die Darstellung der Konferenz auf den Monitoren der Teilnehmer.

Interesse? Lassen Sie sich über die neuen Möglichkeiten dieser Technologie informieren.

Ansprechpartner für das Videokonferenzsystem NRW sind Herr Salomon (norbert.salomon@lds.nrw.de, Tel.: 0211 9449-2262) und Herr Dr. Brakmann (markus.brakmann@lds.nrw.de, Tel.: 0211 9449-2383). Weitere Informationen zum Projekt erhalten Sie im Intranet der Landesverwaltung unter: <http://lv.ldsextern.nrw.de/videokonferenzen>

Angebot zentraler GIS-Dienste

mit dem GeoServer

Nach international verbreiteter Einschätzung besteht bei etwa 80 % aller Informationen ein mittelbarer oder unmittelbarer Raumbezug. Die Möglichkeit raumbezogene Informationen durch Geographische Informationssysteme (GIS) zu nutzen, ist bisher jedoch nahezu ausschließlich Experten vorbehalten geblieben. Routenplaner und Navigationssysteme sind die ersten raumbezogenen Anwendungen, die einem breiten Nutzerkreis zugänglich sind.

In der öffentlichen Verwaltung werden GIS seit längerer Zeit in den Bereichen Vermessung, Planung, Umwelt und Verkehr eingesetzt. Viele andere Bereiche der Verwaltung müssen heute noch immer auf häufig veraltete analoge Karten zurückgreifen oder bekommen auf ihre raumbezogenen Fragen keine Antwort.

Das Graphikzentrum hat begonnen, zentrale GIS-Dienste aufzubauen, die eine sehr flexible Nutzung raumbezogener Informationen ermöglichen werden. Die bisherigen GIS-Anwender werden hiervon vielerlei Nutzen haben. Vor allem aber wird ein sehr viel breiterer Zugang zu den Geodaten möglich sein.

Nachfolgend werden zunächst die technischen Grundlagen für die Bereitstellung von GIS-Diensten dargestellt und die hiermit (noch) verbundenen Grenzen aufgezeigt. Einer Darstellung der einzelnen GIS-Dienste und deren Nutzungsmöglichkeiten in WebGIS-Applikationen folgen Ausführungen zur Geodaten-Infrastruktur in NRW. Letztlich wird noch ein Ausblick auf sich abzeichnende weitere Entwicklungen gegeben.

Technische Entwicklung

Open GIS Consortium (OGC)

Das OGC ist eine 1994 gegründete Organisation, die sich ursprünglich zum Ziel gesetzt hat, Standards zu definieren, um Interoperabilität zwischen Geoinformationssystemen sowie zwischen diesen Systemen und Standardsoftware zu erreichen. Auf der Homepage des OGC (<http://www.opengis.org>) findet sich heute folgendes Selbstverständnis:

Open interfaces and protocols defined by OpenGIS® Specifications support interoperable solutions that "geo-enable" the Web, wireless and location-based services, and mainstream IT, and empower technology

developers to make complex spatial information and services accessible and useful with all kinds of applications.

Neben den Herstellern von GIS-Software, Hochschulen und Behörden finden sich unter den etwa 230 OGC-Mitgliedern auch Microsoft, Sun, Oracle und IBM. Gerade letzteres ist Indikator, dass die Arbeit des OGC wesentlich dazu beiträgt, GIS aus der Expertenecke herauszuführen.

Von den Spezifikationen des OGC hat die Web Map Service Implementation Specification bisher die größte praktische Bedeutung erlangt. Der Charme der WebMapServices liegt in der technischen Vereinfachung. Probleme mit proprietären Datenstrukturen sowie hohe Aufwände für die kartographische Visualisierung der Geodaten entfallen hier. Der Web-MapService begnügt sich mit 3 verschiedenen Anfragen:

getCapabilities	liefert eine XML-Datei mit Metainformationen zum Service, z. B. Ansprechpartner, verfügbare Layer, das Gebiet für den der Service zur Verfügung steht, Lagebezugssystem
getMap	liefert das Kartenbild mit dem angefragten fachlichen und räumlichen Inhalt in wählbarem Format, z. B. gif, jpg oder png
getFeatureInfo	liefert eine XML-Datei mit weiteren Informationen zu einem oder mehreren Geoobjekten.

Der WebMapService ist sicher sehr beschränkt in seinen Möglichkeiten. Er kann aber praktisch in jedem beliebigen Kontext mit Zugriff auf das Web sehr einfach zur Visualisierung von Karten verwendet werden. Abb. 1 veranschaulicht die Anwendung des OGC WebMapServices im GeoServer. Der AED Web-Client greift auf die Dienste des AED MapServers zu. Die farbigen Orthophotos sind auf dem AED MapServer jedoch nicht verfügbar und werden deshalb als OGC WebMapService eingebunden. Ebenso können beliebige weitere Services, auch aus dem Internet, eingebunden werden. Eine Reihe von öffentlichen Stellen, z. B. Kommunen und Verbände, betreibt z. Zt. prototypisch OGC WebMapServices mit unterschiedlichen Inhalten. Die Vernetzung der Geodatennutzung mittels dieser und komplexerer Techniken sind u. a. Ziel der Geodateninfrastruktur NRW (siehe unten).

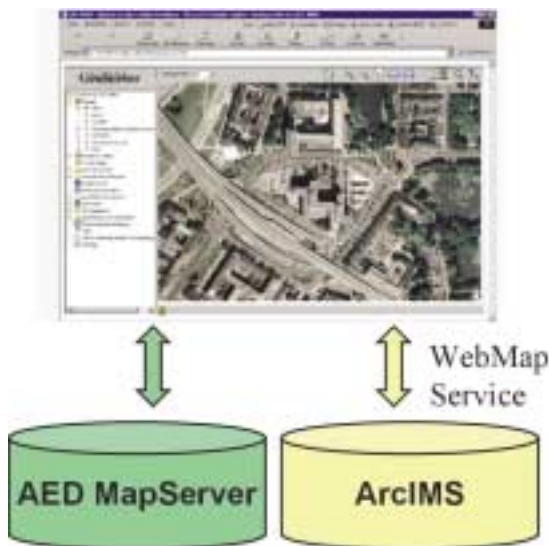


Abb. 1: OGC Web Map Service

Neue GIS-Technik

Die speziellen Anforderungen der verschiedenen Fachbereiche haben bisher dazu geführt, dass die einzelnen Disziplinen spezialisierte GIS mit speziellen Datenstrukturen eingeführt haben. Große Probleme bei der Interoperabilität und hohe Konvertierungsaufwände waren die Folge. Diese Probleme und das Bestreben, GIS in den IT-mainstream zu positionieren, haben einerseits zu erheblichen Anstrengungen der großen GIS-Firmen und gleichzeitig zu einer starken Marktbereinigung bei den Herstellern von GIS-Basistechnik geführt. Ergebnis dieser Entwicklung ist eine neue Generation von GIS, die sich den jeweils aktuellsten IT- und Web-Techniken, wie XML, J2EE und .NET bedient. Durch den flexiblen Zugriff auf unterschiedliche Datenquellen sowie die Möglichkeit der objektorientierten Modellierung der raumbezogenen und sonstigen Objekte, ist es möglich, die Integrationspotenziale von GIS voll auszuschöpfen.

Gleichwohl bleibt viel zu tun, da sich fachliche Probleme nicht in Luft auflösen. Bestehende GIS-Anwendungen sind aufgrund dieser Entwicklung auf die neue GIS-Technik¹⁾ umzustellen.

1) Siehe hierzu auch: Küpper, Stefan; Einführung des Geoinformationssystems „ArcGIS“ in der Landesverwaltung NRW mit Unterstützung des Graphikzentrums im LDS NRW (LDVZ-Nachrichten 01/2002).

Die Voraussetzungen für die Lösung der fachlichen Anforderungen sind aber ungleich besser geworden. Die entstehenden GIS-Dienste bieten zudem ein gutes Fundament sowohl für die Migration bestehender Applikationen als auch für die Schaffung neuer Informationsdienste.

GIS-Software am Arbeitsplatz verursacht jedoch hohe Kosten bei Beschaffung, Pflege und Betreuung und setzt zudem eine hohe Spezialisierung beim Nutzer voraus. Aufgrund der Komplexität der Software und den immer noch kurzen Innovationszyklen in diesem Bereich, scheint ein Einsatz nur sinnvoll zu sein, wenn die Nutzer umfassend geschult sind und die Anwendung häufig und regelmäßig erfolgt.

Viele Stellen der Landesverwaltung streben daher aktuell eine GIS-Entwicklung an, die weitgehend auf Web-basierten Techniken mit zentral betriebenen Diensten beruht. Installationen von GIS-Software am Arbeitsplatz soll so möglichst vermieden werden, um

- hohe Beschaffungs- und Wartungskosten einzusparen,
- den Betreuungsbedarf zu begrenzen und
- die Aufwände für spezielle Schulungen möglichst gering zu halten.

GIS-Software am Arbeitsplatz wird immer dann noch erforderlich sein, wenn hier Geodaten zu erfassen sind oder umfangreiche Analysen der Geodaten erfolgen sollen.

Technische Möglichkeiten zentraler GIS-Dienste

Riesige Datenmengen und lange Rechenzeiten haben viele Jahre die gesamte technische Entwicklung im GIS-Bereich stark beeinflusst. Die enormen Fortschritte im Hardwarebereich haben

hier inzwischen vieles möglich gemacht. Im Web-Umfeld und bei der Einrichtung zentraler GIS-Dienste spielt vor allem das Thema der Datenmengen wieder eine große Rolle.

Die hochwertige Druckausgabe von Karten ist in diesem Zusammenhang besonders problematisch. Während die MapServer die Karten für das Betrachten am Bildschirm mit einer Auflösung von etwa 80 dpi rendern, sind für eine hochwertige Druckausgabe Auflösungen zwischen 300 und 2 400 dpi zu generieren. Da gedruckte Karten zudem häufig große Formate bis hin zu DIN A0 haben können, wird deutlich, dass hier sehr große Datenvolumina entstehen können. Druckdateien, z. B. im Postscriptformat, können so durchaus größer als 100 MB sein. Aufgrund dieses Sachverhaltes ist in den Web-Mapping-Komponenten in der Regel entsprechende Funktionalität nicht vorgesehen.

Eine Ausnahme bildet hierbei der Geo-Server. Um im Web die großen Datenmengen angemessen verarbeiten zu können, wurde der Plotservice hier in das E-Commerce-Framework eingebunden, mit dem auch die Verteilung der Geodaten in der Landesverwaltung erfolgt. Die Erzeugung der Plotdatei im Postscript- oder im PDF-Format erfolgt hier asynchron. Die Auslieferung der Datei kann über einen http-Download (die URL wird via Mail übermittelt) oder mittels Datenträger erfolgen.

GIS-Dienste

MapServices

Das Graphikzentrum bietet zur Zeit Kartendienste auf Basis des AED MapServers und des ArcIMS der Firma ESRI an. Die Dienste werden hierbei jeweils in proprietärer Form zur Nutzung durch die firmenspezifischen Clients und als OGC WebMapService bereitgestellt und können sowohl durch entsprechende Web-Clients als auch durch eine Reihe Desktop-GIS genutzt werden (Abb. 2).

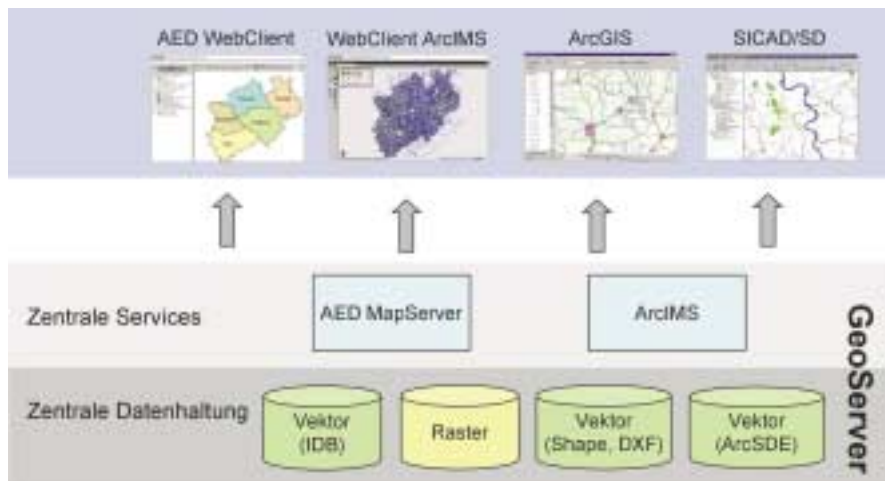


Abb. 2: Zentrale Mapping-Dienste

Entscheidend für die Attraktivität der MapServices ist in erster Linie der Umfang und die Aktualität des inhaltlichen Angebotes. Die Geobasisdaten des Landesvermessungsamtes sind hier wesentliche inhaltliche Grundlage. Neben den vorhandenen Diensten für die Rasterdaten der Deutsche Grundkarte 1 : 5 000 und der topographischen Kar-

ten aller Maßstäbe wurde der Dienst für die attraktiven farbigen Orthophotos vor Kurzem fertiggestellt. Die aus dem ATKIS-Datenbestand abgeleitete farbige DTK10, die auch alle Straßennamen enthält, rundet das Angebot der Geobasisdaten ab. Geodaten aus dem Umweltbereich, der Gebietsentwicklungsplanung sowie geologische und

bodenkundliche Daten des Geologischen Dienstes sind bereits zusätzlich verfügbar. Das inhaltliche Angebot wird sich vermutlich schnell verbreitern. Abb. 3 zeigt die Erzeuger der Geodaten sowie die Möglichkeit, in den Programmen ArcGIS und SICAD/SD neben den zentralen Services gleichzeitig auf lokale GIS-Daten zuzugreifen. MapInfo, das in der Landesverwaltung vor allem im Verkehrsbereich eingesetzt wird, kann die zentralen Services momentan noch nicht nutzen.

Weitere Services

Das Landesvermessungsamt hat in Zusammenarbeit mit den Katasterbehörden des Landes einen Datenbestand erstellt, der die Koordinaten zu allen Gebäuden mit eigener Hausnummer enthält. Dies sind ca. 3,6 Millionen Datensätze. Der Datenbestand ermöglicht es, über die Angaben Gemeinde, Stra-

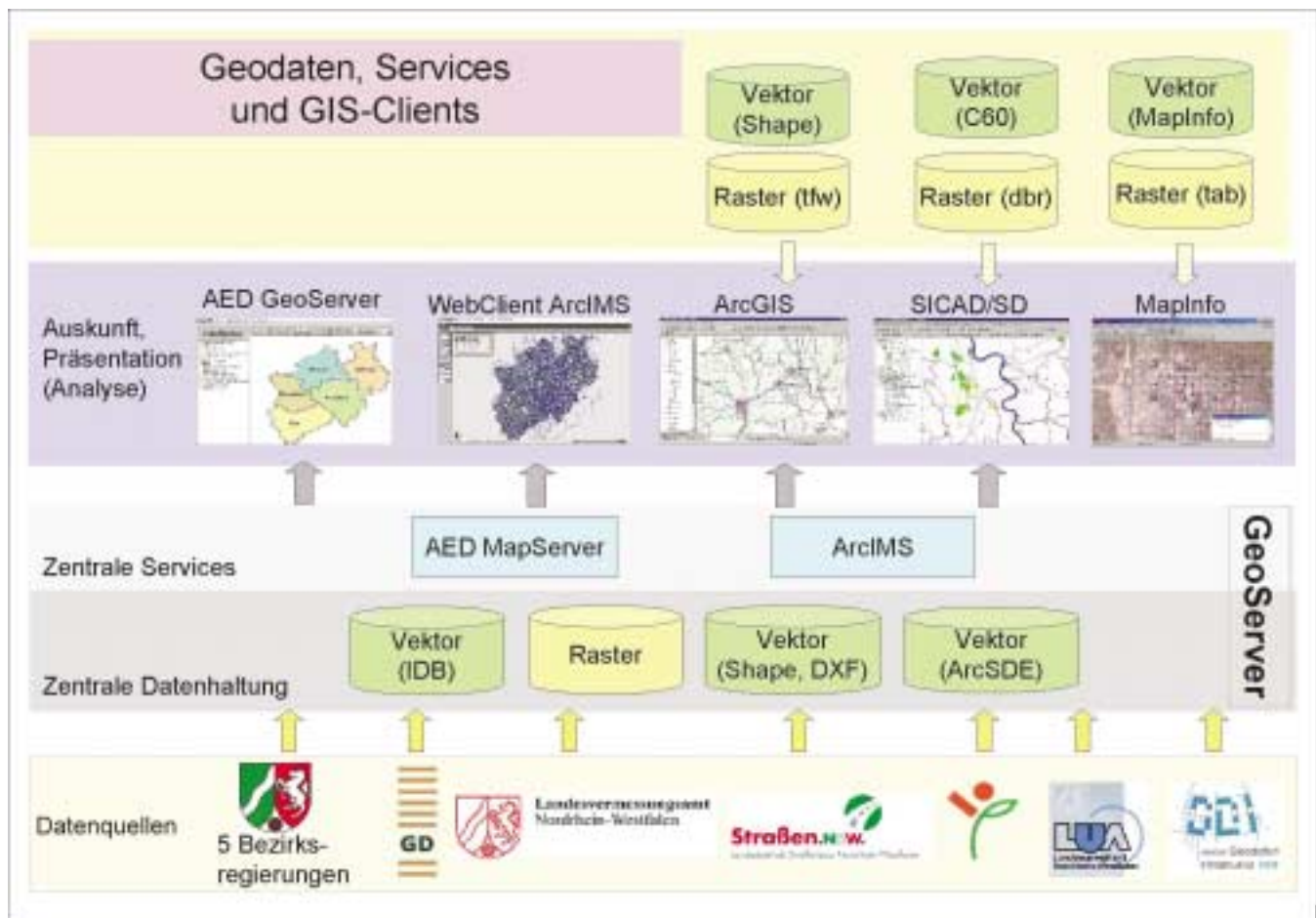


Abb. 3: Geodaten, Services und GIS-Clients

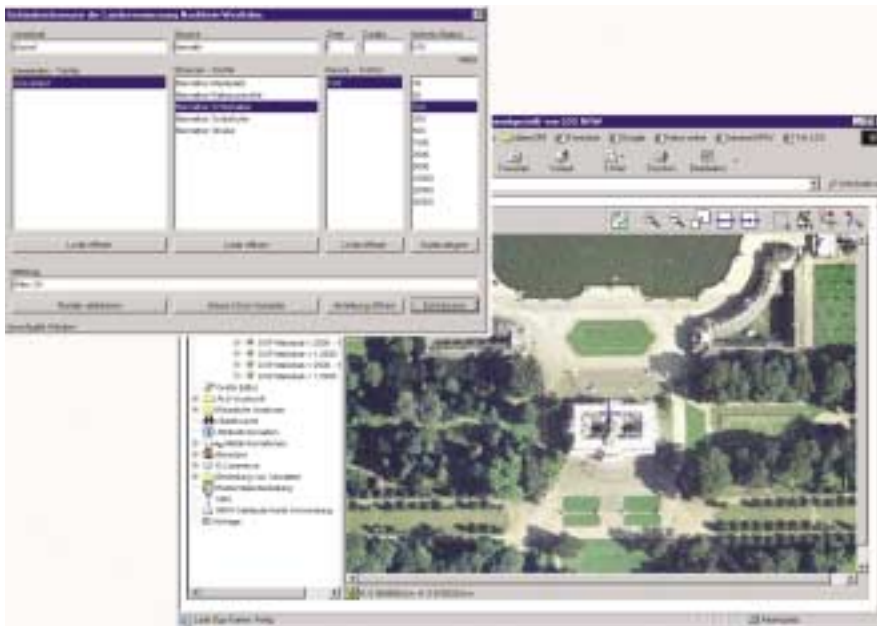


Abb. 4: Räumliche Navigation mittels Adresse

ße und Hausnummer einen gewünschten Kartenausschnitt zu betrachten. Mit diesen Daten können aber auch vorhandene Adress- oder Gebäudedatenbestände um die Lage angereichert bzw. vorhandene Lageangaben plausibilisiert werden.

Das Graphikzentrum hat für die Daten eine MySQL-Anwendung realisiert, die zunächst im GeoServer (s. Abb. 4) und für ArcIMS-Anwendungen genutzt werden kann. Weitere Realisierungen für das Intranet der Landesverwaltung sind in Vorbereitung, z. B. die Nutzung als Web-Service.

WebGIS-Applikationen

Die beiden GeoServer-Plattformen – AED MapServer und ArcIMS – werden auf Basis des Apache-Webrowsers sowie Tomcat (Application-Server) betrieben. Die Anbindung von Datenbanken, z. B. als ODBC-Datenquellen unterscheidet sich nicht von der Anbindung in anderen Web-Applikationen.

Die Nutzung von GIS-Techniken im Web eignet sich besonders gut für Informationsdienste, bei denen zu den Geodaten weitere Sachdaten aus Datenbanken

abgefragt werden. Hierbei sind zwei Realisierungsvarianten zu unterscheiden:

1. Die Web-Applikation wird als integriertes Auskunftssystem von Geo- und Sachdaten realisiert. Diese Variante wird meist zum Einsatz kommen, wenn mit replizierten Daten gearbeitet wird oder eine für die Fortführung der Sachdaten existierende Anwendung bestehen bleiben soll.
2. Die Web-Applikation wird als integriertes System zur Auskunft von Geo- und Sachdaten, sowie zur Pflege

ge der Sachdaten realisiert. Eine evtl. bestehende Client-Server-Applikation würde hiermit gleichzeitig auf Web-Technik umgestellt werden.

Die Geodaten der fachlich relevanten Objekte bilden jeweils den Ausgangspunkt der Überlegungen zur technischen Realisierung:

- Liegen Geometrien dieser Objekte vor (z. B. Schulstandorte, Biotope)?
- Können die Geometrien automatisiert abgeleitet werden (z. B. Adressen)?
- Sind die Geometrien noch zu erfassen?

Nur die Erfassung einfacher Punkte, Linien und Polygone wird nach dem aktuellen Stand der Technik unter bestimmten technischen Voraussetzungen in die WebGIS-Applikationen integriert werden können. Die Erfassung wird in den übrigen Fällen mit GIS-Software am Arbeitsplatz erfolgen müssen.

Die Wahrung der Konsistenz zwischen Geodaten und Sachdaten ist in diesem Zusammenhang eine grundsätzliche Problematik. Die Erfassung der Geodaten ist hierbei wiederum der Ausgangspunkt der Überlegungen. Werden die Geodaten extern gepflegt, ist eine Replikation und die periodische Überwachung der Konsistenz nicht zu vermeiden.



Abb. 5: WebGIS-Applikationen

den. Eine integrierte Pflege von Geo- und Sachdaten durch eine GIS-Applikation stellt technisch kein Problem dar. Allerdings werden hierdurch höhere Aufwände verursacht. Sind die Geodaten derart einfach strukturiert, dass eine Erfassung mit einer Web-Applikation möglich und sinnvoll ist, wird auch auf diesem Weg eine integrierte Datenpflege möglich sein. Ob Geo- und Sachdaten dann in einer Datenbank oder verteilt vorliegen, ist für diese grundsätzlichen Überlegungen von geringerer Bedeutung. Die Auswirkungen wären aber auf der technischen Seite (Transaktionskonzept) zu berücksichtigen.

Geodaten-Infrastruktur NRW (GDI NRW)

Das Leitbild der GDI NRW (<http://gdi.ecc-gmbh.de/iagent/upload/pdf/20021009163844.pdf>) hat sich folgenden Zielen verschrieben:

In der Initiative GDI NRW realisieren das Land NRW, Kommunen, IT-Unternehmen, Forschungseinrichtungen und Nutzer von Geoinformationen in Form eines Public-Private-Partnerships eine Geodateninfrastruktur, um den Geoinformationsmarkt in NRW zu aktivieren und die Nutzung der behördlichen und privaten Geodaten im Rahmen der gesetzlichen Möglichkeiten zu verbessern. Unter Geodateninfrastruktur versteht die Initiative ein offenes Geoinformationsnetz, das Geodatenproduzenten, Geodatenveredler sowie Geoinformationnutzer verbindet und so den Zugang zu allen verfügbaren Geodaten und Geoinformationsdiensten ermöglicht. Insbesondere soll ein problemloser Zugang zur Nutzung und marktwirtschaftlichen Wertschöpfung auf der Basis von behördlichen und privaten Geoinformationen gewährt werden. Dazu werden alle relevanten Geoinformationen der Behörden, Kommunen und privaten Anbieter verbunden und über das Internet verfügbar gemacht.

Die bereits oben dargestellten Spezifikationen des OGC bilden die wesentliche technische Grundlage für die angestrebte Interoperabilität von Diensten und Informationsprodukten.

Das Graphikzentrum ist aktiv an der Realisierung der Geodateninfrastruktur in NRW beteiligt. Neben der Mitarbeit in mehreren Special Interest Groups hat das LDS NRW in den beiden Jahren 2001 und 2002 durchgeführten Testbeds als so genannter Supporter aktiv teilgenommen. In den Testbeds wurden technische und organisatorische Aspekte der Geodateninfrastruktur im Konzept weiterentwickelt und im Rahmen von Implementierungen getestet und demonstriert. Das LDS NRW ist Mitglied in der GDI NRW.

Der GeoServer wird gemäß Erlass des Innenministeriums als zentrale Datendrehscheibe Kernelement der Geodateninfrastruktur des Landes sein.

Ausblick

GIS-Software wird sich auch in den nächsten Jahren weiter deutlich verändern. Die Entwicklung wird noch stärker als bisher durch das Thema „Web“ dominiert werden. Nahezu die gesamte Funktionalität moderner GIS-Software wird dann im Netz verfügbar sein. Diese Entwicklung findet sich bei den Diensten wieder, die auf Basis der Spezifikationen des OGC angeboten werden. Schlagworte des GDI Testbed II, wie WebFeatureService, Coordinate Transformation-Service, Gazetteer-Service²⁾ und Styled-Layer-Description³⁾

2) Verortung geographischer Bezeichnungen – 3) Nutzergesteuerte kartographische Präsentation von Web-MapServices



Abb. 6: Logo GDI NRW Testbed 1.0

stehen auch hier für zunehmende Komplexität der Dienste. Bei aller Euphorie für das technisch Machbare sollte der Charme einfacher Lösungen nicht verloren gehen. Die zukünftige Bedeutung von GIS wird wesentlich davon abhängen, ob auch „GIS-Laien“ erreicht werden können.

Durch die zentralen Dienste wird insgesamt eine wesentlich flexiblere Nutzung der Geodaten möglich. Das Graphikzentrum bietet diese zentralen GIS-Dienste in Abstimmung mit den Nutzern an und wird dafür Sorge tragen, dass zukünftige GIS-Dienste für den Anwender ergonomisch nutzbar sein werden. Ferner steht das Graphikzentrum auch für die Konzeption und Entwicklung webbasierter GIS-Applikationen und ggf. erforderlicher Desktop-GIS-Komponenten zur Verfügung.

Christoph Rath
Telefon: 0211 9449-6318
E-Mail: christoph.rath@lds.nrw.de

Landtagsinformationssystem NRW

Zum Landtagsinformationssystem Nordrhein-Westfalen gehören die Datenbanken Landtagsdokumentation seit der 10. Wahlperiode (1985 – 1990), Parlamentsspiegel, Pressedokumentation, Bibliotheksdatenbank, die Datenbank Parlamentsrecht/-gutachten, Parlamentsthesauri, die Abgeordnetendatenbank, LEGAL und MILAN (Material- und Informationsdatenbank).¹⁾ Im Internetangebot des Landtags können nur die Datenbanken Landtagsdokumentation, Parlamentsspiegel und Bibliotheksdatenbank abgefragt werden. Für die Erfassung der Daten und für spezielle Auswertungen wird das Datenbanksystem INFORMIX (4GL und SQL) eingesetzt. Die beim Landtag erfassten Daten werden bisher täglich mit Dateitransfer vom Rechner des Landtags zur BS2000-Anlage des LDS NRW geschickt und in die entsprechenden GOLEM-Datenbanken eingespeichert.

Im Frühjahr 2002 hat der Landtag Nordrhein-Westfalen das LDS NRW beauftragt, das Landtagsinformationssystem mit allen Datenbanken im Intranet des Landtags zur Verfügung zu stellen und die Anwendungen hierfür zu programmieren. Zu den Aufgaben gehören:

Datenbankdesign

Das LDS NRW erstellt in Zusammenarbeit mit dem Referat Informationsdienste des Landtags NRW das Datenbankdesign. Es sollen sowohl die Anforderungen der Fachanwender als auch die der Laiennutzer mit den zur Verfügung stehenden Techniken abgedeckt werden.

Entwicklung der Programme

Das LDS NRW entwickelt alle Programme und Verfahrensabläufe, die erforderlich sind, um die Daten, die in die Erfassungsdatenbanken (mit dem Datenbanksystem INFORMIX) eingegeben werden, in die Recherchedatenbanken zu übertragen.

Entwicklung der Oberflächen für die Suche und die Anzeige der Rechercheergebnisse

Gemeinsam mit dem Landtag (Referat Informationsdienste) werden die Suchoberflächen und die Anzeige der Suchergebnisse für die Fachanwender und die Laiennutzer des Landtags entwickelt.

Der Landtag hat 1998 entschieden, für die Recherche das Retrievalsystem BASIS[®] mit den entsprechenden Komponenten einzusetzen. Als erste Datenbank wurde von Mitarbeiterinnen und Mitarbeitern des Landtags die Pressedokumentation umge-

stellt. Sie enthält Beiträge aus den täglichen Presseschauen des Landtags NRW, ergänzt um Beiträge aus der überregionalen Presse und den Wochenzeitungen. Zusätzlich ist eine Recherche in den Volltexten möglich. BASIS[®] kann auf verschiedenen Systemplattformen eingesetzt werden, im Landtag läuft es auf einem Server mit dem Betriebssystem AIX. Werkzeuge für die Suche bzw. Recherche sind SQL und FQM (Fundamental Que-

Suchmöglichkeiten mit BASIS[®]

Hierarchische Suchfähigkeiten
Suche in mehreren Datenbanken, Crossrecherche
Näherungs- und Wildcardsuche
Vordefinierte Suche (Makros)
Links-, Mitte- und Rechtstrunkierung
Wort-, Begriff- und Abstandssuche
Numerische Suche, einschließlich Daten und Bereichen
Phonetische Suche

ry).

BASIS[®] benutzt ein leistungsstarkes Data Dictionary (DDB) für Datenbankdefinitionen; folgende Sichten bzw. Architekturen bieten eine flexible Gestaltung der Daten:

Das ACTUAL DATA MODEL (ADM) ist die logische Beschreibung der Datenbank.

Das USER DATA MODEL (UDM) ist die Beschreibung der Daten, so wie sie von den einzelnen Benutzern oder den Programmen gesehen werden.

Das STRUCTURAL DATA MODEL (SDM) ist die physikalische Beschreibung der Datenbank.

WEBTOP ist das Werkzeug für die Anbindung der BASIS[®] Datenbanken an marktübliche Browser (Webserver) wie Netscape Navigator oder Internet Explorer. Die Anpassung an internetfähige Oberflächen wird mit Java Server Pages, Java Beans, Javascript, HTML, Cascading Style Sheets (CSS) realisiert. Die Abbildung 1 stellt die Beziehungen zwischen den verschiedenen Anwendungen dar.

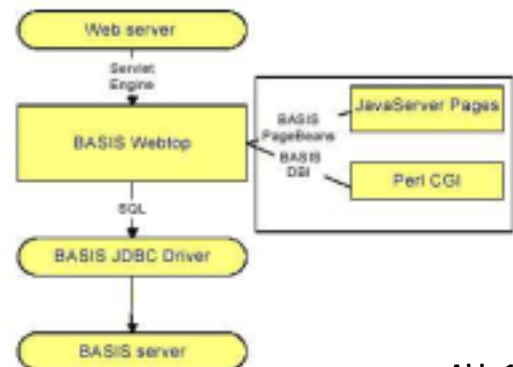


Abb. 1

¹⁾ LDVZ-Nachrichten 1/2001, S. 17



Abb. 4

Das Layout für die Datenbanken wurde gemeinsam mit dem Referat Informationsdienste, unter der Angabe, dass es sich in den Rahmen des Gesamtauftritts im Intranet des Landtags einpasst, entwickelt. Über die Leitseite des Referates und über einen vom Referat gestalteten Punkt „Fachinfo“ kommt man auf die Seite der gewünschten Datenbank.

Inzwischen wurden vom LDS NRW die Verfahren Parlamentsrecht/-gutachten und MILAN umgestellt. Die Eingangsseite enthält eine kurze Beschreibung der jeweiligen Datenbank, eine „Schnellsuche“ zur Eingabe einfacher Suchfragen, und es werden vorgefertigte Suchfragen „Profile“ angeboten.

In den Suchbegriffen bzw. Indices, wie z. B. bei der Kategorie ‚Autor‘, kann man navigieren und die gewünschten Deskriptoren in die Suchfrage übernehmen, indem man die entsprechenden Begriffe anklickt.

In der Ergebnismenge (vgl. Kurz- (Abb. 5) bzw. Detailansicht Abb. 6) kann man über das Anklicken des □ Symbols die Originale der nachgewiesenen Dokumente ansehen und auch drucken. Mit dem Buchsymbol wird ein Auftrag für eine Ausleihe per E-Mail an die Bibliothek des Landtags geschickt.

Für die einmalige Übernahme der vorhandenen Daten aus den INFORMIX-Datenbanken wird das Verfahren HVU (high volume update von BASIS®) eingesetzt. Die Mitarbeiterinnen und Mitarbeiter des Referates Informationsdienste erfassen die Daten weiterhin online mit INFORMIX. Danach sorgt das vom LDS NRW in der Programmiersprache JAVA erstellte Programm DB_Transfer für die Spiegelung der Daten. Sobald z. B. ein bearbeitender Dokumentar oder eine Dokumentarin in einer der vier Teildatenbanken Bibliothek, Gutachten, Presse oder Urteile der Parlamentsrechts-Datenbank einen Datensatz neu anlegt, ändert oder



Abb. 5: Rechercheergebnis in der Kurzfassung



Abb. 6: Rechercheergebnis in der Detailansicht

löscht, werden die Datenbanken auf beiden Systemen INFORMIX und BASIS synchronisiert. Dieses Programmsystem ist so allgemein gehalten, dass es über Parameter gesteuert für alle weiteren Datenbanken und Verfahren des Landtags eingesetzt werden kann.

Das Referat Informationsdienste des Landtags plant zusätzlich neue Dienstleistungen:

1. Standardisierte Suchprofile im Intranet und z. T. auch im Internet des Landtags zur schnellen und komfortablen Ermittlung benötigter Informationen.
2. Auf Wunsch Zusendung individueller Suchstrategien per E-Mail zur Abspeicherung und zur persönlichen Nutzung.
3. Zu aktuellen Anlässen Abspeicherung von Suchergebnissen als Link, eingestellt auf der Intranethomepage des Landtags.
4. Standard-Cross-Recherchen über mehrere Landtagsdatenbanken.
5. Suchprofile als regelmäßiger E-Mail-Dienst über Verteilerlisten.

Die hierfür notwendigen Verfahren werden ebenfalls vom LDS NRW entwickelt. Zusätzlich zu den beschriebenen Suchverfahren werden Recherche-Oberflächen für die erweiterte bzw. Expertensuche programmiert.

Brigitte Corves
 Telefon: 0211 9449-6305
 E-Mail: brigitte.corves@lds.nrw.de

Microsofts .NET-Technologie

– wohin geht die Programmierung?

Que vadis oder Where do you want to go today?

Seit Anfang des Jahres 2002 wird in Programmiererkreisen über die von Microsoft entwickelte .NET-Architektur gesprochen. Dieser Artikel soll beleuchten, was sich dahinter verbirgt und für wen .NET interessant sein könnte.

Das .NET-Framework

Die englische Bezeichnung *Framework* bedeutet in etwa „Rahmen“, „Gerüst“. Es handelt sich also nicht einfach um eine neue Programmiersprache, sondern um ein Grundgerüst – eine *Architektur* – für die Entwicklung und Ausführung von Programmen, und zwar insbesondere (aber nicht ausschließlich) von Internetanwendungen.

Viele sehen .NET als Microsofts Antwort auf Java, und die historischen Ursprünge sind auch tatsächlich dort zu finden. Aber auch Java ist ja nicht nur eine Programmiersprache, sondern bringt eine ganze Reihe von Technologien mit sich – die nahezu alle in .NET eine Entsprechung finden.

Die wesentlichen Bestandteile der .NET-Architektur sind die Laufzeitumgebung (CLR, *Common Language Runtime*) und die .NET-Klassenbibliothek (FCL, *Framework Class Library*).

Die Common Language Runtime

Die Laufzeitumgebung erfüllt eine ähnliche Funktion wie bei Java die Virtual Machine: Anstatt Programme in Maschinencode für einen bestimmten Prozessor zu übersetzen, erzeugen die Compiler der .NET-Sprachen einen maschinenunabhängigen Zwischencode (IL, *Intermediate Language*). Dieser Zwischencode kann dann auf jeder Plattform ablaufen, auf der die CLR installiert ist. Die CLR kann als Teil des .NET Framework SDK bei Microsoft kostenfrei heruntergeladen und zusammen mit Anwendungen ausgeliefert werden. Bei zukünftigen Windows-Versionen soll das Framework sogar Bestandteil des Betriebssystems sein, so dass .NET-Programme ohne zusätzliche Installationsarbeiten ausgeführt werden können.

Vor der Ausführung wandelt ein *Just-In-Time-Compiler* (JIT), der Bestandteil der CLR ist, den Zwischencode in Maschinencode für den jeweiligen Zielrechner um, und

zwar gezielt nur diejenigen Programmteile, die gerade gebraucht werden. Dieser Vorgang kostet zwar bei der ersten Ausführung etwas Zeit, insgesamt müssen solche Programme aber nicht unbedingt langsamer sein als vorab übersetzte, da der Just-in-Time-Compiler zur Laufzeit über Informationen bezüglich des Zielsystems verfügt, die eine bessere Optimierung ermöglichen können. Bei Bedarf kann der komplette Code aber auch mit einem im .NET-Framework-SDK enthaltenen Tool vorkompiliert werden.

Bei der Just-In-Time-Kompilierung führt die CLR eine Reihe von Prüfungen durch, die eine hohe Zuverlässigkeit des übersetzten Programmcodes garantieren. Beispielsweise wird sichergestellt, dass keine Speicheradresse ausgelesen wird, die nicht vorher beschrieben wurde, dass Funktionsaufrufe (Aufrufe von Klassenmethoden) mit der richtigen Anzahl von Argumenten und den richtigen Datentypen erfolgen, dass jede Methode einen Wert zurückgibt und dieser im aufrufenden Programm korrekt verarbeitet wird usw.

Da durch die Codeüberprüfung Programme voreinander geschützt sind, können sogar mehrere Anwendungen (bei .NET *AppDomains* genannt) in demselben Prozess ausgeführt werden.

Die .NET Klassenbibliothek

Die .NET-Klassenbibliothek stellt einen Versuch dar, die Programmierung sprachübergreifend zu vereinheitlichen: alle Programmiersprachen, für die .NET-Unterstützung verfügbar ist, können auf dieselben Klassen zurückgreifen – und dazu sollen neben C#, Visual Basic, und Visual C++ auch Java (Visual J#) sowie für die Programmierung im Internet-Umfeld eher untypische Sprachen wie etwa Cobol zählen.

Die Klassenbibliothek enthält ein breites Spektrum von Klassen für die verschiedensten Aufgaben wie beispielsweise Gestaltung der grafischen Oberfläche, Netzwerkprogrammierung, Webanwendungen, XML, um nur einige zu nennen. So können beispielsweise mithilfe der Windows-Forms-Klassen grafische Benutzeroberflächen mit Fenstern und den üblichen Steuerelementen (Schaltflächen, Textfelder, Kombinations- und Listfelder, Kontrollkästchen, Optionsfelder usw.) entwickelt werden – unabhängig davon, ob die Programmierung in C#, Visual Basic oder einer anderen Sprache erfolgt.

Wichtige Bestandteile der Klassenbibliothek, welche eigentlich eigene Technologien verkörpern, sind ADO.NET für Datenbankzugriffe und ASP.NET für die Entwicklung von Webanwendungen.

Die Nachfolgeversion des Windows-2000-Betriebssystems, die in Kürze erscheint, wird .NET Server heißen. Auch daran ist zu erkennen, wie umfassend die .NET-Initiative gedacht ist: Die Unterstützung von Anwendungen durch Betriebssystemdienste (die im Wesentlichen auch schon in bisherigen Windows-Versionen vorhanden sind) wird unter der Bezeichnung „.NET“ vor allem für Webanwendungen besonders herausgestellt und vermarktet.

Neben dem eigentlichen Framework und dem Betriebssystem hat Microsoft auch eine Reihe von Serverprodukten mit dem Label „.NET“ versehen:

- Application Center 2000
- BizTalk Server 2000
- Commerce Server 2000
- Exchange 2000
- Host Integration Server 2000
- Internet Security und Acceleration (ISA) Server 2000
- Mobile Information Server 2002
- SQL Server 2000

Diese Produkte sind aber selbst (noch?) nicht mit .NET geschrieben, sondern sollen lediglich .NET-Anwendungen unterstützen. Die nächste Version des SQL-Servers wird jedoch einen CLR-COM-Server enthalten, und gespeicherte Prozeduren können dann in .NET-Programmiersprachen entwickelt werden.

Interoperabilität

Ein wichtiges Ziel, das mit dem .NET-Framework erreicht werden soll, stellt die *Interoperabilität* (Fähigkeit zur Zusammenarbeit) zwischen verschiedenen Programmkomponenten dar, und zwar unabhängig davon, in welcher Sprache

diese entwickelt wurden. Die Interoperabilität der .NET-Sprachen untereinander geht sogar so weit, dass Typen (Klassen) von in einer anderen Sprache entwickelten Typen abgeleitet werden können.

.NET ermöglicht auch die Kommunikation zwischen Komponenten, die auf verschiedene Rechner verteilt sind: Im Extremfall arbeiten so verschiedene Programmteile, die von unabhängigen Herstellern zur Verfügung gestellt werden, über das Internet reibungslos zusammen. Insofern können mit .NET auch Aufgaben realisiert werden, für die bislang eine relativ aufwändige Programmierung mit DCOM bzw. RMI und CORBA notwendig war.

XML-Webservices (s. u.) ermöglichen die Kommunikation über das Internet auf Basis von Funktionsaufrufen – zwischen .NET-Anwendungen untereinander und mit anderen Anwendungen (z. B. Java-Programmen).

Die Interoperabilität zwischen einzelnen, unabhängigen Programmkomponenten wird realisiert auf der Basis von so genannten Assemblies (vergleichbar den JAR-Archiven).

Ein Assembly¹⁾ ist eine selbst beschreibende, wiederverwendbare Programmkomponente mit Mechanismen für Versionsverwaltung und Sicherheit. Physisch ist ein Assembly eine DLL oder eine EXE-Datei.

Was bedeutet nun „selbst beschreibend“? Zu jedem Assembly gehört ein *Manifest*, das in dem Assembly selbst oder in einer separaten Datei enthalten sein kann. Das Manifest beschreibt die von dem Assembly bereitgestellten Klassen mit ihren Memberfunktionen und deren Signatur (Anzahl und Datentypen der Parameter sowie Rückgabewerte). Somit wird anderen Komponenten ermöglicht, die Funktionalität des Assemblys zu nutzen.

Da jedes Manifest auch Versionsinformationen enthält, ist es möglich, dass mehrere Versionen desselben Assembly auf einem System nebeneinander existieren. Clientanwendungen können gezielt auf eine bestimmte Version zugreifen – womit einige altbekannte Probleme der Vergangenheit angehören, z. B., dass eine Anwendung eine DLL mit einer Version überschreibt, die mit anderen Anwendungen (welche dieselbe DLL benötigen) nicht verträglich ist. Andererseits können Systemadministratoren aber auch die Verwendung einer bestimmten Version erzwingen – dann tragen sie auch selbst die Verantwortung für die Kompatibilität.

Da Assemblies mit Standard-Verschlüsselungsverfahren signiert werden können, sind sie fälschungssicher. Ein über den Inhalt der Datei berechneter *Hash-Wert* wird mit dem privaten Schlüssel des Herstellers verschlüsselt und zusammen mit dem zugehörigen öffentlichen Schlüssel in der Datei eingebettet. Vor der Installation beim Kunden berechnet das System denselben Hash-Wert, entschlüsselt den eingebetteten mit dem öffentlichen Schlüssel und vergleicht die beiden. So ist einerseits gewährleistet, dass die Datei tatsächlich mit dem zum öffentlichen Schlüssel gehörigen privaten Schlüssel signiert wurde und dass andererseits auch der Inhalt der Datei nicht nachträglich verändert wurde. Eine wirkliche eindeutige Identifikation des Herstellers wird durch zusätzliche Anwendung der Authenticode-Technik (Anbieter-Zertifikate) ermöglicht. Damit sind die Voraussetzungen für eine Isolation und Zugriffssteuerung auf Basis der Herkunft des Codes gegeben. So können etwa Programme und Skripte aus dem Internet mit stärkeren Beschränkungen ausgeführt werden als solche aus vertrauenswürdigen Quellen.

1) In deutschen Texten wird von einigen Autoren die sächliche, von anderen die weibliche Form verwendet: das Assembly oder die Assembly.

Common Type System

Ein wichtiges Merkmal der .NET-Programmierung ist die Objektorientierung. Die eingesetzten Programmiersprachen sind in der Regel objektorientiert (wie C# und Visual Basic .NET), und sogar die vom Compiler erzeugte IL stellt eine Art objektorientierte Maschinensprache dar. Zentrales Element der objektorientierten Programmierung sind selbst definierte *Typen* (Klassen). Die Handhabung dieser Typen ist in den verschiedenen Programmiersprachen unterschiedlich, auf der Ebene der IL jedoch einheitlich. Die Definition von Typen und deren Verhalten ist deshalb in einer Spezifikation, dem *Common Type System* (CTS) festgelegt. Das CTS legt u. a. fest, welche *Member* ein Typ enthalten kann (Felder, Methoden, Eigenschaften, Ereignisse), gibt die möglichen Einstellungen für die *Sichtbarkeit* von Typen (Public, Assembly) und deren Methoden (Private, Family, Family und Assembly, Assembly, Family oder Assembly, Public) an, regelt die *Vererbung*, *virtuelle Funktionen*, Lebensdauer von Objekten usw.

Common Language Specification

Eine weitere Spezifikation, die *Common Language Specification* (CLS), legt Regeln für die Programmierung von Komponenten fest, die mit in anderen Sprachen geschriebenen Programmen zusammenarbeiten sollen. Dies ist notwendig, weil die Programmiersprachen sich teilweise doch erheblich voneinander unterscheiden und jeweils verschiedene Untermengen des durch CLR und CTS festgelegten Gesamtspektrums an Möglichkeiten realisieren.

So unterscheiden z. B. C# und C++ zwischen Groß- und Kleinschreibung, Visual Basic jedoch nicht. Nicht alle Sprachen haben die Fähigkeit, Operatoren zu überladen oder Methoden mit einer variablen Anzahl an Argumenten zu definieren usw.

Die CLS zwingt den Programmierer also, auf bestimmte Merkmale der von ihm gewählten Programmiersprache zu verzichten, wenn sein Code mit anderen Programmiersprachen verträglich sein soll. Ein besonderes Attribut innerhalb des Quelltextes veranlasst den Compiler, die CLS-Konformität zu überprüfen. Solange keine Interoperabilität benötigt wird, kann man natürlich alle Fähigkeiten der bevorzugten Programmiersprache ausnutzen.

Rechnerübergreifende Zusammenarbeit

Die Kommunikation zwischen Komponenten über die Grenzen von Anwendungen (AppDomains), Prozessen und sogar Rechnern hinweg kann mit den Remotingmechanismen des .NET-Frameworks oder über XML-Webdienste erfolgen.

.NET-Remoting ist ein programmier-technisch relativ unkomplizierter Mechanismus, der es ermöglicht, Methoden von Objekten in anderen AppDomains (auch auf anderen Rechnern) aufzurufen. Als Protokoll kann beispielsweise TCP, HTTP oder SMTP eingesetzt werden, das Format ist wahlweise binär oder XML (SOAP). .NET-Remoting wird für die Kommunikation von .NET-Komponenten untereinander eingesetzt, z. B. in verteilten Anwendungen im Netzwerk.

Webservices sind Dienste, die von Clients über das Internet aufgerufen werden können. Sie stellen beispielsweise Informationen zur Verfügung, ermöglichen Bestellungen und Reservierungen usw.

Während Benutzer auf herkömmliche Webanwendungen über einen Browser zugreifen und folglich mittels HTML bedient werden, kommunizieren Webservices mittels XML (*eXtended Markup Language*), und zwar in der Regel nicht direkt mit Benutzern, sondern eher mit anderen Programmen. Indem

Funktionsaufrufe in einem einheitlichen, fest definierten Format erfolgen und die Ergebnisse in einem ebensolchen festen Format zurückgegeben werden, kann ein Programm Informationen von einem oder mehreren Webdiensten abrufen und gemeinsam verarbeiten.

Das Clientprogramm kann selbst wieder eine Webanwendung sein, der die Informationen ggf. in einen neuen Zusammenhang stellt und an die Endkunden weiterleitet. So könnte z. B. die Anwendung eines Reisebüros die Fahrplandaten der Bahn und die Flugpläne der Fluggesellschaften abfragen, mit den Reservierungssystemen verschiedener Reiseveranstalter oder Hotels kommunizieren und dem Benutzer schließlich ein individuell zusammengestelltes Komplettangebot unterbreiten – und ihn als Zugabe vielleicht auch noch mit Wetterberichten und Devisenkursen für das Reiseziel versorgen.

Ähnlich wie auf einem einzelnen Rechner verschiedene .NET-Komponenten zusammenarbeiten, kann über das Internet die Zusammenarbeit sogar zwischen Komponenten erfolgen, die Tausende von Kilometern voneinander entfernt auf verschiedenen Rechnern laufen. Im Gegensatz zu .NET-Remoting können XML-Webservices auch z. B. mit Java-Programmen auf anderen Plattformen zusammenarbeiten.

XML-Webdienste kommunizieren über SOAP (*Simple Object Access Protocol*). SOAP ermöglicht die Übertragung von Daten in Form von XML über das HTTP-Protokoll. Das funktioniert im Gegensatz zu klassischen Remoting-Mechanismen sogar durch Firewalls hindurch.

SOAP und XML-Webservices wurden nicht erst mit .NET erfunden, sondern die Entwicklung wird von einer Reihe von Herstellern unterstützt, u. a. Sun, IBM und Microsoft. Auch in Java können Webdienste programmiert werden

(mithilfe des Java Web Services Developer Pack, künftig auch in J2EE 1.4 integriert) – allerdings ist die Unterstützung bei .NET nahtloser, da Webservices eine Kernfunktionalität des .NET-Konzepts darstellen.

An der Bedeutung von Webservices zweifelt heute kaum jemand, wenn es auch zurzeit erst wenige praktische Anwendungen gibt.

UDDI (Universal Description, Discovery and Integration) ist ein Standard zur Veröffentlichung und zum Suchen von Informationen über XML-Webdienste. Der Webservice selbst wird dabei meist in WSDL (Web Services Description Language) beschrieben.

.NET in der Praxis

Im Folgenden sollen die wichtigsten .NET-Programmiersprachen und die dazugehörigen Entwicklungswerkzeuge sowie die Technologien ASP.NET und ADO.NET näher betrachtet werden.

Für erste Gehversuche mit .NET reicht das Framework-SDK allein – für die professionelle Anwendungsentwicklung ist Visual Studio.NET geeignet.

Theoretisch benötigt man zum Erstellen von .NET-Anwendungen keine besondere Entwicklungsumgebung: Quellprogramme können mit jedem beliebigen Texteditor erstellt werden (Notepad, wenn es sein muss). Für die Übersetzung und Ausführung wird das .NET-Framework-SDK (*Software Development Kit*) gebraucht, welches von Microsoft kostenlos zur Verfügung gestellt wird und auch zusammen mit Anwendungen weitergegeben werden darf. Das Framework SDK enthält kommandozeilen-gesteuerte Werkzeuge (Compiler, Assembler/Disassembler usw.).

Weitaus komfortabler geht es natürlich mit Microsofts Entwicklungsumgebung Visual Studio.Net: Diese enthält

die Sprachen C#, Visual Basic und C++ mitsamt Editor, Compiler, Linker, Debugger und Online-Hilfe sowie Werkzeuge und Assistenten zum Entwerfen grafischer Benutzeroberflächen, zum Erstellen von Installationsprogrammen und vieles mehr. (Einzelne Sprachen sind auch separat erhältlich.) Das Framework-SDK ist natürlich ebenfalls enthalten.

Die Sprache **C#** (sprich: C-Sharp) wurde zusammen mit dem .NET-Framework neu entwickelt. Die Bezeichnung ist angelehnt an die Schreibweise in der Musik für den um einen Halbton erhöhten Ton „C“ (im Deutschen Cis) und soll also ähnlich wie der Name „C++“ aussagen, dass es sich hier um eine Sprache handelt, die an C angelehnt ist, aber noch etwas mehr kann. Dementsprechend sieht C#-Code mit seinen geschweiften Klammern und C-Kontrollstrukturen ähnlich aus wie C/C++- oder Java-Code. Auch in der Objektorientierung hat C# viele Ähnlichkeiten insbesondere mit Java.

C# ist für die ISO-Standardisierung vorgesehen.

Neben C# ist **Visual Basic .NET** die Sprache mit der engsten .NET-Anbindung: Hier sind natürlich Blöcke und Kontrollstrukturen in Basic-Syntax; dieses Visual Basic ist aber auch eine vollwertige objektorientierte Programmiersprache. Da alle Programmieraufgaben mit der .NET-Klassenbibliothek gelöst werden, lassen sich VB-.NET-Programme praktisch 1:1 in entsprechende C#-Programme übersetzen.

Der Preis für diese Einheitlichkeit ist allerdings der, dass VB .NET nur noch wenig mit früheren Visual-Basic-Versionen zu tun hat: VB6-Programme sind nicht unter VB.NET ablauffähig, und trotz eines Migrationswerkzeugs in Visual Studio ist die Umstellung eine extrem aufwändige Aufgabe. Programmierer müssen „ihre“ Sprache praktisch neu erlernen und mit vielen alten Gewohnheiten brechen. Die Gründe für

diese radikale Änderung liegen teils darin, dass VB.NET jetzt alle Merkmale einer echten objektorientierten Sprache bietet und streng typisiert ist, andererseits dass viele Aufgaben, für die bislang „eingebaute“ Funktionen zur Verfügung standen, nun mithilfe der .NET-Klassenbibliothek gelöst werden. VB-Programmierer haben aber langfristig keine andere Wahl, als auf VB .NET umzusteigen – in diesem Bereich wird sich .NET sicherlich durchsetzen.

Die .NET-Anbindung von **Visual C++ 7.0** wirkt weit weniger elegant und nahtlos als diejenige von C# und Visual Basic. Hier werden .NET-Programme realisiert mit den so genannten *Verwalteten Erweiterungen (Managed Extensions)*. Zwar kann der damit entwickelte verwaltete Code (s. u.) mit normalen, unverwalteten Programmen integriert werden, dabei sind jedoch eine Reihe von technischen Details zu beachten, wodurch die Freude an der Einfachheit des verwalteten Codes deutlich getrübt wird. Zudem wurde die Sprache C++ um .NET-spezifische, neue Sprachelemente erweitert, die durch einen speziellen Compilerschalter aktiviert werden. Solche Programme sind natürlich nicht mehr mit jedem C++-Compiler und erst recht nicht auf anderen Betriebssystemen übersetzbar.

Programmierer von Windows-Anwendungen müssen sich künftig entscheiden, ob sie wie bisher die MFC-Klassenbibliothek verwenden wollen (die Microsoft-Dokumentation erweckt den Eindruck, als sei dies nur noch zur Pflege von „Altlasten“ wünschenswert), oder ob sie mit den neuen .NET-Klassen (z. B. Windows Forms) arbeiten. Auch die direkte Programmierung der Win32-API soll durch die .NET-Klassen abgelöst werden.

Für die Programmierung mit Windows Forms lässt die Unterstützung bei Visual C++ bislang allerdings noch einiges zu wünschen übrig: Formulare und Dialogfelder können nur per Programmcode entwickelt werden, ein vi-

suelles Werkzeug fehlt. Es stellt sich daher die Frage, ob man in diesem Fall nicht besser gleich Visual Basic oder C# benutzt.

Im kommerziellen Umfeld wird die MFC-Klassenbibliothek sicherlich noch für einige Zeit das Werkzeug der Wahl für C++-Programmierer bleiben – allerdings hat Microsoft auf lange Sicht wohl nicht die Absicht, diese weiter zu unterstützen.

.NET-Programme werden auch als „verwalteter Code“ bezeichnet. Damit ist ein **Code** gemeint, welcher der Laufzeitüberprüfung und insbesondere der Speicherverwaltung des .NET-Frameworks unterliegt: Dynamisch (mit dem Operator *new*) allozierter Speicherplatz wird automatisch wieder freigegeben, sobald er nicht mehr benötigt wird (*Garbage Collection*). In den Sprachen C# und Visual Basic (und übrigens auch in Java) ist dies eine Selbstverständlichkeit, in normalen C++-Programmen (ohne Managed Extensions) ist der Programmierer dagegen selbst für die Speicherverwaltung verantwortlich. Wo unverwalteter C++-Code mit verwaltetem zusammenarbeiten soll, ist besondere Sorgfalt mit der Speicherfreigabe geboten.

Datenbankzugriff mit ADO.NET

Der Datenbankzugriff mit ADO (Active Data Objects) wird durch .NET gegenüber vorhergehenden ADO-Versionen stark verändert: So genannte *DataSets* werden vorab mit Daten gefüllt und ermöglichen dann eine *verbindungslose* Verarbeitung. So kann auch bei Mehrbenutzerzugriff im Netz eine maximale Performance gewährleistet werden, und mobile Geräte sind sogar zeitweilig in der Lage, ganz ohne Verbindung zur Datenbank zu arbeiten.

Programmierer können den Kern des Datenzugriffs in wenigen Zeilen selbst schreiben oder aber in Visual Studio werkzeuggestützt aus Kompo-

nenten „zusammenbauen“. Steuerelemente mit Datenbindung (z. B. **Data-Grid** für eine tabellarische Darstellung) ermöglichen das Anzeigen von Datenbankinhalten mit wenig oder gar keinem Programmcode.

Für den Zugriff auf die Microsoft-Datenbank SQL Server stehen eigene, speziell angepasste Klassen zur Verfügung, auf andere Datenbanken wird über OLEDB/ODBC zugegriffen.

Webentwicklung mit ASP.NET

Auch die Webentwicklung mit ASP (*Active Server Pages*)-Technik wird durch .NET neu gestaltet. Die Architektur ist jetzt wesentlich modularer und basiert auf Komponenten. Im Gegensatz zum „klassischen“ ASP wird der Programmcode kompiliert (nicht interpretiert). Das ereignisbasierte Programmiermodell ähnelt dem Programmieren mit Visual-Basic-Formularen.

Die *WebForms*-Klassen der .NET-Klassenbibliothek und die Werkzeuge (Assistenten) in der Entwicklungsumgebung Visual Studio stellen Unterstützung für das Erstellen von Webseiten zur Verfügung, wobei sowohl in den HTML-Quelltext eingebetteter Visual-Basic- oder C#-Code als auch „Code-Behind“ verwendet werden kann. Letzteres bedeutet, dass der Programmcode von der eigentlichen Seite getrennt in einer eigenen Datei steht und über Er-

eignisse mit Steuerelementen (Eingabefelder, Schaltflächen usw.) im HTML-Dokument (typischerweise ein Formular) verbunden sind. So ist eine saubere Trennung von Design und Programmlogik gegeben. Die Steuerelemente sind dann keine gewöhnlichen HTML-Tags, sondern spezielle ASP-Tags, die auf dem Server (durch den Webserver) ausgeführt werden. (In Java kennt man ähnliche Techniken unter den Bezeichnungen *Tag-Libraries* und *Presentation-Frameworks*).

Für Standardaufgaben wie Eingabefelder mit Gültigkeitsprüfung, Anzeigen und Bearbeiten von Datenbankinhalten usw. stehen fertige Serversteuerelemente zur Verfügung. Aber auch eigene Komponenten und Steuerelemente können problemlos entwickelt und werkzeuggestützt in die Seiten eingebunden werden.

Und Java?

Da die .NET-Technologie gerade im Bereich Webanwendungen im Enterprise-Umfeld auf ähnliche Problemstellungen zielt wie Java (Java 2 Enterprise Edition, J2EE), und auch ähnliche Techniken zur Lösung dieser Probleme anbietet, ist eine Konkurrenz unbestreitbar. Durch die Integration in künftige Windows-Betriebssysteme hat .NET sicherlich einen nicht zu unterschätzenden Wettbewerbsvorteil.

Vergleich zwischen J2EE und .NET					
	Herkunft	Plattform	Präsentationsschicht	Geschäftslogik	Datenbankzugriff
J2EE	Spezifikation Sun Microsystems, Produkte von verschiedenen Herstellern, z. B. IBM, Bea	praktisch alle	Servlets/JSP	Java Enterprise Beans (EJB)	JDBC
.NET	Microsoft	Windows	ASP.NET	(Serviced) Components	ADO.NET

Der Hauptvorteil von Java ist die Plattformunabhängigkeit (und implizit Herstellerunabhängigkeit): .NET-Programme laufen ausschließlich unter Windows, Java-Programme auf einer Vielzahl von Architekturen und Betriebssystemen. Diese Plattformunabhängigkeit hat aber auch ihren Preis: Wo .NET-Programme unmittelbar auf Betriebssystemdienste zugreifen können (z. B. für Transaktionen, Messaging, usw.), benötigen Java-Enterprise-Anwendungen einen Application Server als Container zur Bereitstellung von Infrastruktur. Jedes Abstraktionslayer kostet letztendlich Performance.

Mehrschichtige Anwendungen

Kommerzielle Webanwendungen werden in der Regel in mehrere Schichten aufgeteilt: Die Präsentationsschicht (Benutzeroberfläche) ist für die Interaktion mit dem Anwender zuständig, während im Hintergrund die Datenschicht mit einer Datenbank kommuniziert. Dazwischen findet sich häufig eine Ebene, welche die eigentliche Anwendungslogik – die Geschäftsprozesse – darstellt. Dieser Business-Schicht kann noch eine „Fassade“ vorgeschaltet sein, welche es ermöglicht, dass Kunden und Geschäftspartner mit den Geschäftsobjekten arbeiten, ohne Zugriff auf Interna zu haben. Weitere Schichten sind möglich.

Mehrschichtige Anwendungen verbessern die Skalierbarkeit (Anpassung an wachsende Anforderungen, z. B. steigende Benutzerzahlen) und die Lastverteilung (Load Balancing).

Die Präsentationsschicht wird bei .NET mittels ASP .NET realisiert, bei Java mit Servlets oder JSP (Java Server Pages).

- JSP-Webanwendungen laufen in einem in einem JSP-Container, der Bestandteil des Application Servers ist.
- ASP .NET-Webanwendungen werden direkt durch den Microsoft-Webserver IIS (Internet Information Services) ausgeführt und nutzen dessen Dienste z. B. für SSL, Authentifizierung und Autorisierung.

Mehrschichtige Java-Anwendungen verwenden meist Enterprise Beans (Session Beans/Entity Beans), welche wiederum einen speziellen Container benötigen – auch dieser ist Bestandteil des Application Servers. Der Container ist z. B. verantwortlich für Transaktionen und Persistenz (Speicherung).

Bei .NET kann Geschäftslogik durch selbst entwickelte Komponenten realisiert werden, welche auf Betriebssystemdienste zugreifen. Dies sind insbesondere die so genannten COM+-Dienste (Enterprise Services) u. a.:

- Microsoft Transaction Server
- Microsoft Message Queue

Datenbankzugriffe werden unterstützt durch vorgefertigte Komponenten von ADO.NET.

Vergleich

Ein direkter Vergleich fällt schwer, einerseits, da im Java-Umfeld ein breites Spektrum an Produkten (zu dramatisch unterschiedlichen Preisen) zur Verfügung steht, andererseits, weil bislang eher wenige kommerzielle Anwendungen mit .NET realisiert sind.

Eines soll jedoch hier noch einmal deutlich herausgestellt werden: .NET läuft ausschließlich auf Windows-Plattformen. Zwar existiert eine Vorzeigimplementierung für das Unix-Derivat FreeBSD, diese darf aber ausdrücklich nicht für kommerzielle Zwecke eingesetzt werden. DotGnu (www.dotgnu.org) und das Mono-Projekt (www.go-mono.com) aus der OpenSource-Bewegung haben sich eine Portierung auf Linux zum Ziel gesetzt, die jedoch bislang noch unvollständig ist und auch mittelfristig kaum für Produktionszwecke einsetzbar sein wird.

Wer auf .NET setzt, legt sich also auf Windows fest. Durch die Möglichkeit, via Webservices mit anderen Anwendungen auch auf anderen Plattformen zu kommunizieren, wird dieser Nachteil etwas abgemildert. Und fairerweise muss hinzugefügt werden, dass auch Java-

Application-Server untereinander weniger kompatibel sind, als die Werbung verspricht – wenn einmal ein bestimmtes Produkt gewählt wurde, ist man hier also ebenfalls von einem Hersteller abhängig.

Jeder Vergleich zwischen .NET und J2EE muss auch Kostenaspekte umfassen. Java selbst ist kostenlos, und wenn man Linux verwendet, gilt das auch für das Betriebssystem. Entwicklungswerkzeuge sind ebenfalls frei verfügbar (wenn auch nicht alle von hoher Qualität). Hinzu kommen aber in der Regel erhebliche Kosten für Application-Server, die dann im Komplettpaket auch professionelle Entwicklungswerkzeuge mitbringen.

.NET selbst ist ebenfalls kostenfrei – nicht jedoch das Windows-Betriebssystem. Und kommerzielle Anwendungen wird wohl kaum jemand mit dem Framework SDK allein entwickeln, die Kosten für Visual Studio müssen also hinzugerechnet werden. (Im Hinblick auf objektorientiertes Design genügt Visual Studio mit Visio nicht allen professionellen Ansprüchen, da sind eventuell zusätzliche Tools notwendig.) Ein Application Server wird dagegen nicht benötigt; diese Funktion übernimmt, wie bereits erwähnt, das Betriebssystem.

Fazit

.NET ist angetreten, um in der Oberliga der Enterprise-Anwendungen mitzuspielen und will sich folglich auch an den Spitzenprodukten im Java-Umfeld messen lassen. Im technischen Vergleich schneidet .NET nicht schlecht ab, wenn es auch letztlich gegenüber Java wenig wirklich Neues bietet. Bei künftigen Entscheidungen für die eine oder die andere Technologie werden sicherlich nicht nur die reine Leistungsfähigkeit eines Produkts, sondern auch Aspekte wie vorhandenes Know-How (VB6/VisualC++/Windows oder Java/Linux), vorhandene Betriebssysteminfrastruktur/gewünschte Plattform, Werkzeugunterstützung usw. eine Rolle spielen.

Dr. Susanne Wigard
Telefon: 0211 9449-6013
E-Mail: susanne.wigard@lds.nrw.de

Programmierung mit dem VisualAge Generator im Projekt OBELIX

Römer und Software

Viele von uns erinnern sich an die berühmte Comic-Serie „Asterix und OBELIX“, gezeichnet von Albert Uderzo, die originellen Texte stammten von René Goscinny. Neben Asterix war OBELIX „der Dicke“ ständig hungrig auf Wildschweinbraten, Kultfigur und kampfbereit.

Was hat nun OBELIX mit Software zu tun?

Statt Wildschweinbraten füttern wir OBELIX mit Zahlfällen und „hinten heraus“ kommt Geld – ein wahrer Goldesel? Lax gesagt: Ohne **BE**züge **L**äuft **nIX**. – oder korrekt formuliert **O**ne **BE**zügeverfahren des **L**andes **N**ordrhein **W**estfalen mit **I**nterner und **eX**terner Unterstützung.

Jetzt ist klar – OBELIX steht für das neue Bezügeverfahren des Landes NRW.

Mit dem 4GL-Tool „VisualAge Generator“ aus dem Hause IBM entwickelt – in Zusammenarbeit mit einer Design- und Test-Abteilung – ein 19-köpfiges Programmerteam des LDS NRW eine moderne, webbasierende Lösung für das Landesamt für Besoldung und Versorgung (LBV).

VisualAge Generator – Was sagt IBM dazu?

Der einfache Weg zum e-Business; VisualAge Generator – Preisträger des OO/4GL-Award

VisualAge Generator, im folgenden Text mit VAGen bezeichnet, kann eine Vielzahl unterschiedlicher Applikationsarten erzeugen; dazu gehören auch die Web-Applikationen. Dieser Artikel beschränkt sich ausschließlich auf die Web-Transaktionen, da OBELIX nur diese Technologie verwendet. Die Abbildung 1 zeigt ein 3-Schichten-Modell. Schicht 1 ist der Web-Browser, Schicht 2 ein Web-Applikations-Server, der zwischen Browser und gestartetem Programm vermittelt. Schicht 3 stellt die Server-Plattform dar, in der die generierten Programme Daten von Schicht 2 empfangen, verändern und zur Schicht 2 zurücksenden. Diese drei Schichten befinden sich gewöhnlich auf separaten Maschinen, wobei die Funktionalität der Schicht 2 und 3 über mehrere Rechner verteilt werden kann.

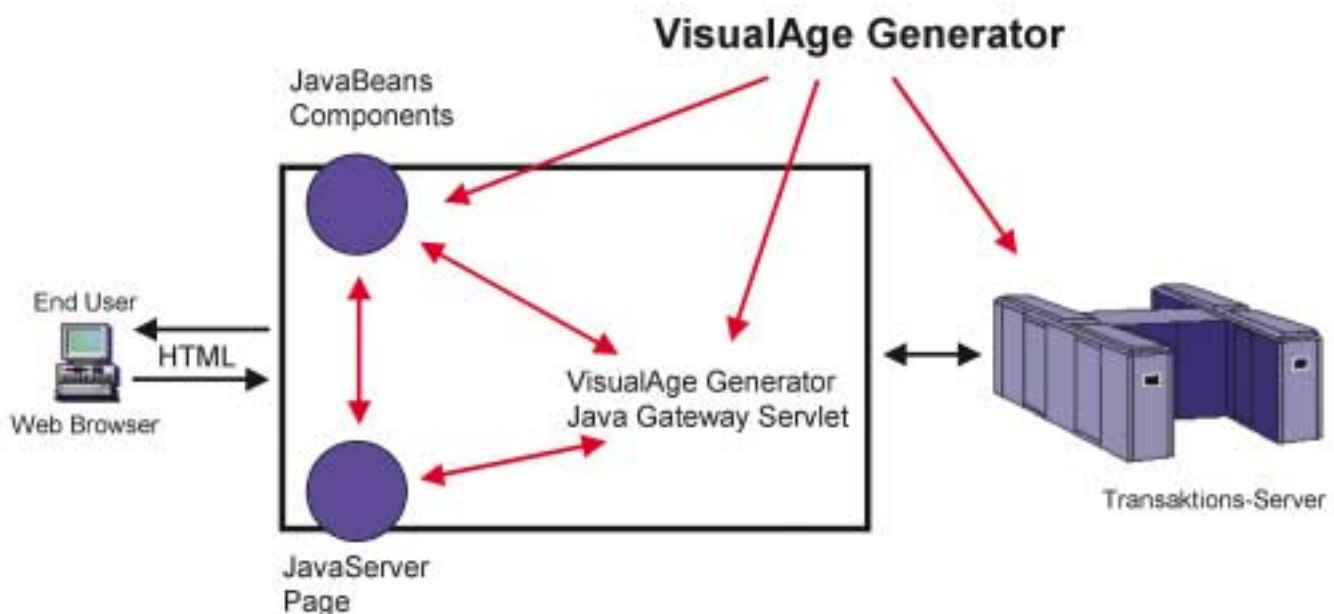


Abb. 1: e-Business mit RAD

Highlights des VAGen

- Entwicklung und Test von webbasierten Client/Server-Applikationen für Stand-Alone- oder Workstation-Systeme
- Interaktives Screen-Design für GUI und TUI
- RAD/Rapid Application Development
- High-Level Spezifikationen und 4GL-Sprache
- Zugriff auf relationale, hierarchische, VSAM und andere traditionelle Datenhaltungssysteme
- Interaktiver Syntax-Check und Trace
- Ausführung in mehrschichtigen und wechselbaren Umgebungen

VAGen ist ein High-End-Systementwicklungswerkzeug für die Erstellung von e-Business, mehrschichtigen oder textbasierenden Anwendungen. Es bietet vielseitige Lösungen unter Ausnutzung von skalierbaren mehrschichtigen Plattformen bei vernetzten Systemen. Das in hohem Grade produktive Entwicklungstool VAGen vereinfacht die Komplexität aller unterstützen Topologien durch Abschirmung dieser gegenüber dem Entwickler. Dies ermöglicht den Entwicklern mit wenig oder keinen JAVA-Kenntnissen „end-to-end“ JAVA-e-Business-Systeme zu implementieren. Außerdem bietet es objektorientierten Java-Entwicklern auch traditionelle Transaktionsplattformen, wie z. B. CICS oder MVS, ohne spezielle Kenntnisse im Mainframe-Bereich (z. B. DL/I, VSAM) mit einzubeziehen.

Integrierte Java-Entwicklungsumgebung

VAGen unterstützt die Entwicklung von Internet- oder Intranet-Anwendungen durch Java-Clients wie Applets, Servlets und Java-GUIs, sowie die Möglichkeit, neue transaktionsbasierende Serverprogramme herzustellen. VAGen ist völlig in das IBM VisualAge for Java oder IBM VisualAge Small-

talk integriert, wodurch es ermöglicht wird, in VAGen-Anwendungen Hunderte von Komponenten, welche von IBM oder aus einer dritten Entwicklerquelle stammen, wiederzuverwenden. Um die visuellen Programmkomponenten zu vervollständigen, hat der Programmierer die Möglichkeit, Code in 4GL, in Java oder in beiden Sprachen zu schreiben.

Es soll einfach sein!

VAGen erzeugt automatisch Java-Code für die mittlere Schicht und steuert eigenständig die Datenkonvertierung, das Statusmanagement und die Serververbindung. VAGen erstellt ebenso den transaktionsspezifischen Server-Code. Unter Verwendung des Interaktiven Testtools des VAGen und einem Web Browser ist es möglich, auf derselben Workstation leicht und einfach die bisher entwickelte Software zu testen. Somit wird es überflüssig, für jeden Entwickler eine volle n-Schichten Entwicklungsumgebung aufzubauen und beschleunigt so den Entwicklungszyklus.

Es soll schön aussehen!

VAGen erzeugt automatisch eine Default JSP für das Server-Programm und Default HTML für den End-User Browser. Diese JSP-Komponenten können anschließend von einem Web-Designer nach bestimmten Gesichtspunkten verbessert werden. Eine spezifische Struktur, der User-Interface-Records, ermöglicht die Trennung von Entwicklung und Design. Somit sind diese unabhängig voneinander.

Es soll heterogen sein!

VAGen Anwendungen können auf den folgenden Plattformen laufen: Web Browser, Win NT, Win 95, Win 98, IBM OS/2, IBM OS/400, IBM AIX, HP UX, IBM VSE, IBM VM, OS/390 und Sun Solaris. VAGen optimiert Java-Code für Client-Web-Application-Server, Win 2000, Win NT sowie COBOL und C++ Code für Server Programme.

Anwendungen können vollautomatisch, ohne dass der Quellcode geändert werden muss, für unterschiedliche operative Umgebungen generiert werden.

Workbench – die grafische Oberfläche des VAGen

Sämtliche Objekte, seien es Programme, Funktionen, Records oder Items werden zentral in einem Serverrepository (IDE – Integrated Development Environment) verwaltet. In diesem Repository legt der Administrator Projekte an, die in einzelne Packages unterteilt werden. Diese wiederum enthalten die zugehörigen Funktionen, die so genannten Types.

Eine volle Versionskontrolle, Vergleichen unterschiedlicher Stände, Einladen älterer Programmversionen, gleichzeitiges Entwickeln der Software im Team zeichnen dieses mächtige Tool aus. Die Versionierung der Types, Packages und Projekte werden in der IDE durchgeführt. Ebenso wird hier die User- und Rechteverwaltung an den einzelnen Objekten durchgeführt. Über eine PLP-Verwaltung (Project-List-Part) können versionsabhängig eine ganze Gruppe von Objekten zusammengehalten werden.

Ausblicke auf die neue Version – WebSphere Studio Enterprise Edition

Die IBM hat seit kurzem ihre Produkte bestimmten Bereichen zugeordnet und damit auch die Namensvergabe neu gestaltet. Sämtliche Entwicklungstools gehören jetzt zum Bereich WebSphere.

So muss auch der Name VisualAge aus unseren Köpfen weichen und Platz machen für die Zukunft: WebSphere Studio.

Die Umstellung ist im vollen Gange. Schon Ende 2002 ist der Support für den VAGen auf einigen Umgebungen

und von VAGen-verwandten Produkten eingestellt worden. Die im Projekt OBELIX eingesetzte Version vom VisualAge for Java (VAJava) für Windows NT wird weiterhin bis Ende 2004 unterstützt. Ebenfalls die VAGen-Host-Services, die für die Verarbeitung von VAGen-Programmen auf dem Host benötigt werden.

nik und die Administration verantwortlich und für alle gibt es noch einen Teamleiter. Das Programmiererteam gehört zum insgesamt 55 Mitarbeiter starken Teilprojekt DV-Umsetzung, zu dem auch die Bereiche Teilprojektleitung, Systemarchitektur, Datenbank (DB2) und Design gehören.

grammierung wird diese Aufgabe für den VAGen und ClearCase (Versionierungstool) erfüllt.

Und zu guter Letzt wird vom Programmiererteam noch ein Großteil der Pflege und Darstellung des Intranetauftritts vom Projekts OBELIX gewährleistet.

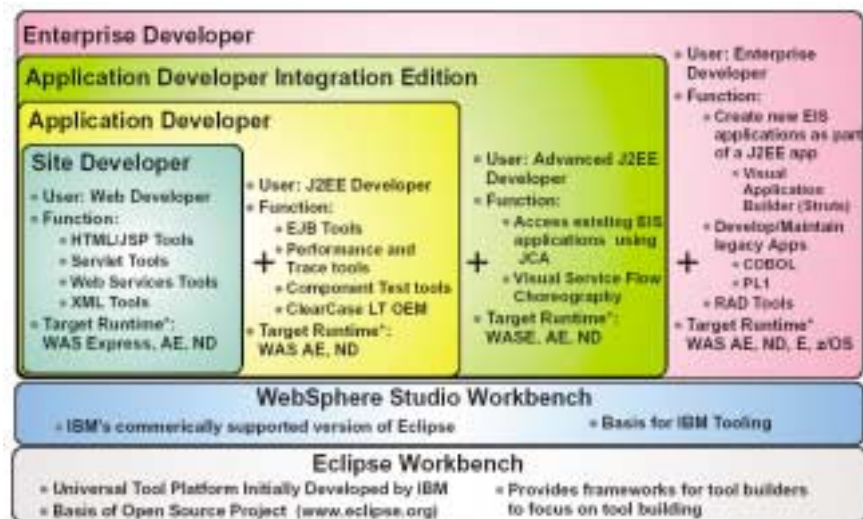


Abb. 2: Ausblick WebSphere Studio

Zuerst muß die Eclipse Workbench auf dem Computer vorhanden sein. Diese Plattform wurde von IBM für ein Open Source Projekt entwickelt. An diesem Projekt nehmen verschiedene Hersteller teil. Der Sinn dieser Plattform ist es, unterschiedliche Tools von verschiedenen Herstellern für spezifische Teilaufgaben verwenden zu können. Jedes Tool muss als Plug-In in die Eclipse-Workbench integriert werden.

Damit lässt sich dann z. B. auch die Versionsverwaltung mit dem Produkt ClearCase von Rational durchführen.

„Ein Jahr im Steinbruch von OBELIX“

Das Programmiererteam

Das Programmiererteam besteht aus 19 Mitarbeitern, wovon sich 16 auf drei Programmiergruppen aufteilen. Zwei weitere Mitarbeiter sind für die Tech-

Neben dem Teilprojekt DV-Umsetzung gibt es im Projekt OBELIX noch die Teilprojekte Erstellung des fachlichen Feinkonzepts, Migration und Test.

Die Hauptaufgabe des Programmiererteams ist die Programmierung des neuen Bezügeverfahrens, angefangen vom Dialog und der Migration über die Berechnung bis hin zur Ergebnisverarbeitung. Eine weitere Aufgabe ist es, Auswertungsfunktionen und Bearbeitungstools für den eigenen Bedarf als auch für andere Projektbereiche zu erstellen.

Daneben gibt es noch die Pflege und Administration der Systeme wie CICS, CTG, WebSphere und Http-Server, die für das Bezügeverfahren benötigt werden. Zu diesem Aufgabenbereich gehört auch die Bereitstellung der Software auf den 8 Ausführungsumgebungen. Vier Umgebungen werden für den Test, eine für das Design und drei für die Programmierung zur Verfügung gestellt. Im Projekt OBELIX gibt es in jedem Bereich mit spezieller Software eine eigene Administration. In der Pro-

Entscheidung für den VAGen

Zwei Gründe für die Entscheidung, den VAGen für die Programmierung des neuen Bezügeverfahrens einzusetzen: Die Mitarbeiter, die das alte Bezügeverfahren warten, sollen in Zukunft auch bei der Pflege und Wartung des neuen Bezügeverfahrens mitwirken. Da das alte Bezügeverfahren in den prozeduralen Sprachen Cobol und Assembler geschrieben ist und der Generator ebenfalls mit einer eher prozeduralen 4GL-Sprache arbeitet, ist die Art der Programmierung ähnlich. Die für den Web-Bereich der Anwendung notwendigen Programme in der objektorientierten Sprache Java werden vom Generator aus dem in der 4GL-Sprache geschriebenen Code erzeugt.

Der zweite Grund für den VAGen ist die Möglichkeit, die fertige Anwendung für verschiedene Hardware-Plattformen zu generieren, sowie die Möglichkeit, die Komponenten unterschiedlich auf die jeweiligen Schichten zu verteilen. Hierdurch kann mit relativ geringem Aufwand auf mögliche hardware-bedingte Performance-Probleme reagiert werden. Der Weg ist offen, die Anwendung für andere Organisationen auf einer zweiten, technisch anders getarteten Plattform zu installieren.

Infrastruktur der Entwicklungs- und Zielumgebung

Die Abbildung 3 zeigt eine prinzipielle Darstellung der Entwicklungsumgebung im Projekt OBELIX und besteht aus zwei Systemen, dem Host- und dem NT-System.

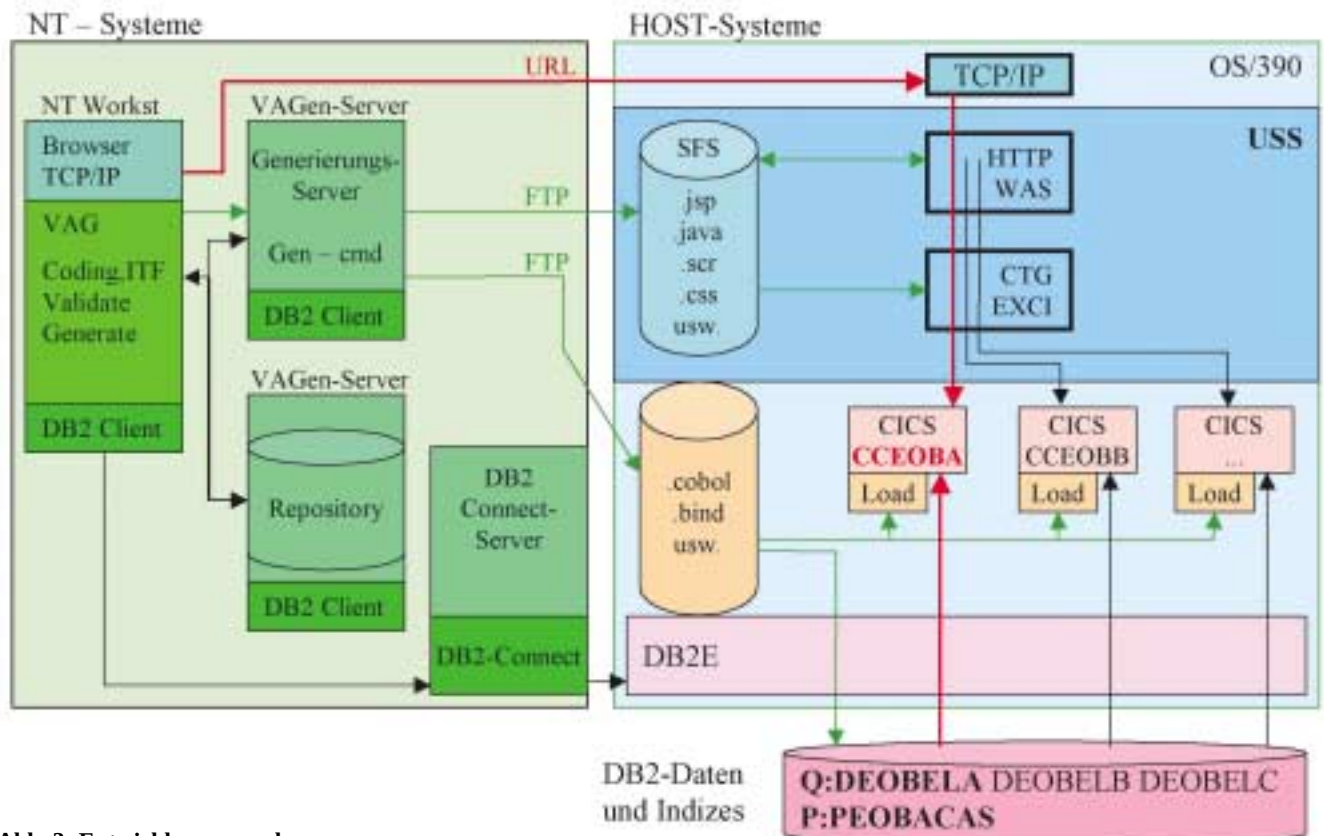


Abb. 3: Entwicklungsumgebung

Die erforderlichen Komponenten einer NT-Workstation sind ein Web Browser, der VAGen und ein DB2-Client.

Die Programmierung im VAGen umfasst die Codierung, den Test und die Fehlersuche im ITF (Interactive Test Facility) sowie die Validierung und die Generierung des Sourcecodes. Ein DB2-Client ermöglicht den Zugriff über DB2-Connect auf Daten und Datenstrukturen in DB2.

Für die zentrale Speicherung existiert ein Repository. Hier werden der Programm-Code, die Record- und die Userinterface-Definitionen in Paketen und Projekten zusammengefasst und gespeichert. Pakete und Projekte werden in Versionen fortgeschrieben, so dass ein Zugriff auf ältere Versionen jederzeit möglich ist.

Generierung (grüne Pfeile)

Im Projekt sind für die verschiedenen Teilbereiche, wie z. B. Entwicklung

oder Test, mehrere Arbeitsumgebungen (je CICS und Batch) notwendig. Die Generierung erfolgt für jede Arbeitsumgebung gesondert.

Ein Generierungskommando startet die Generierung auf dem Generierungsserver. Die benötigten Daten werden vom Repository-Server angefordert.

Auf dem Generierungs-Server wird der Source-Code erzeugt und im Zuge der Preparation per FTP auf den Host übertragen und dort in ausführbaren Code umgewandelt.

Zugriff auf die Anwendung (rote Pfeile)

Der Zugriff auf die Anwendung wird über den Browser durch die URL gesteuert. Die URL hat einen bestimmten Aufbau und enthält folgende Informationen:

- Ziel-Server
- Welcher Port wird auf dem Ziel-Server benutzt?

- Welches Programm soll auf dem Ziel-Server ausgeführt werden?
- evtl. ergänzende Daten

Der Browser verbindet sich mit dem Webserver über TCP/IP und stellt Anfragen über HTTP an den Websphere Application Server (WAS). Im WAS werden in der JVM die Java-Klassen ausgeführt und mit dem CICS-Transaction-Gateway (CTG) über das External-CICS-Interface (EXCI) auf die entsprechende CICS-Umgebung zugegriffen.

Die CICS-Anwendung schickt ihr Ergebnis zurück an den WAS. Der WAS interpretiert das Ergebnis mit der zugehörigen JSP und baut daraus die HTML-Seite zusammen und schickt diese an den Browser.

Unser Programmierprozess

Im organisatorischen Umfeld ist festgelegt, wie der Programmierer bei der Programmerstellung vorzugehen, was

er dabei alles zu beachten, nicht zu beachten und welche Ergebnisse er abzuliefern hat. Richtlinien sorgen für einen hohen Grad an Standardisierung, Wiederverwendbarkeit, Lesbarkeit und damit auch für die Wartbarkeit der Programme in der Zukunft.

Es existieren hierzu zwei Regelwerke:

Zum einen die in OBELIX-Codiererkreisen inzwischen fast schon legendär gewordene StauRi (Standards und Richtlinien-Dokument), die festlegt, wie die einzelnen Programme auszusehen haben. Hierin finden sich beispielsweise Namenskonventionen, Fehlerbehandlung, Behandlung von Funktions- und Programmaufrufen usw.

Neben der StauRi gibt es die Prozess-Beschreibung, eine Art Checkliste, die den Ablauf eines Programmierprozesses und die zu erstellenden Ergebnisse definiert.

Ein solcher Programmierprozess könnte (stark vereinfacht) wie folgt aussehen:

- a) Beauftragung: Das Designteam übergibt dem PGM-Team das komplette Design eines Paketes, das aus mehreren Programmvorgaben, Makrostrukturen und einer Auflistung der verwendeten Relationen besteht. Dieses Paket wird nun einem so genannten Tracker übergeben, der für die weitere Abarbeitung verantwortlich ist.
- b) Vorbereiten: Vorgaben werden gesichtet, Daten und Datenrecords auf ihr Vorhandensein überprüft, die Programmierungsumgebung, das Projekt, das Package eingerichtet und die zu verwendenden Projekte nachgeladen. Zusätzlich werden die Testdaten vom Testteam angefordert und auf Vollständigkeit und Korrektheit überprüft.
- c) Programmieren und Dokumentieren: Handelt es sich bei dem anzufertigenden Programm um ein Standardprogramm (Datenaufbereitung, Stamm-

datenpflege, Stammdatenübersicht usw.), so kann ein entsprechender Prototyp in die Workbench eingeladen und angepasst werden. Die Programmierung in der VAGen-Umgebung unterscheidet sich kaum von der in anderen grafischen Programmierungsumgebungen. Programmierung und Dokumentation gehen hier Hand in Hand. Mit Hilfe eines selbst erstellten Tools (StauRichter) kann jeder Programmierer sein Programm auf Einhaltung der Regeln aus der StauRi überprüfen.

- d) Testen: Zur Programmierung gehört auch ein Modultest auf Basis der White-Box-Methode. Dazu ist es erforderlich, das für jedes Modul ein Testtreiber geschrieben wird. Mit einem weiteren speziell hierfür entwickelten Werkzeug werden die nötigen Testeingabedaten erstellt und verwaltet.
- e) Abschließen: Alle Ergebnis-Dokumente sind nun in die entsprechenden Verzeichnisse zu stellen und der Programmabschlussbericht ist zu verfassen. Das Package und das Projekt werden versioniert. Ein PLP-File (Project-List-Part), welches das eigene und alle verwendeten Projekte aufführt, wird erstellt, um mit ihm dann das fertige Programm zu generieren.

Was sagen WIR zum VAGen?

Verwalten

Projekte sind die oberste Strukturebene im VAGen. Hier wurde für OBELIX eine breite Gliederung nach fachlichen und technischen Gesichtspunkten gewählt. Die fachlich orientierten Projekte enthalten jeweils zu einem Vorgang alle Programme für eine Softwarekomponente wie zum Beispiel den Änderungsdienst oder die Stammdatenpflege. Hier bilden die Packages eine Aufteilung in einzelne Programme. Die technisch orientierten Projekte beinhalten zum Beispiel die DB2-Zugriffsschicht. Falls fachlich oder technisch

notwendig, können diese Projekte noch in mehrere Packages aufgeteilt sein.

Programmieren

Sprachumfang

Der Hauptvorteil der Programmiersprache VAGen besteht darin, Programme in verschiedenen anderen Programmiersprachen, etwa in JAVA, C++ oder COBOL, und für unterschiedliche Betriebsumgebungen generieren zu können. Eine VAGen-Programmsource könnte etwa als C-Programm unter UNIX, als COBOL-Programm in MVS und/oder in Windows auf der JAVA-VM laufen. Dieser Vorteil einer hohen Portierbarkeit wird jedoch erkauft durch den Nachteil eines relativ beschränkten Sprachumfangs. Da es offenbar einfacher ist, aus einer technisch weniger entwickelten Programmiersprache eine technisch höherstehende zu generieren als umgekehrt, richtete man sich beim Design von VAGen nach dem kleinsten gemeinsamen Nenner aller oben genannten Programmiersprachen.

Datenstrukturen

Wie in COBOL wird in VAGen mit Records gearbeitet. Sie bieten den Vorteil, Daten hierarchisch zu strukturieren und ganze Datenblöcke auf einmal bewegen zu können. Zwar dürfen in VAGen auch Variablen einzeln deklariert und benutzt werden, diese sind jedoch nur in den jeweiligen Funktionen gültig und werden daher in der Regel nur als Schleifenzähler, Hilfsfelder oder Übergabeparameter für Funktionen verwendet. Für alles wirklich Wichtige – seien es nun Tabellenzugriffe, Aufrufe von Unterprogrammen oder die Darstellung von Bildschirmmasken – benötigt man Records. Diese haben immer eine feste Länge, so dass Daten nicht dynamisch erzeugt und verwaltet werden können. Daher gibt es im VAGen keine Zeiger, keine dynamischen Speicherstrukturen wie Queues, Listen und dergleichen. Auch bei den einzelnen Datentypen hat man sich an das in

COBOL Übliche angelehnt und auch die feste Länge übernommen.

Ablaufstrukturen

VAGen besitzt selbstverständlich alle Sprachelemente, um ein strukturiertes Programm zu erstellen: Verzweigungen (IF), Schleifen (WHILE), Funktions- und Unterprogrammaufrufe. Funktionsaufrufe können nicht innerhalb einer Anweisung verschachtelt werden und sie können auch nicht in einer Bedingungsanweisung stehen. Ebenfalls nicht möglich in VAGen sind solche Konstrukte wie IF (LEN(A)>5) THEN oder X ABC(LEN(A)), was den Programmcode je nach Problemstellung ziemlich in die Länge zieht.

Schnittstellen

Ein weiterer Vorteil von VAGen sind seine Schnittstellen und deren recht einfache Handhabung. Der Zugriff auf Datenbestände (DB2) ist in die Workbench integriert und erfolgt über spezielle Zugriffsfunktionen. Da der Zugriff nur über Records möglich ist, gibt es auch hier keine Dynamik, keine Möglichkeit also, Tabellenzugriffe in Abhängigkeit des Programmablaufs zu definieren.

Als letztes Highlight ist die Anbindung ans Web zu nennen. Ohne Java-Kenntnisse lässt sich unter VAGen eine voll Web-fähige Anwendung erstellen, die sich über die Workbench mit jedem gängigen Browser direkt testen lässt.

VAGen-Hilfen

Wie alle anderen Entwicklungsumgebungen bietet auch der VAGen dem Programmierer vorgefertigte Hilfswerkzeuge. Die wichtigste Hilfe, die eine Programmiersprache bieten kann, sind sicherlich vorgefertigte Standardfunktionen. Im Generator können diese einfach über den Schalter „Statement-Templates“ in kompletter Syntax in den Quelltext eingefügt werden. Anzahl und Reihenfolge der geforderten Parameter werden mit dem Statement in den Quelltext eingefügt.

Standardfunktionen werden im VAGen „EZE-Funktionen“ genannt. Hier gibt es Funktionen zur Stringbehandlung, für mathematische Probleme usw. Für uns sind im Projekt OBELIX allerdings häufig Funktionen zur Berechnung von Zeiträumen von Nutzen. Leider bietet die Entwicklungsumgebung in diesem Bereich nicht viel.

Der Editor des VAGen lässt etwas zu wünschen übrig. Syntaxhervorhebungen, die bei guten Editoren automatisch durchgeführt werden, fehlen komplett. Optische Hervorhebungen der Programmstrukturen sind nur sehr eingeschränkt vorhanden.

Allerdings können während des Programmierens Kontextmenüs aufgerufen werden, um Funktionen, Rekordvariablen usw. in den Code einzufügen. Dies ist sehr praktisch, um bei der Vielzahl der im Projekt benutzen Variablen, den Überblick zu behalten und Schreibfehler zu vermeiden. Eine zentrale VAGen-Hilfe stellt die umfangreiche „Validate“-Funktion dar, die den kompletten Syntaxcheck durchführt.

Die Anlage eines SQL-Records ist im VAGen sehr komfortabel. Über eine Schnittstelle ist der VAGen mit der Datenbankumgebung verbunden, wodurch die Möglichkeit besteht, die vorhandenen Record-Strukturen in die Arbeitsumgebung zu übertragen.

Bei der Programmierung von Datenbankzugriffen bietet der ausgezeichnete SQL-Statement-Editor-Hilfe. Die Vorgabe der zugrundeliegenden DB2-Tabelle sowie die gewünschte Zugriffsart reichen aus, um eine Standard SQL-Anweisung zu generieren, welche noch angepasst werden muss. Das so entstandene SQL-Statement lässt sich direkt gegen die Datenbank validieren. Über ein Kontextmenü wird ein aufgabenspezifischer Error-Handler eingebunden. Auf diese Weise kann sehr leicht gesteuert werden, wie das Programm sich im Fehlerfall verhalten soll.

Alle projektweit genutzten Variablen (u a. Datenbankfelder) sind als „shared“ definiert. Dies ermöglicht ein hohes Maß an Sicherheit bei der Änderung von Eigenschaften einer Variablen. Wird der Typ oder die Länge eines Feldes geändert, wirkt sich dies automatisch auf jede Stelle im Projekt aus, die dieses Feld benutzt. Eine manuelle Nachbehandlung ist nicht mehr erforderlich.

Selbst erstellte Hilfen

Wie bereits weiter oben erwähnt, fehlen dem Generator einige wichtige Standardfunktionalitäten, wie z. B. Datumsfunktionen. Eine Abhilfe schaffen hier die mittlerweile in einer großen Vielzahl selbst erstellten Funktionen, die einmal programmiert, in allen Modulen benutzt werden können und in einem zentralen VAGen-Projekt verwaltet werden. Diese Vorgehensweise ermöglicht ein maximales Maß an Flexibilität bei späteren Änderungen einer Funktionalität.

Oftmals existieren ähnliche Typen von Programmen, die zwar in ihrem Aufbau gleich sind, jedoch in ihrer Funktionalität Unterschiede aufweisen. Um die Arbeit zu vereinfachen haben wir zu jedem dieser Programmgruppen Prototypen erstellt. Diese können in einem Text-Format durch „Suchen“ und „Ersetzen“ an die jeweiligen Bedürfnisse angepasst werden. Somit hat der Programmierer schon mal ein Grundgerüst, in das er die programmindividuellen Funktionalitäten einpflegen kann.

Eine weitere wichtige Hilfe bei der täglichen Arbeit ist unser selbst erstellter „StauRichter“. Dieser überprüft den Quellcode daraufhin, ob der Programmierer gegen eine in den Standards und Richtlinien festgelegte Regel verstoßen hat.

Ein Rundflug über die Bearbeitungsmasken

Die Workbench oder das Main Control Center ist sozusagen das Herzstück der Entwicklungsumgebung. Hier sind die eingeladenen Projekte und deren Unterstrukturen (Packages/Types) ersichtlich.

Die jeweiligen „Besitzer“ der ausgewählten Projekte, Packages und Types sind im unteren Teil der Workbench ebenfalls zu sehen.

Im VAGen-Parts-Browser werden alle Packages und die zugehörigen VAGen-Parts angezeigt, die sich in den eingeladenen Projekten befinden. Nur von hier aus können die einzelnen Parts angezeigt und editiert werden.

Im Funktions-Editor werden auch die Datenzugriffe realisiert, wie in dem folgenden Beispiel ersichtlich ist.

Der Programm-Editor gibt die hierarchische Struktur des ausgewählten Programms wieder und stellt die benutzten Tables und Called-Parameters dar. Daran schließt sich die Programmstruktur mit ihren Funktionen und Unterfunktionen an.

Webtransaktion

Die Web-Seiten werden von dem Design schon als ESF-File (Extended Source Format des VAGen) an die Programmierung übergeben. Dieses ESF-File enthält Informationen für einen UI-Record, der eine VAGen-spezifische Definition für eine Web-Seite darstellt und in den VAGen importiert werden kann. Der UI-Record wird von der Programmierung um Felder für die Kommunikation zwischen zwei Programmen ergänzt und anschließend als JAVA-Serverpage generiert. Die unformatierte JAVA-Serverpage wird anschließend vom Web-Designerteam bearbeitet. Die fertige Web-Seite wird von der Programmierung je nach Anforderung auf die entsprechende Ausführungsumgebung übertragen.



Abb. 4: Die Workbench

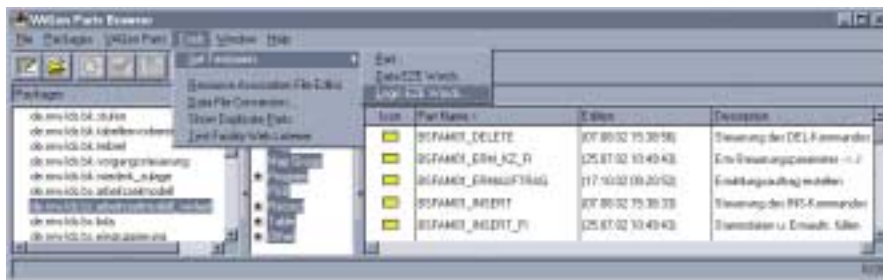


Abb. 5: Der Parts Browser

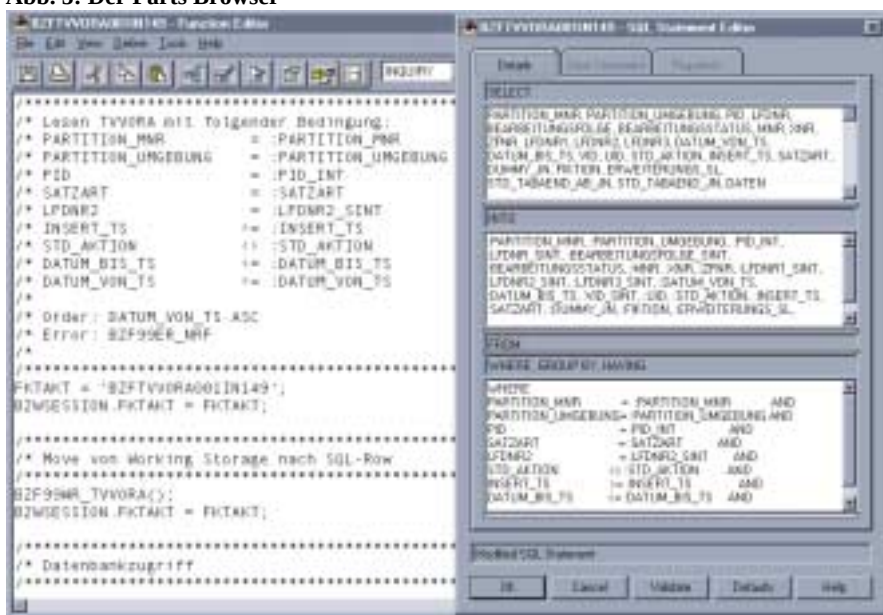


Abb. 6: Funktion-Editor-SQL

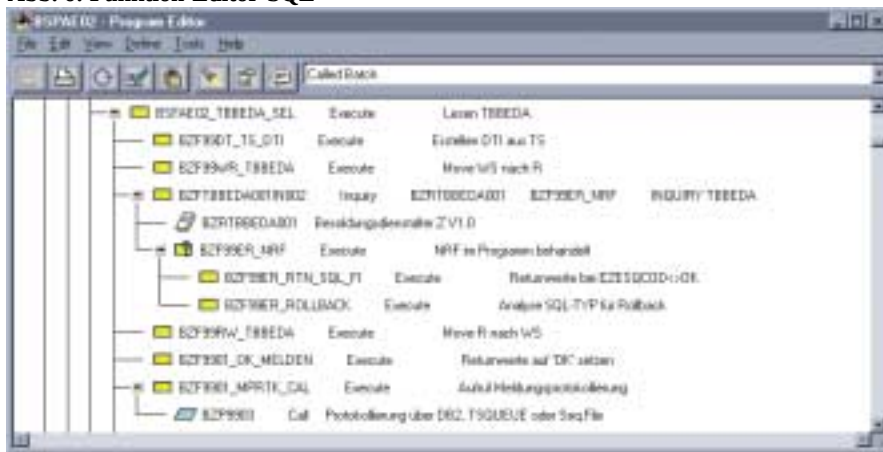


Abb. 7: Programm-Editor

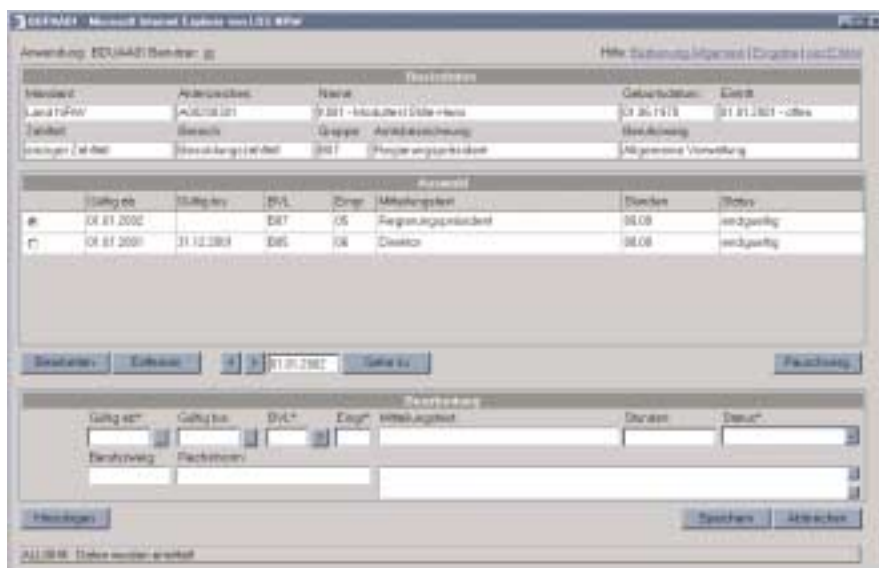


Abb. 8: Bildschirmbeispiel aus OBELIX

Testen

Warum eigentlich Testen? Natürlich nur, damit am Ende auch Qualität entsteht. Wie sieht der Testprozess des Programmerteams aus?

Grundsätzlich unterscheiden wir bei OBELIX zwei Arten des Tests:

1. der Modultest
2. der Ablauftest

Der Modultest

Im Modultest werden als erstes die vom Teilprojekt Test übergebenen Daten in die Stamm- und Referenztabellen eingepflegt. Anschließend werden mit einem selbst entwickelten Erfassungstool die Modultestdaten erstellt und in einer DB2-Tabelle abgelegt. Zuletzt wird auf Basis eines Prototyps der Testtreiber für das zu testende Modul erstellt.

Mit IBM DB2 Connect werden die DB2-Daten direkt in DB2 getestet (es entfällt das Kopieren der Daten auf die Workstation). Das vollintegrierte ITF (Interactive Test Facility) des VAGen ermöglicht das Tracen von Web-Applikationen vom Browser bis hin zur angebundene Datenbank.

Das Setzen von Break-Points und Watch-Points sowie Verändern der In-

halte von Variablen und DB2-Rekords an jeder beliebigen Stelle des Ablaufs ermöglicht ein sehr komfortables Tracen.

Den Abschluss des Modultests bildet der sog. „Cross-Check“. Das heißt, ein zweiter Programmierer überprüft die Logik und Syntax des Programms.

Der Ablauftest

Der Ablauftest untersucht das Zusammenspiel mehrerer Module eines Dialoges und der zugehörigen Batch-Prozesse. Ausgehend von der Anmeldung am OBELIX-System hat das Team die Möglichkeit, den Ablauf im ITF Schritt für Schritt zu debuggen und die Ergebnisse des jeweiligen Moduls auszuwerten.

Entscheidend für den Ablauftest sind die Übergabeschnittstellen zu anderen Modulen. Da das aufgerufene Modul mit Ergebnissen des rufenden Moduls und umgekehrt arbeitet, muss die Kommunikation der beiden Module einwandfrei sein. Besonders die modulinterne Fehlerbehandlung muss hier beachtet werden.

Da das ITF ein Interpretationssystem ist und alle Zugriffe auf ein Relationales Datenbanksystem als Cursorzugriffe realisiert werden, ist ein weiterer

Ablauftest auf einer Ausführungsumgebung im generierten Zustand notwendig, um dabei Fehler in der Commit- und Rollback-Steuerung sowie spezielle Zugriffsfehler feststellen zu können. Kommunizieren die Module einwandfrei und ist das Ergebnis auf der Ausführungsumgebung identisch, dann ist der Ablauftest abgeschlossen.

Dokumentieren

Der VAGen bietet eine Reihe von Möglichkeiten, eine ansprechende Art der Dokumentation eines Programmmoduls zu erstellen. Durch die Option des direkten Zugriffs auf das Repository und den Export des Quellcodes in eine Textdatei wird die Informationsgewinnung und deren Aufbereitung erheblich erleichtert.

Im OBELIX-Projekt enthält jedes Objekt im VAGen einen erklärenden Kommentarblock. Dieser steht entweder im Prolog oder zu Beginn des Quell-Codes und wird für die automatische Dokumentation verwendet.

Ein Java-Modul bereitet eine Anzahl von Informationen des Repository sowie die Modulstruktur und den gesamten Quellcode auf und erzeugt daraus eine XML-Datei. Dieses Modul wurde natürlich mit dem Generator entwickelt.

Das Modul nutzt die verschiedenen Fähigkeiten des Zugriffs auf interne Informationen. Eine entsprechende XSL-Datei bereitet diese für das OBELIX-Intranet auf.

Dieser Programmdokumentation sind alle Informationen über das Modul aus der Sicht eines Programmierers zu entnehmen. Fragen nach Übergabeparameter und deren Reihenfolge, Programmstruktur oder in welchem Package sich das jeweilige Modul befindet, lassen sich durch einen Blick ins Intranet schnell beantworten.

[illegible]

Abb. 9: Programmdokumentation



Abb. 10: Generierungsmaske

Die Dokumentationen der Module stehen dabei keineswegs für sich allein, denn sie sind untereinander verlinkt, so dass sich ganze Abläufe verfolgen lassen. Zwei weitere Berichte der Dokumentation sind das Ergebnis des Staurichters mit der Information über die Einhaltung der Standards und Richtlinien und der Bericht vom Test-Coverage, aus dem ersichtlich ist, zu wieviel Prozent die internen oder externen Funktionen des Programms während des Testes durchlaufen wurden. Der obligatorische Programmablaufplan und ein abschließender Programmierbericht nach der Codierung und dem Test runden die Dokumentation eines Moduls ab.

Generieren

Wie oben beschrieben wird der Programmcode in einer 4GL-Sprache geschrieben. Die Programme sollen nach der Erstellung und nach dem Modultest des Programmierers auf einer Ausführungsumgebung laufen. Dazu ist eine Umwandlung des Codes in eine herkömmliche Programmiersprache notwendig. Es kann mit dem VAGen Java, COBOL oder C++ generiert werden.

Der erste Schritt einer Generierung ist eine Umwandlung in die Programmiersprache. Anschließend wird der Code auf die Zielumgebung übertragen und dort in Maschinencode umgewandelt. Als Zielumgebungen seien hier als Beispiel Windows NT, MVS und VM genannt. Zusätzlich können mit den Programmen Web-Anwendungen erstellt werden. Der Programmierer arbeitet zu diesem Zweck mit sogenannten UI-Records. Diese werden während der Generierung in eine Java Server-Page und weitere Java Komponenten, die für die Anzeige im Web notwendig sind, umgewandelt. Im Projekt OBE-LIX ist der Prozess Generierung für einen Programmierer so aufgebaut, dass er nur die Generierungsmaske ausfüllen und starten muss. Alles andere wird durch einen automatisierten Ablauf erledigt.

Die Abbildung 10 zeigt die Generierungsmaske. Das Programm und das zugehörige PLP-Projekt sind hier noch anzugeben. Die Parameter, die man bei den Optionen angeben kann, werden schon durch das Laden eines entsprechenden VAGen Projekts mit den notwendigen Werten vorbelegt.

Fazit

Nach einem Jahr des Arbeitens mit dem VAGen liegt ein breiter Erfahrungsschatz vor. Die Versionsverwaltung, das gemeinsame Repository, die Möglichkeit, einfachen Code in verschiedene Sprachen und Umgebungen zu generieren, sind die hervorstechenden Merkmale des VAGen. Abschließend kann man sagen, dass dieses Tool das Prädikat empfehlenswert verdient hat.

Thomas Fischell

Telefon: 0211 9449-5194

E-Mail: thomas.fischell@lds.nrw.de

Jochen Gottelt

Telefon: 0211 9449-5627

E-Mail: jochen.gottelt@lds.nrw.de

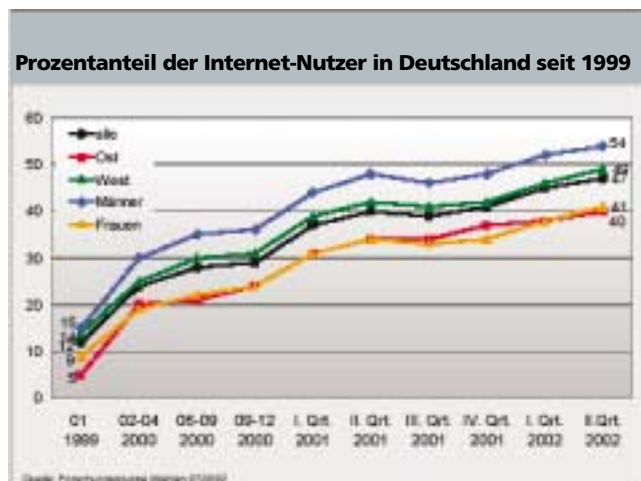
Webhosting-Dienstleistungen des LDS NRW

„I think there is a world market for maybe five computers.“
(Thomas Watson, IBM, 1943)

Im Juli 2002 bestand das Internet aus 162 Mill. Host-Rechnern (Internet Software Consortium, <http://www.isc.org/ds/host-count-history.html>).

Wachstum des Internets

Das Internet, 1969 aus der Vernetzung militärischer und universitärer Rechner entstanden, hat sich mittlerweile zu einem Massenmedium gemausert. So ist in Deutschland etwa jeder zweite Erwachsene online. Auch wenn der Höhepunkt der Interneteuphorie der 90er-Jahre bereits überschritten ist, das Internet und damit die vielfältigen Möglichkeiten, die dieses Medium bietet, breitet sich weiter aus.



„Seit Anfang 1999 hat der Anteil der deutschen Erwachsenen, die das Internet nutzen, deutlich zugenommen: von zwölf Prozent Anfang 1999 über 24 Prozent Anfang 2000 und 37 Prozent im I. Quartal 2001 auf nunmehr 47 Prozent im II. Quartal 2002. Der Internet-Boom des Jahres 2000 verlangsamte sich damit im Jahr 2001 leicht und erst mit Beginn des neuen Jahres steigen die Zahlen wieder merklich an.“ (Internet-Strukturdaten II. Quartal 2002, Forschungsgruppe Wahlen Online)

Ein weiterer Indikator macht das stete Wachstum des Internets deutlich. Die Anzahl der Domänen (System zur Namensvergabe für Internetrechner, z. B. nrw.de), die in Deutschland und weltweit registriert werden, steigt weiter an.

Jahr (31. 12.)	DE-Domains
1994	1 611
1995	5 514
1996	46 398
1997	105 344
1998	288 191
1999	1 381 384
2000	3 738 732
2001	5 147 297
2002	ca. 6 000 000

DE-Domainentwicklung 1994 – 2002 (Quelle: DENIC eG)

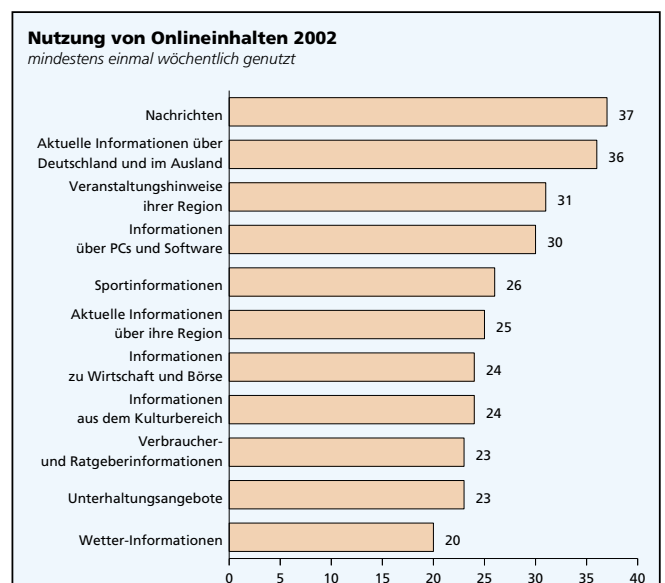
Top Level Domain	Domains per 30. 11. 2002
COM	21 892 550
DE	5 938 340
UK	3 868 421
NET	3 672 309
ORG	2 418 517

Domainzahlen im internationalen Vergleich (Quelle: DENIC eG)

Nutzung von Onlineinhalten

Viele Firmen und Institutionen präsentieren sich heute im Internet. Damit haben sich Kundeninformationen und Serviceleistungen zu einem großen Teil in das Internet verlagert.

Wie eine Online-Umfrage von ARD/ZDF zeigt, werden Webseiten mit informativen Inhalten sehr häufig genutzt.



Basis: Onlinebenutzer ab 14 Jahre in Deutschland (2002: n=1 011) – Quelle: ARD/ZDF-Online-Studie 2002

Das Internet wird auch immer stärker zu einem wichtigen Kommunikationsweg zwischen Bürgern und Verwaltung. Der Bürger möchte von seinem PC aus jederzeit Verwaltungstransaktionen – geänderten Wohnort melden, die Steuererklärung abgeben oder digitale Personaldokumente bestellen – veranlassen können.

Das Land Nordrhein-Westfalen bietet für diese Serviceleistungen bereits eine Vielzahl von Webpräsentationen, um Bürgerinnen und Bürger des Landes zu informieren, eine Kontaktaufnahme zu ermöglichen oder Prozesse zu initiieren.

Infrastruktur des LDS NRW

Das Landesamt für Datenverarbeitung und Statistik (LDS NRW) fungiert seit 1995 als Internet-Dienstleister für Einrichtungen und Behörden des Landes NRW. Da sich die Art der Darstellung im Internet von den statischen Seiten Mitte der 90er-Jahre zu komplexen Webapplikationen weiterentwickelt hat, sind die Hardware- und Softwareanforderungen für die Bereitstellung (Hosting) von Internetdiensten zum heutigen Zeitpunkt sehr hoch.

Das LDS NRW stellt eine leistungsfähige, weitestgehend redundant konzipierte Internet-Infrastruktur zur Verfügung.

Diese besteht, wie die folgende Darstellung zeigt, aus Netzwerkstruktur, Hardware und verschiedenen Softwarekomponenten.

Die Webserver im LDS NRW setzen auf verschiedenen Hardware- und Betriebssystemplattformen auf. So werden als Hardware Intel-basierte Server, Sparc-Architekturen und Mainframes, als Betriebssysteme Windows 2000, Linux und Solaris eingesetzt. Als Webserver-Software wird in der Regel Apache oder der Internet-Information-Server verwendet. So können für unterschiedlichste Kundenanforderungen, die jeweils benötigten Plattformen bereitgestellt werden.

Behörden und Einrichtungen des Landes können diese Infrastruktur verwenden, um sich im Internet zu präsentieren bzw. über das Internet Dienste und Dienstleistungen anzubieten. Die Informationsangebote der Landesverwaltung sind i. d. R. unter www.<behördenkürzel>.nrw.de im Internet zu erreichen.

Viele Ministerien und Behörden nutzen bereits die Möglichkeit, Informationen und Serviceleistungen für Bürger des Landes NRW im Internet bereitzustellen. So existieren zur Zeit ca. 600 nrw.de-Domänen und diese Zahl ist weiter ansteigend. 400 Webpräsentati-

onen des Landes NRW werden zur Zeit im LDS NRW gehostet.

Dienstleistungen des LDS NRW

Das LDS NRW bietet eine Vielzahl von Softwarekomponenten und Dienstleistungen, die notwendig sind, um ein reibungsloses Funktionieren der unterschiedlichsten Webangebote zu gewährleisten.

Es werden zur Namensvergabe für die Webangebote Subdomains von nrw.de standardmäßig eingerichtet, aber auch Registrierungen von anderen Domänen (z. B. unter den Top-Level-Domains .net, .info, .com) durch das LDS NRW sind möglich.

Wird ein FTP-Zugang benötigt, um die Internetseiten zu publizieren, d. h. auf den Webserver zu übertragen und zu pflegen, wird dieser durch das LDS NRW eingerichtet und überwacht.

Des Weiteren wird eine sichere Übertragung via SSL (Secure Socket Layer) angeboten. Diese Option wird benötigt, wenn der Besucher der Seite vertrauliche Informationen über das Internet senden soll. Mit SSL werden die Daten verschlüsselt und sind für Dritte nicht lesbar.

Speziell für Webanwendungen gibt es sehr viele Skriptsprachen, die bequemerweise in die HTML-Webseiten integriert werden können und das dynamische Erstellen von Webseiten erheblich vereinfachen können. Hier werden vom LDS NRW eine Reihe von Skriptsprachen wie zum Beispiel Perl und PHP unterstützt.

Datenbanksysteme sind konzipiert worden, um Informationen komfortabel und schnell zu speichern und Anfragen auch hoher Komplexität zu beantworten. Durch den Einsatz von web-zugänglichen Datenbanken können umfangreiche Recherchen durchgeführt und die Ergebnisse benutzer-

Zugriff auf ein Webangebot, das im LDS gehostet wird



freundlich im Browser angezeigt werden. Als Datenbanken bietet das LDS NRW hier beispielsweise MySQL oder Oracle an.

Die Protokollierungsfunktionen der Web-Server zeichnen vielerlei Informationen über das System und die Besucher einer Website in Logdateien auf. Durch Auswertung dieser Logfiles können auf Wunsch Statistiken erzeugt werden, die dem Besitzer der Seite wertvolle Informationen über das Besucherverhalten liefern. Hierbei wird im LDS NRW besonders darauf geachtet, die Vorgaben des Datenschutzes umzusetzen.

Auf dem Wahl-Server <http://www.wahlen.lids.nrw.de/> des Landesamtes für Datenverarbeitung wurde so festgestellt, dass in der Wahlnacht der Bundestagswahl am 22. 9. 2002 und am Tag danach etwa 345 000 Seitenanfragen stattgefunden haben; unter anderem auch aus dem Oman, Costa Rica und Singapur.

Gerade bei sehr umfangreichen Webangeboten kann es schwierig werden, die benötigten Inhalte zu finden. Hier bietet sich der Einsatz einer Suchmaschine an, die den Besucher zu den Themen leiten kann, die ihn interessieren. Die hierzu notwendige Software, wie beispielsweise [ht://Dig](http://Dig) wird vom LDS NRW installiert und konfiguriert. Die Suchformulare und die Ausgabeseiten können von den Betreibern der Webseite individuell gestaltet und angepasst werden.

Viele Anbieter von Webseiten sind an Kritik und Anregungen ihrer Besucher interessiert. Oder es sollen Diskussionsplattformen zu bestimmten Themen bereitgestellt werden. Hier bietet sich der Einsatz von Webforen an. Es handelt sich hierbei um eine Art schwarzes Brett, in das die Nutzer für alle sichtbar ihre Einträge vornehmen können. Die für ein solches Forum benötigte Software wird durch das LDS NRW zur Verfügung gestellt und installiert.

Regelmäßig werden seit September 2002 mittels Video-Streaming Plenarsitzungen des Landtags ins Internet übertragen (<http://www.landtag.nrw.de>). Auch die Veranstaltung eNRW am 14. November 2002 wurde innerhalb des Landesverwaltungsnetzes (LVN) live übertragen (<http://lv.lids.nrw.de/enrw/>), so dass alle Mitarbeiterinnen und Mitarbeiter im Lande NRW die Vorträge mitverfolgen konnten. Auch nachträglich kann auf die archivierten Beiträge zugegriffen werden.

Ein entsprechender Real-Video-Server steht im LDS NRW bereit und ermöglicht solche Video-Übertragungen.

In der ersten Generation von Webseiten war jede Seite statisch auf dem Webserver abgelegt, was die Möglichkeiten für Anwendungen und die Interaktionsfähigkeit stark einschränkte. Abhilfe schafft hier der Einsatz von Servlets, im Speicher liegende Java-Programme. Diese laufen in einem Servlet-Container und werden zum Bearbeiten von Anfragen verwendet. Der Vorteil hierbei ist, dass sie keine eigenen Prozesse starten müssen und daher sehr schnell sind. Als Servlet-Container bietet das LDS NRW die Möglichkeit, Apache Tomcat innerhalb eines Webangebotes zu nutzen.

Bei Projekten, die komplexere Anwendungen unter Einsatz des Mediums Internet realisieren wollen, unterstützt das LDS NRW durch Beratung und bietet eine zeitlich begrenzte Mitwirkung in entsprechenden Arbeitsgruppen an. Auch das Programmieren von Webapplikationen kann in diesem Zusammenhang als Dienstleistung des LDS NRW in Anspruch genommen werden.

Der Einsatz von Content Management Systemen (CMS) oder Redaktionssystemen ermöglicht es den Anbietern einer Internetpräsentation, die Angebote und Inhalte von Webseiten einfach zu aktualisieren, ohne umfangreiche Kenntnisse in HTML zu besitzen. Ein Kern-

prinzip von Content Management Systemen ist dabei die Trennung von Layout und Inhalt, wodurch eine dezentralisierte Pflege der Inhalte möglich wird. Als landesweites CMS wird das Produkt NPS (Network Productivity System) von Infopark eingesetzt. Das CMS-Competence-Centers (<http://ccc.lids.nrw.de/>) des Landesamtes für Datenverarbeitung und Statistik NRW bietet weitergehende Informationen zum Einsatz von NPS.

Für Präsentationen der Landesverwaltung mit besonderem Schutzbedarf wurde der so genannte Sicherheitsbereich 2 eingerichtet. Hier gelten sehr strenge Bestimmungen und es werden besondere Maßnahmen durchgeführt, um für ein Höchstmaß an Sicherheit zu sorgen. Dieser Bereich wird durch eine Firewall mit restriktiven Regeln abgeschirmt und mit Standard-Softwaretools wird sowohl die Betriebssystem-sicherheit, die Sicherheit der Anwendungen als auch die Sicherheit von Netzwerkzugriffen geprüft.

Fazit

Die Art der Präsentationen von Informationen im World Wide Web hat sich von reinen HTML-Seiten hin zu komplexen Anwendungen weiterentwickelt. Auch die dafür notwendigen Technologien haben sich diesen Anforderungen angepasst. Zu Beginn des WWW genügte ein reiner Webserver wie beispielsweise der Apache, heute werden zusätzliche Softwareprodukte wie Datenbanken und Servlet-Container benötigt. Das LDS NRW stellt mit seinen umfangreichen Dienstleistungen diese Technologien zur Verfügung.

Mit seinen heutigen Produkten hat das LDS NRW die Voraussetzungen dafür geschaffen, auch in Zukunft den Einrichtungen und Behörden des Landes NRW als kompetenter Berater in Sachen Internet zur Verfügung zu stehen.

Genauere Informationen, u. a. über die erforderlichen organisatorischen Schritte, und fachkundige Gesprächspartner und -partnerinnen zu allgemeinen Fragen oder zu individuellen Fachthemen werden gerne von der LVN-Anwenderbetreuung (Telefon: 0211 9449-2350, Fax: 0211 9449-8350, E-Mail: lvn-nutzerservice@lds.nrw.de) vermittelt.

Dipl.-Chem.'in Susanne Schmitz
Telefon: 0211 9449-2060
E-Mail: susanne.schmitz@lds.nrw.de