Erschienen in: it - Information Technology Jg. 51 (2009) H. 4, S. 197-206.

Ein XML-basiertes Datenbanksystem für digitale Wörterbücher – Ein Werkstattbericht aus dem Institut für Deutsche Sprache

An XML-Based Database System for Online Dictionaries – A Report on Lexicographic Work at the Institute for German Language

Carolin Müller-Spitzer, Roman Schneider, Institut für Deutsche Sprache, Mannheim

Zusammenfassung Das Online-Wortschatz-Informationssystem Deutsch (OWID) ist ein digitales Wörterbuchportal des Instituts für Deutsche Sprache. Alle darin zusammengeführten lexikografischen Daten sind auf XML-Basis feingranular strukturiert. Speicherung, Verwaltung und Retrieval dieser Daten übernimmt das Oracle-basierte Electronic Dictionary Administration System (EDAS). Der vorliegende Beitrag erläutert die XML-basierte Modellierung der Daten, XML-spezifische Fragen der Speicherung, sowie das Retrieval mit XPath und SQL/XML. ►►► **Summary** The Institute for German Language (IDS) hosts the lexicographic portal OWID for online dictionary access. All lexicographic data share a fine-grained XML structure. Storing, administration, and retrieval are done using the ORACLE-based Electronic Dictionary Administration System (EDAS). This article copes with questions of XML modelling for dictionary data, storing of XML fragments within.

Schlagwörter Relationale Datenbanken, Markup Sprachen, Linguistik ►►► **Keywords** H.2.4 [Information Systems: Database Management: Systems] Relational databases; H.3.1 [Information Systems: Information Storage and Retrieval: Content Analysis and Indexing]; H.3.3 [Information Systems: Information Storage and Retrieval: Information Search and Retrieval]; I.7.2 [Computing Methodologies: Document and Text Processing: Document Preparation] Markup languages; J.5 [Computer Applications: Arts and Humanities] Linguistics

1 Projekthintergrund

Gedruckte Wörterbücher wurden in der Regel genau für eine Publikation zielgerichtet erstellt und anschließend eventuell für Neuauflagen überarbeitet. Daten für digitale Wörterbücher werden dagegen zunehmend so aufbereitet, dass sie nicht nur in genau einer Form publiziert werden können, sondern dass die in ihnen integrierten Daten möglichst in mehrfacher Hinsicht zu verwenden sind: sei es für die Integration in ein Wörterbuchportal, in andere Spin-off-Produkte wie Kurzfassungen oder Handwörterbücher oder in sprachtechnologische Produkte. Daraus leitet sich die Anforderung ab, die Daten anders zu modellieren und zu strukturieren, als das bei Printwörterbüchern der Fall war. Auch Zugriffsstrukturen auf die lexikografischen Inhalte, die dem digitalen Medium angemessen sind, und/oder eine selektive Präsentation der Daten verlangen eine bestimmte Form der Modellierung. Bei der Datenhaltung lexikografischer Da-

ten spielen schon lange, d. h. seit Beginn der 1990er Jahre, Standards wie SGML oder später XML ein wichtige Rolle, sowohl in der Verlagslexikografie als auch in der wissenschaftlichen Wörterbuchlandschaft (vgl. z. B. [6; 19]). Anfangs wurden diese Standards vor allem deshalb eingesetzt, um die Langlebigkeit und Softwareunabhängigkeit der Datenhaltung zu gewährleisten (bei Wörterbuchressourcen ein sehr wichtiges Thema). Im Kontext von digitalen Wörterbüchern, insbesondere der Verbindung von Wörterbuchressourcen in einem Portal, geht es nun allerdings noch mehr darum, *wie* XML eingesetzt wird und *wie* auf die XML-strukturierten Daten zugegriffen werden kann.

Im Online-Wortschatz-Informationssystem Deutsch – kurz OWID [25] – des Instituts für Deutsche Sprache (IDS) in Mannheim spielen alle oben genannten Fragen eine wichtige Rolle. OWID ist das Portal für wissenschaftliche, korpusbasierte lexikografische Arbeiten des IDS und soll perspektivisch auch um nicht am IDS erarbeitete Wörterbücher bzw. lexikologische Datensammlungen erweitert werden [23]. Ziel von OWID ist es, die am IDS erarbeiteten lexikografischen Daten in computerlexikografisch angemessener Weise im Internet zugänglich zu machen. Momentan sind in OWID folgende digitale Wörterbücher bzw. lexikografische Produkte integriert: *elexiko* [4; 13], das *Neologismenwörterbuch* [14; 24], *Feste Wortverbindungen* [8] und das *Diskurswörterbuch 1945–55* [3; 16; 17]. Das *Handbuch deutscher Kommunikationsverben* [11; 12; 18] und ein erster Teil des *Valenzwörterbuches deutscher Verben* [5; 29] werden 2009 folgen. Die OWID-Gesamtstichwortliste, eine Kompilierung der Stichwortlisten der einzelnen Wörterbücher, besteht aus 300.000 Einheiten. Zu allen diesen Stichwörtern können Angaben zur Rechtschreibung, Silbentrennung und automatisch ausgewählte Textbeispiele aus den Korpora des IDS abgerufen werden. Zusätzlich zu diesen Angaben, die für eine große Masse an Benutzern interessant ist, bieten die einzelnen Wörterbücher in OWID unterschiedliche Informationen zum deutschen Wortschatz, die eher für Germanisten im In- und Ausland, Lerner des Deutschen und interessierte Laien gedacht sind. Alle Angebote von OWID sind kostenlos und ohne Zugangsbeschränkungen im Internet zu erreichen.

Erstes Ziel der Arbeiten ist es, die unterschiedlichen lexikografischen Daten über eine gemeinsame Suchseite miteinander zu verbinden und sie somit gemeinsam zugreifbar zu machen. In diesem Punkt entspricht OWID vielen anderen Wörterbuchportalen im Internet (z. B. [30; 31], vgl. auch [7]). Dementsprechend kann man auf der Startseite ein Wort in das Suchfeld eingeben und in allen Wörterbüchern suchen (s. Bild 1). Sucht man z. B. nach „frei*“ in allen Wörterbüchern, bekommt man ein Suchergebnis, in dem die einzelnen Stichwörter farblich unterschiedlich markiert sind und so den einzelnen Wörterbüchern zugeordnet werden können: „Freiheit“ ist sowohl ein Stichwort in elexiko (in schwarz dargestellt) wie im Diskurswörterbuch, „frank und frei“ ist rot dargestellt und gehört zu den Festen Wortverbindungen und „Freisprecheinrichtung“ (blau) gehört zum Neologismenwörterbuch.

Bild 1 Startseite OWID.

Allerdings kann man auch etwas anderes an diesem Suchergebnis erkennen, was OWID deutlich von anderen Wörterbuchportalen unterscheidet: die lexikografischen Daten sind feingranular und inhaltsorientiert ausgezeichnet, sodass zum Beispiel weiter unten in der „frei*"-Suchergebnisliste Angaben zu sehen wie:

- *frei* Sublemma zu **Freiheit**
- *frei* Basiselement zu **frank und frei**
- *Freisprech-Anlage* nichtnormgerechte Schreibvariante zu **Freisprechanlage**
- *Freisprechanlagen* Nominativ Plural von **Freisprechanlage**
- *Freisprech-Einrichtung* nichtnormgerechte Schreibvariante zu **Freisprecheinrichtung**
- *Freisprecheinrichtungen* Nominativ Plural von **Freisprecheinrichtung**

Dass alle Angaben der einzelnen Wörterbücher so ausgezeichnet sind, dass ihre Inhalte maschinenlesbar zugeordnet und ausgewertet werden können, ist auch die Voraussetzung für erweiterte Suchfunktionen, wie sie in elexiko oder im Neologismenwörterbuch angeboten werden (vgl. auch [22]). Zum Beispiel kann recherchiert werden, welche „Nomen" innerhalb der Neologismen „Neulexeme" sind und in der „Mitte der 90er Jahre" in die Deutsche Sprache eingegangen sind. Als Suchergebnisse erhält der Benutzer in diesem Fall Stichwörter wie „Dosenpfand", „DVD" oder „Palliativmedizin". Eine solche detaillierte Suchanfrage ermöglicht die Abfrage interessanter Stichwortmengen, die – im Falle der Neologismen – viele Rückschlüsse auf neue Produkte oder Entwicklungen aus dieser Zeit ziehen lassen. Oder man kann – eher sprachwissenschaftlich orientiert – nach allen „Kurzwörtern" suchen, die „Neulexeme" sind. Im letzten Fall findet man Stichwörter wie „der/die Ex", die „Site" oder „Rinderwahn". Gerade diese erweiterten Suchmöglichkeiten sollen noch weiter ausgebaut werden, um Experten neue Zugriffsstrukturen auf die OWID-Inhalte zu eröffnen.

Die feingranulare Datenauszeichnung ist besonders für elexiko auch deshalb wichtig, weil die Inhalte perspektivisch benutzeradaptiv unterschiedlich dargestellt werden sollen. D. h. es soll möglich sein, vor einer Suche im Wörterbuch anzugeben, ob man beispielsweise Muttersprachler ist oder nicht, ob man einen Text liest, einen Text schreibt usw., und je nach Profil unterschiedliche Informationen aus den Wörterbuchartikeln zu erhalten. Dies gilt sowohl für die Terminologie der Benutzeroberfläche (u. a. bei den generierten Überschriften), als auch für die Angabearten, die je nach Benutzungssituation unterschiedlich dargestellt werden sollen. Auch aus diesem Grund sind alle lexikografischen Angaben einzeln ausgezeichnet. Zur Illustration ist in Bild 2 ein Auszug aus der XML-Instanz des elexiko-Artikels „abverlangen" zu sehen, der die Angaben zur Valenz[1] zeigt. Online werden im

```
<vb-valenz-neu>

<satzbauplan>
<satzbauplanA>JEMAND / ETWAS verlangt
(JEMANDEM / ETWAS) ETWAS ab</satzbauplanA>
</satzbauplan>

<satzbauplan>
<satzbauplanA>JEMAND / ETWAS verlangt ETWAS
(VON JEMANDEM / ETWAS) ab</satzbauplanA>

<angabe-zusatz><kommentar>
<k-absatz>Das Akkusativkomplement ist
obligatorisch. Daneben können alternativ ein
Dativkomplement oder eine
Präpositionalphrase als Komplement
auftreten. Die Präpositionalphrase ist im
elexiko-Korpus seltener belegt als das
Dativkomplement (vgl. die
Belege).<belege><beleg>...</beleg></belege><
/k-absatz>
</kommentar></angabe-zusatz>

</satzbauplan>

<vb-komplemente-neu>
<subjekt-komp-neu obligatorisch="ja"><nom-
nominalphrase-neu/></subjekt-komp-neu>

<objekt-komp-vb obligatorisch="ja"><akk-
nominalphrase-vb/><dass-satz-vb/></objekt-
komp-vb>

<objekt-komp-vb obligatorisch="nein"><dat-
nominalphrase-vb/><dat-reflexivum-
vb/></objekt-komp-vb>

<objekt-komp-vb
obligatorisch="nein"><praepositionalphrase-
vb praeposition="von"/></objekt-komp-vb>
</vb-komplemente-neu>

</vb-valenz-neu>
```

Bild 2 Auszug aus der XML-Instanz des elexiko-Artikels „abverlangen" (Angaben zur Valenz).

Bereich der Valenz in elexiko nur die Satzbaupläne und der dazugehörige Kommentar dargestellt (hier unterstrichen), alle anderen Angaben, wie z. B. die der möglichen Komplemente, Angaben zu ihrer Obligatorik und den Realisierungen, dienen nur der Recherche.

Die Wörterbücher in OWID sind inhaltlich unabhängig voneinander. Allerdings gibt es in bestimmten Angabebereichen Überschneidungspunkte. Beispielsweise konnten sich die Mitarbeiter des elexiko- und des Neologismenprojekts auf eine gleiche Struktur der grammatischen Angaben oder der Angaben zur Wortbildung einigen. Ein anderes Beispiel ist die gleiche Modellierung allgemeiner Elemente, die in allen OWID-Wörterbüchern genutzt werden, wie Belege, Kommentare, Absatzmodelle usw. Daneben gibt es aber Strukturen, die für die

[1] „Unter Valenz wird – in metaphorischer Anlehnung an die chemische Valenzbindung – die Eigenschaft sprachlicher Ausdrücke (als Valenzträger) verstanden, Leerstellen zu eröffnen, die durch andere sprachliche Ausdrücke bestimmter Art (Komplemente) gefüllt werden können. Valenzträger können Verben, Adjektive und Nomen sein." (GRAMMIS Terminologisches Wörterbuch, http://hypermedia.ids-mannheim.de/pls/public/termwb.ansicht?v_id=133) [10].

unterschiedlichen Wörterbücher verschieden modelliert werden müssen, da allen eine eigene wissenschaftliche Konzeption zugrunde liegt. Beispielsweise wird im elexiko-Wörterbuch bei vielen Angaben vermerkt, ob sie aus dem elexiko-Korpus stammen oder nicht, was bei anderen Wörterbüchern nicht vermerkt wird. Um auf Ebene der Modellierung diese Situation abzubilden und sicherzustellen, dass gleiche Strukturen in verschiedenen Wörterbüchern auch identisch modelliert werden können, wurden die einzelnen Artikelstrukturen in einer XML-DTD-Bibliothek[2] zusammengeführt. Diese DTD-Bibliothek enthält etwa 650 Elemente und dazugehörige Attribute. Durch die Aufteilung der Bibliothek in mehrere Ebenen (ganz OWID/identisch für bestimmte Wörterbücher/speziell ein Wörterbuch) ist schnell zu erkennen, welche der Strukturbausteine portalübergreifend bzw. für mehrere Wörterbücher gleich sind. Auch Änderungen an allgemeinen Elementen sind so in konsistenter Weise auszuführen. Das heißt aus dieser DTD-Bibliothek können auch die Zugriffsstrukturen, über die auf mehr als einem Wörterbuch gesucht werden kann, abgeleitet werden. Die Vorteile im Umgang mit solchen verteilten DTDs in Hinsicht auf Änderungen und Konsistenz überwiegen die Nachteile in der Anwendung.

Alle Wörterbuchdaten – Artikelinhalte ebenso wie Metadaten wie DTDs, XML-Schemata sowie XSLT-Stylesheets – werden im *Electronic Dictionary Administration System*, kurz EDAS, verwaltet. Die technischen Grundlagen und Hintergründe von EDAS werden im Folgenden erläutert.

2 XML-Verarbeitung in einem Datenbankmanagementsystem

Eine gleichermaßen effiziente und dauerhafte Verwaltung großer Mengen heterogener XML-Instanzen in der Unicode-Kodierung UTF-8 – das Datenvolumen der über OWID abfragbaren Wörterbücher beträgt derzeit ca. 2 GB, in absehbarer Zeit ca. 5 GB – stellt bestimmte Anforderungen an Architektur und Konfiguration der eingesetzten Softwarelösung. Vorrangige Ziele sind dabei gemeinhin Datenintegrität, -konsistenz und -sicherheit, das Handling konkurrierender Zugriffe in kollaborativen Arbeitsumgebungen, permanente Verfügbarkeit sowie die Bereitstellung von Verfahren für Backup und Recovery. Um diese Ziele zu erreichen, bietet sich der Einsatz eines Datenbankmanagementsystems (DBMS) an, das nicht nur die persistente Speicherung der Wörterbuchdaten erlaubt, sondern auch integrierte Werkzeuge und Schnittstellen für die Weiterverarbeitung und Analyse der Inhalte zur Verfügung stellt.

Im Bereich der Datenbankunterstützung für semistrukturierte Daten lässt sich seit einigen Jahren eine rasante Entwicklung beobachten [1], die auf absehbare Zeit noch nicht abgeschlossen sein dürfte und sich neben funktionalen und marktabhängigen Aspekten nicht zuletzt an den weiteren Empfehlungen des World Wide Web Consortiums (W3C) für XML-Technologien orientieren wird. Die kommerziellen Datenbankhersteller IBM, Microsoft und Oracle haben ihre etablierten relationalen bzw. objektrelationalen Produkte bereits um eine Vielzahl von Erweiterungen für die XML-Verarbeitung ergänzt. Hierzu zählen in erster Linie native XML-Datentypen für die effiziente Speicherung von XML-Inhalten, interne XML-Parser, die Unterstützung von Abfragetechniken wie XPath und XQuery sowie XSLT-Transformationen [15]. Dabei lässt sich eine weitgehende Konformität zu einschlägigen Standards – speziell ISO SQL-2003 (SQL/XML) – beobachten. Alternativ kommen in diesem Zusammenhang auch native XML-Datenbanksysteme wie Tamino, eXist oder Apache Xindice in Betracht, die aber erfahrungsgemäß für eng terminierte und datenintensive Projekte aus anderen Gründen (Neuanschaffungskosten, Einarbeitungszeit, Skalierbarkeit, Integration von Nicht-XML-Daten) nicht erste Wahl sind. Prominente relationale Open Source-Produkte wie MySQL, PostgreSQL oder Firebird scheinen bei anspruchsvollem XML-Processing hinsichtlich des Leistungsumfangs derzeit noch nicht mithalten zu können; insbesondere fehlen ausgereifte integrierte Lösungen für XML-Techniken wie XPointer, XQuery, XSLT usw. Eine verbindliche vergleichende Bewertung zur Performanz der verfügbaren Systeme fällt schwer – und soll im Rahmen dieses Beitrags auch nicht geleistet werden –, weil projektspezifische Anforderungen (Schwerpunkt auf Retrieval-Geschwindigkeit vs. Update-Performanz, Einbenutzer- vs. Mehrbenutzer-Umgebung etc.) stets variieren und die Non-Disclosure-Passagen in den Lizenzvereinbarungen kommerzieller Datenbankhersteller eine Verbreitung von Leistungsmessungen einschränken. In diesem Zusammenhang sei jedoch erwähnt, dass im einschlägigen Forschungsumfeld einige vielversprechende Benchmark-Initiativen, z. B. XMach-1 [2] oder die INEX-Initiative [9] gestartet wurden.

Vor diesem Hintergrund erfolgte die Implementierung von EDAS (*Electronic Dictionary Administration System*), dem IDS-Datenbanksystem für XML-basierte Wörterbücher, unter Verwendung des bereits seit vielen Jahren institutsweit eingesetzten Datenbankmanagementsystems Oracle. Oracle hat im aktuellen Release 11 g ([20]) seines objektrelationalen DBMS die bereits in Version 9i eingeführte XML-Unterstützung ([21]) konsequent ausgebaut, bietet integrierte XPath/XQuery-Funktionen ebenso wie projektrelevante Features wie XML Update oder Schema Evolution, und hat insbesondere die Performanz seines XML-Datentyps XMLType durch Vorstellung einer zusätzlichen binären Speicheroption erweitert [27].

XMLType dient der direkten Speicherung von XML-Instanzen bzw. Fragmenten und kann gleichermaßen für das Anlegen spezieller XMLType-Tables, bei der Definition relationaler Tabellenspalten sowie in Stored Procedures (PL/SQL oder Java) benutzt werden. Die

[2] Beim Import in die Datenbank werden die DTDs automatisch in XML-Schemata umgewandelt.

detaillierten Speicheroptionen bleiben für den Anwender transparent, d. h. sie beeinflussen nicht die für den Datentyp erlaubten Operationen. Persistente XMLTypes erlauben die Wahl zwischen drei Optionen: dokumentenzentriert als CLOB (Character Large Object), objekt-relational sowie binär. Die erste Variante – die das DBMS automatisch verwendet, sofern der Entwickler keine anderweitigen Vorgaben macht – gewährleistet „document fidelity", d. h. XML-Inhalte werden zeichengenau in einem Stück und unter Beibehaltung sämtlicher Metainformationen, Whitespaces usw. gespeichert. Dieses Vorgehen beschleunigt die Ein- und Ausgabe kompletter Instanzen. Die zweite XMLType-Variante wird durch den Zusatz „store as object relational" aktiviert und erzeugt intern unter der Verwendung des zur jeweiligen XML-Instanz gehörenden Schemas passende Objekttypen und objekt-relationale Tabellen. Bei der Speicherung verteilt das System die einzelnen Bestandteile einer XML-Instanz (Elemente, Attribute, Text) automatisch in diese generierten Strukturen, so dass Knotenstruktur sowie die Beziehungen zwischen den Knoten erhalten bleiben (DOM fidelity). Spätere DML-Operation, z. B. das Bearbeiten einzelner Knoteninhalte (piecewise update), werden dadurch ebenso wie die selektive Abfrage deutlich performanter. Der Nachteil dieser Speicheroption liegt in der eingeschränkten Flexibilität, denn jede Änderung am XML-Schema zieht entsprechende Modifikationen an der objektrelationalen Struktur nach sich. Binary XML, die dritte Speichervariante für XMLType, legt XML-Inhalte komplett in einem kompakten binären Format ab und garantiert DOM fidelity. Die Angabe eines zugehörigen XML-Schemas ist optional, d. h. das System prüft während der Eingabe in diesem Fall lediglich auf Wohlgeformtheit des jeweiligen XML-Fragments. Infolgedessen eignet sich der binäre XML-Datentyp besonders für Anwendungen, die mit heterogenen bzw. variablen XML-Strukturen sowie hohen Textanteilen umgehen müssen, und wurde für die Implementierung von EDAS gewählt.

Vor jeder physikalischen Umsetzung steht idealerweise eine logische Modellierung. Im vorliegenden Fall musste dabei gleichermaßen die Mikrostruktur, d. h. die interne Gliederung der Wörterbuchartikel, wie auch die Makrostruktur, d. h. Inhalte und Vernetzung der wörterbuchübergreifenden Anwendung, beachtet werden. Da es für einzelne Wörterbuchprojekte dabei unterschiedliche Vorgaben gibt (Werden Autorenname und Bearbeitungsdatum im Fließtext gekennzeichnet? An welcher Stelle im Artikel steht die Grundform (Lemma)? Wie sollen verschiedene Lesarten eines Eintrags modelliert werden? Wie änderungsfreundlich soll ein Artikel aufgebaut sein? usw.), ist es erfahrungsgemäß heikel, einen verbindlichen Trennstrich zwischen diesen beiden Ebenen zu ziehen. Die Lösung bestand darin, sämtliche für den Workflow relevanten Informationen auf der Makroebene zu modellieren und den beteiligten Wörterbuchprojekten dadurch maximale Freiheiten für die interne Artikelorganisation zu belassen. Bei der Implementierung gilt für Wörterbuchdatenbanken wie EDAS – und darüber hinaus für sämtliche Projekte, bei denen zwischen textorientierten Primärdaten und systeminternen Metadaten unterschieden werden kann – die Prämisse: Primärinhalte lassen sich flexibel und komfortabel in XML kodieren, Metadaten für zeitkritische Operationen sollten eher auf der (relationalen) DBMS-Ebene angesiedelt werden; doppelt kodierte Inhalte (z. B. Autorenname sowohl im Artikeltext als auch in den Workflow-Tabellen) können ggf. per Trigger abgeglichen werden.

XMLType-Inhalte lassen sich durch eine Vielzahl spezieller Funktionen abfragen, auswerten und ändern. Für die Suche nach bestimmten XML-Knoten und -Inhalten stehen die SQL/XML-Befehle *existsNode()*, *extract()* und *extractValue()* zur Verfügung. Für die Adressierung der Knoten kommen XPath-Ausdrücke zum Einsatz. Beispielsweise sucht das Statement *SELECT co_id FROM tb_lexikon WHERE extractValue(co_artikel, '//ortho-variante/@typ') = 'norm-variante';* nach allen Lexikoneinträgen mit einem Knotenelement namens „ortho-variante", dessen „typ"-Attribut den Wert „norm-variante" hat – d. h. nach allen Wörterbucheinträgen mit normgerechten orthografischen Varianten. Das Ändern von XMLType-Inhalten – beispielsweise das nachträgliche Einfügen des Autorennamens in die Instanz – geschieht über den SQL/XML-Befehl *updateXML()*; Funktionen wie *XMLElement*, *XMLConcat*, *XMLForest* und *XMLAgg* helfen beim Sammeln, Ordnen und Formatieren [28].

Die Abfragekosten eines SQL-Statements mit integriertem XPath-Ausdruck – sprich: die für die Ausführung der Abfrage benötigte Zeit – lassen sich durch den Einsatz spezialisierter Indizes und integrierter kostenbasierter Optimierer signifikant reduzieren. Für häufig wiederkehrende XPath-Ausdrücke bieten sich funktionsbasierte Indizes (function based indexes) an, die analog zu B-Tree-Indizes organisiert sind und z. B. durch *CREATE INDEX idx_orthotyp ON tb_lexikon (extractValue(co_artikel, '//ortho-variante/@typ'));* angelegt werden. Kann das Spektrum der in der Zukunft benötigten Abfragen nicht von vornherein eingegrenzt werden, was etwa bei der freien Recherche in komplexen Wörterbuchinhalten der Fall ist, kommt der auf XML-Inhalte spezialisierte Indextyp *XMLIndex* ins Spiel. Dieser arbeitet intern mit Schattentabellen und besteht aus drei Bestandteilen: Einem Pfadindex für die Indizierung von XPath-Adressen innerhalb einer Instanz, einem Positionsindex für die Verwaltung von *child/ancestor/sibling*-Beziehungen zwischen den Knoten sowie einem Inhaltsindex für die Textinhalte einzelner Knoten. Weiterhin werden ROWID-Informationen sowie – bei Verfügbarkeit eines passenden Schemas – Angaben zum Datentyp des Knotens verwaltet. Der Index aktualisiert sich standardmäßig bei jeder DML-Operation (parallel maintenance); optional lässt sich dieser Vorgang auch in betriebsarme Zeiten verlegen (asynchronous maintenance). Neben exakten

Abfragen wird die Suche nach Wertebereichen (range queries) oder Wildcard-Ausdrücken unterstützt.

Für die Kombination von Volltextabfragen und Anfragen zur Dokumentenstruktur (Markup) hat sich darüber hinaus der *Oracle Text Index* (*CONTEXT*) als leistungsstarke Alternative erwiesen (siehe [20] und [26]). CONTEXT-Indizes erlauben feinkörnige sprachspezifische Einstellungen zur Worttrennung, Groß-/Kleinschreibung oder fehlertoleranten Suche sowie Stoppwortlisten und semantische Suchoptionen. Die Indexerstellung verläuft in mehreren Schritten; zunächst kann eine Stoppwortliste erstellt werden, um hochfrequente Tokens zu ignorieren:

```
begin

 ctx_ddl.create_stoplist('EDAS_STOPLIST');

 ctx_ddl.add_stopword('EDAS_STOPLIST','der');

 ctx_ddl.add_stopword('EDAS_STOPLIST','die');

 ctx_ddl.add_stopword('EDAS_STOPLIST','das');

end;
```

Die Lexer-Präferenz legt fest, wie mit Komposita oder Worttrennern umgegangen werden soll. Außerdem lässt sich u. a. einstellen, ob zwischen Groß- und Kleinschreibung unterschieden oder ein Thesaurus für eine thematische Suche integriert wird:

```
begin

 ctx_ddl.create_preference('EDAS_LEXER',
 'BASIC_LEXER');

 ctx_ddl.set_attribute('EDAS_LEXER','COMPOSITE',
 'GERMAN');

 ctx_ddl.set_attribute('EDAS_LEXER',
'PRINTJOINS','-');

 ctx_ddl.set_attribute('EDAS_LEXER',
 'MIXED_CASE','YES');

 ctx_ddl.set_attribute('EDAS_LEXER',
 'INDEX_THEMES','NO');

end;
```

Mit Hilfe der Wordlist-Präferenz kann automatisches Stemming oder Fuzzy Matching ein- und ausgeschaltet werden. Relevant für die Performanz von Wildcard-Suchen sind die Einstellungen zum Substring- und Prefix-Indexing:

```
begin

 ctx_ddl.create_preference('EDAS_WORDLIST',
 'BASIC_WORDLIST');

 ctx_ddl.set_attribute('EDAS_WORDLIST',
 'STEMMER','GERMAN');

 ctx_ddl.set_attribute('EDAS_WORDLIST',
 'FUZZY_MATCH','GENERIC');

 ctx_ddl.set_attribute('EDAS_WORDLIST',
 'FUZZY_NUMRESULTS','0');

 ctx_ddl.set_attribute('EDAS_WORDLIST',
 'SUBSTRING_INDEX','TRUE');

 ctx_ddl.set_attribute('EDAS_WORDLIST',
 'PREFIX_INDEX','TRUE');

 ctx_ddl.set_attribute('EDAS_WORDLIST',
 'PREFIX_MIN_LENGTH','2');

 ctx_ddl.set_attribute('EDAS_WORDLIST',
 'PREFIX_MAX_LENGTH','10');

 ctx_ddl.set_attribute('EDAS_WORDLIST',
 'WILDCARD_MAXTERMS','10000');

end;
```

Weitere Parameter betreffen den Speicherort der zu indizierenden Dokumente (z. B. eine XMLType-Spalte, ein Nested Table oder eine externe Datei), die Formatierung der Dokumente (XML, PDF, Word usw.) sowie die Pfadsuche mit INPATH- und HASPATH-Operatoren. Eine detaillierte Beschreibung aller Optionen muss hier aus Platzgründen entfallen; abschließend folgt ein Beispielkommando für den Indexaufbau:

```
create index LEXIKON_INDEX on TB_LEXIKON
(CO_ARTIKEL) indextype is CONTEXT parameters('

 datastore      DEFAULT_DATASTORE

 filter         NULL_FILTER

 section group  PATH_SECTION_GROUP

 lexer          EDAS_LEXER

 wordlist       EDAS_WORDLIST

 stoplist       EDAS_STOPLIST');
```

3 Retrieval mit SQL und XPath

Wie anfangs gesagt wurde, sind die lexikografischen Daten des IDS auch deshalb so feingranular modelliert, damit die einzelnen Angaben gezielt zugreifbar sind. Dies ist sowohl für die Benutzer als auch für die Lexikografen von zentraler Bedeutung. Für die Benutzer von außen soll es möglich sein, auf übersichtliche Weise komplexe Suchanfragen stellen zu können und in schneller Zeit das passende Suchergebnis zu erhalten. Für Lexikografen ist es darüber hinaus wichtig, flexibel eigene Suchanfragen bzw. Kombinationen von verschiedenen Anfragen formulieren zu können.

Dabei ist es eine typische und interessante Suchanfrage, beispielsweise nach allen Stichwörtern zu suchen, die in der Bedeutungserläuterung eine bestimmte Zeichenkette enthalten, z. B. „Computer*“. Dies ist nun in EDAS in unkomplizierter Weise realisierbar, das Sys-

tem übersetzt den eingetippten XML-Knotenbezeichner *//paraphraseA* mit dem Inhalt *Computer** automatisch in die SQL-Abfrage

SELECT co_id FROM tb_lexikon t WHERE contains (t.co_artikel,'Computer% inPath(//paraphraseA)') > 0.

Dieses Beispiel nutzt vermittels des CONTAINS-Operators explizit den bereits erwähnten CONTEXT-Indextyp, einen für komplexe Text Queries und XPath-Abfragen gleichermaßen performanten Ansatz. CONTEXT-Indizes lassen sich auf Tabellenspalten für Markup-Dokumente wie auch auf proprietären Formaten (DOC, PDF usw.) erstellen. Der CONTAINS-Operator liefert für eine gegebene Text Query (hier: *'Computer% inPath(//paraphraseA)'*) einen Relevance Score (*> 0*) für jede passende XML-Instanz; der Recall lässt sich optional u. a. via Query Rewrite bzw. Query Relaxation anpassen.

In diesem Fall erhält man Stichwörter wie *Barcode*, *Infohighway* oder *Trojaner*, die alle zum computerbezogenen Wortschatz gehören. Darüber hinaus ist es möglich, sehr spezielle Suchanfragen zu stellen, die zum Teil für interne Konsistenzprüfungen wichtig sind, aber auch für Expertennutzer interessant wären. Bezogen aus dem vorne gezeigten Angabebereich Valenz kann man beispielsweise den Datenbestand nach starken Verben durchsuchen (//vollverb), die ein obligatorisches Objekt-Komplement haben (//objekt-komp-vb/@obligatorisch= „ja"), welches als Nominalphrase im Dativ realisiert ist (//dat-nominalphrase-vb). Als Ergebnis erscheinen Verben wie „abverlangen", „abziehen" oder auch „emailen". Die Notation von XPath ist dabei einfach genug, um auch von technisch nicht besonders versierten Mitarbeitern gut erlernt werden zu können – die Übersetzung der Abfrage in den komplexen SQL/XML-Code

SELECT co_id FROM tb_lexikon t WHERE contains (t.co_artikel,'hasPath(//vollverb) and ja inPath(//objekt-komp-vb/@obligatorisch) and hasPath(//dat-nominalphrase-vb)') > 0

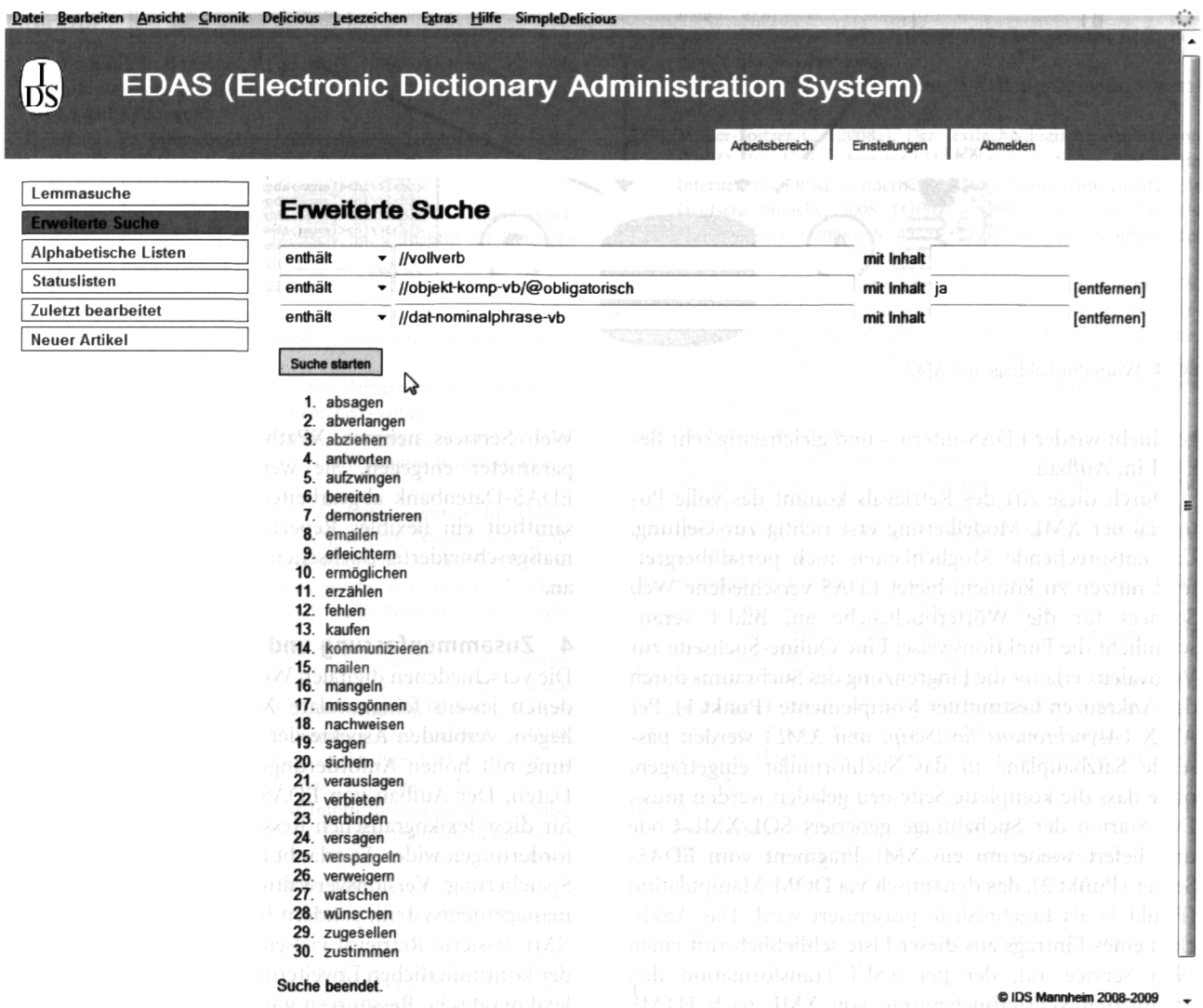

Bild 3 XPath-Recherche in EDAS.

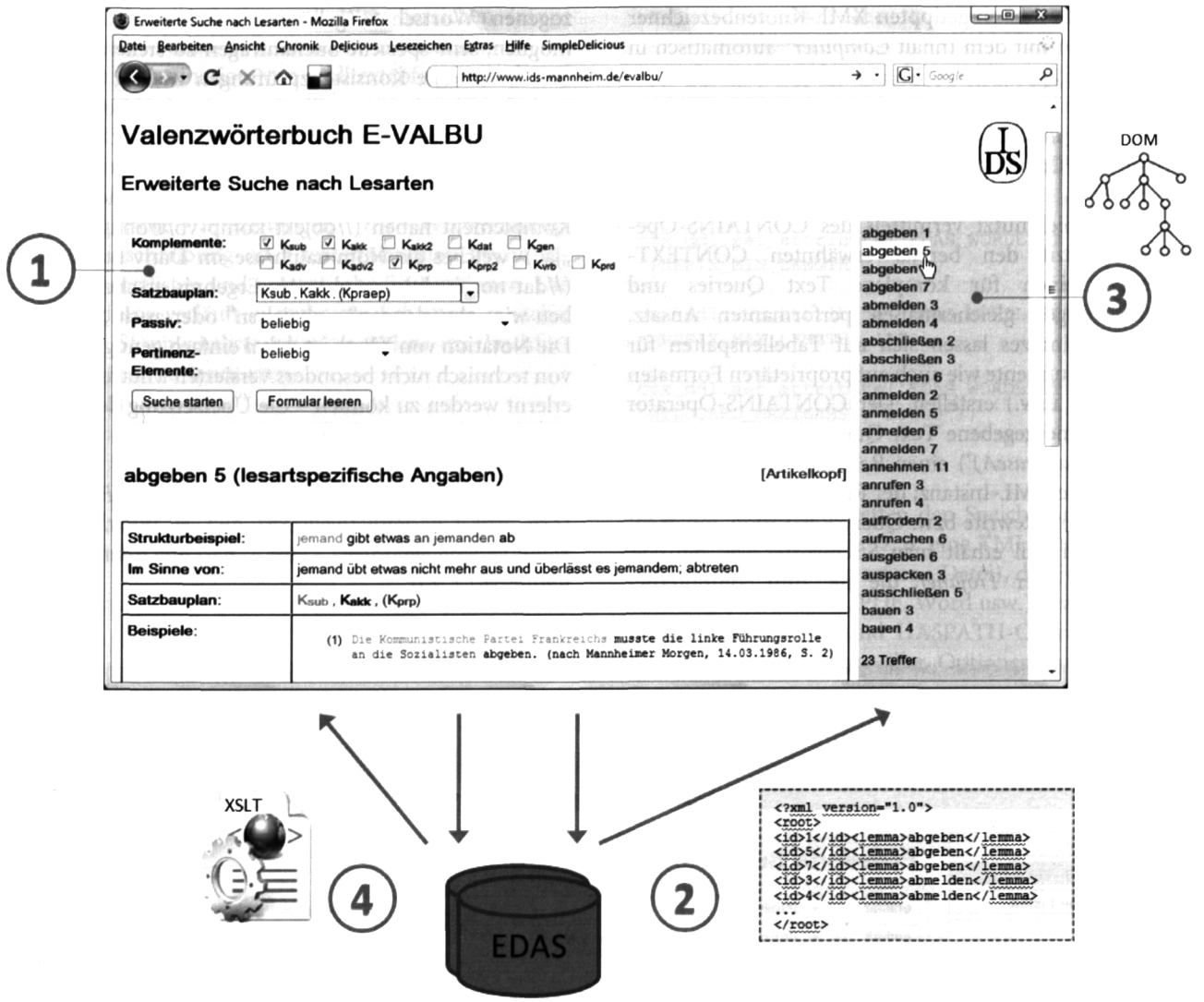

Bild 4 Wörterbuchabfrage mit AJAX.

geschieht wieder EDAS-intern – und gleichzeitig sehr flexibel im Aufbau.

Durch diese Art des Retrievals kommt das volle Potenzial der XML-Modellierung erst richtig zur Geltung. Um entsprechende Möglichkeiten auch portalübergreifend nutzen zu können, bietet EDAS verschiedene Web Services für die Wörterbuchsuche an. Bild 4 veranschaulicht die Funktionsweise: Eine Online-Suchseite zur Verbvalenz erlaubt die Eingrenzung des Suchraums durch das Ankreuzen bestimmter Komplemente (Punkt 1). Per AJAX (*Asynchronous JavaScript and XML*) werden passende Satzbaupläne in das Suchformular eingetragen, ohne dass die komplette Seite neu geladen werden muss. Das Starten der Suchabfrage generiert SQL/XML-Code und liefert wiederum ein XML-Fragment vom EDAS-Server (Punkt 2), das dynamisch via DOM-Manipulation (Punkt 3) als Ergebnisliste präsentiert wird. Das Anklicken eines Eintrags aus dieser Liste schließlich ruft einen Web Service auf, der per XSLT-Transformation den kompletten Wörterbucheintrag von XML nach HTML übersetzt und in die Seite integriert (Punkt 4). Sämtliche Web Services nehmen XPath-Notationen als Anfrageparameter entgegen. Sie werden unmittelbar in der EDAS-Datenbank abgearbeitet und bieten in ihrer Gesamtheit ein flexibles Repertoire für die Konstruktion maßgeschneiderter Suchseiten für digitale Wörterbücher an.

4 Zusammenfassung und Ausblick

Die verschiedenen digitalen Wörterbuchprojekte des IDS, denen jeweils feingranulare XML-Strukturen zugrunde liegen, verbinden Aspekte der XML-basierten Datenhaltung mit hohen Anforderungen an das Retrieval dieser Daten. Der Aufbau von EDAS, dem Verwaltungssystem für diese lexikografischen Ressourcen, spiegelt diese Anforderungen wider. Es erlaubt nicht nur eine komfortable Speicherung, Versionsverwaltung usw. wie jedes Datenmanagementsystem, sondern bietet insbesondere für das XML-basierte Retrieval vielseitige Möglichkeiten. Neben der kontinuierlichen Erweiterung des Systems durch neue lexikografische Ressourcen gilt es in der weiteren Arbeit, EDAS noch weiter den spezifischen Bedürfnissen der le-

xikografischen Nutzer anzupassen. Dies betrifft z. B. die Präsentation von Suchergebnissen. Grundsätzlich wäre es für manche Arbeiten sinnvoll, neben einem ganzen Artikel auch einzelne Knoten aus der XML-Struktur als Ergebnis ausgeben zu können oder auch Suchergebnismengen schneiden zu können. Genauso gilt es, die Verwaltung der Vernetzungsstrukturen noch komfortabler zu gestalten.

Daneben wird es in Zukunft für uns darum gehen, die Möglichkeiten des Retrievals in neuartige Zugriffsstrukturen für die OWID-Nutzer umzusetzen. Neben der technischen Grundlage sind dafür jedoch auch Forschungen wichtig, mit welchen Formen des Zugriffs die Nutzer gut umgehen können und mit welchen nicht. So ist z. B. zu vermuten, dass ein Suchfeld, in dem reine XPath-Ausdrücke eingegeben werden müssen, bei wachsender Komplexität mehr und mehr Nutzer überfordert. Deshalb stellt sich die Frage, wie Suchmöglichkeiten zum einen möglichst flexibel, aber auch noch intuitiv nutzbar gestaltet werden sollen.

Literatur

[1] Abiteboul, S.; Buneman, P.; Suciu, D. (1999). Data on the Web: From Relations to Semistructured Data and XML. San Francisco: Morgan Kaufmann.

[2] Böhme, T.; Rahm, E. (2001). XMach-1: A Benchmark for XML Data Management. In: Proc. of German Database Conf. BTW2001. Berlin:Springer. S. 264–273.

[3] Diskurswörterbuch 1945–55 (2007). In: OWID – Online Wortschatz-Informationssystem Deutsch, hg. v. Institut für Deutsche Sprache, Mannheim, www.owid.de/Diskurs1945-55/index.html.

[4] elexiko (2003ff.). In: OWID – Online Wortschatz-Informationssystem Deutsch, hg. v. Institut für Deutsche Sprache, Mannheim, www.owid.de/elexiko_/index.html (zuletzt besucht am 01.04.2009).

[5] E-VALBU (2009ff.). Elektronisches Valenzwörterbuch deutscher Verben, http://www.ids-mannheim.de/e-valbu/.

[6] Engelberg, S.; Lemnitzer, L. (2001). Lexikographie und Wörterbuchbenutzung (= Stauffenburg Einführungen, Band 14). Tübingen: Narr.

[7] Engelberg, S.; Müller-Spitzer, C. Dictionary portals. In: Dictionaries. An international encyclopedia of lexicography. Supplementary volume: Recent developments with special focus on computational lexicography, ed. by R. H. Gouws; U. Heid; W. Schweickhard; H. E. Wiegand. Berlin/New York: de Gruyter (forthcoming).

[8] Feste Wortverbindungen (2007ff.). In: OWID – Online Wortschatz-Informationssystem Deutsch, hg. v. Institut für Deutsche Sprache, Mannheim, www.owid.de/Wortverbindungen/index.html.

[9] Fuhr, N.; Lalmas, M. (2007). Advances in XML retrieval: the INEX initiative. In: IWRIDL '06: Proc. of the 2006 Int'l Workshop on Research Issues in Digital Libraries. New York: ACM. S. 1–6.

[10] GRAMMIS – das grammatische Informationssystem des Instituts für Deutsche Sprache, http://www.ids-mannheim.de/grammis/.

[11] Harras, G.; Winkler, E.; Erb, S.; Proost, K. (2004). Handbuch deutscher Kommunikationsverben. Teil 1: Wörterbuch (= Schriften des Instituts für Deutsche Sprache 10.1). Berlin/New York: de Gruyter.

[12] Harras, G.; Proost, K.; Winkler, E. (2007). Handbuch deutscher Kommunikationsverben. Teil 2: Lexikalische Strukturen (= Schriften des Instituts für Deutsche Sprache 10.2). Berlin/New York: de Gruyter.

[13] Haß, U. (ed.) (2005). Grundfragen der elektronischen Lexikographie. elexiko – das Online-Informationssystem zum deutschen Wortschatz. (Schriften des Instituts für Deutsche Sprache), Berlin/New York: de Gruyter. Kommunikationsverben online (2009ff.). In: OWID – Online Wortschatz-Informationssystem Deutsch, hg. v. Institut für Deutsche Sprache, Mannheim, www.owid.de/. . ..html.

[14] Herberg, D.; Kinne, M.; Steffens, D. (2004). Neuer Wortschatz. Neologismen der 90er Jahre im Deutschen. Unter Mitarbeit von E. Tellenbach und D. al-Wadi (Schriften des Instituts für Deutsche Sprache 11). Berlin/New York: de Gruyter.

[15] Klettke, M.; Meyer, H. (2002). XML und Datenbanken. Heidelberg: dpunkt-Verlag.

[16] Kämper, H. (2005). Der Schulddiskurs in der frühen Nachkriegszeit. Ein Beitrag zur Geschichte des sprachlichen Umbruchs nach 1945 (Studia Linguistica Germanica 78). Berlin/New York: de Gruyter.

[17] Kämper, H. (2007). Opfer – Täter – Nichttäter. Ein Wörterbuch zum Schulddiskurs 1945–1955. Berlin/New York: de Gruyter.

[18] Kommunikationsverben online (2009ff.). In: OWID – Online Wortschatz-Informationssystem Deutsch, ed. by Institut für Deutsche Sprache, Mannheim (forthcoming).

[19] Lemberg, I.; Schröder, B.; Storrer, A. (Hgg.): Chancen und Perspektiven computergestützer Lexikographie. Hypertext, Internet und SGML/XML für die Produktion und Publikation digitaler Wörterbücher. Tübingen: Niemeyer, 2001. (Lexicographica: Series maior 107).

[20] Loney, K. (2009). Oracle Database 11g: The Complete Reference. New York: McGraw-Hill.

[21] Muench, S. (2000). Building Oracle XML Applications. Sebastopol, CA: O'Reilly.

[22] Müller-Spitzer, C. (2008a). Der texttechnologische Aufbau von OWID. In: Klosa, Annette (Hg.): Lexikografische Portale im Internet. (= OPAL Sonderheft 1/2008). Mannheim: Institut für Deutsche Sprache, 2008. (*OPAL – Online publizierte Arbeiten zur Linguistik* 1/2008), S. 45–55. http://www.ids-mannheim.de/pub/laufend/opal/.

[23] Müller-Spitzer, C. (2008b). The Lexicographic Portal of the IDS. Connecting Heterogeneous Lexicographic Resources by a Consistent Concept of Data Modelling, In: Bernal, Elisenda/DeCesaris, Janet (eds.): Proc. of the XIII Euralex Int'l Congress. Barcelona: Institut Universitari de Linguistica Aplicada/Universitat Pompeu Fabra. S. 457–461.

[24] Neologismenwörterbuch (2005ff.). In: OWID – Online Wortschatz-Informationssystem Deutsch, hg. v. Institut für Deutsche Sprache, Mannheim, www.owid.de/Neologismen/index.html.

[25] OWID – Online-Wortschatz-Informationssystem Deutsch (2008ff.), hg. v. Institut für Deutsche Sprache, Mannheim, www.owid.de.

[26] Schneider, R. E-VALBU: Advanced SQL/XML processing of dictionary data using an object-relational XML database. In: SDV – Sprache und Datenverarbeitung/Int'l Journal for Language Data Processing. Vol. 32.1/2008. S. 35–46.

[27] Schneider, R. Oracle 11g in der Praxis. In: iX – Magazin für professionelle Informationstechnik, Heft 12/2007. S. 86–90.

[28] Schneider, R. All inclusive. Native XML-Unterstützung in Oracle. In: iX – Magazin für professionelle Informationstechnik, Heft 12/2006. S. 146–150.

[29] Schumacher, H.; Kubczak, J.; Schmidt, R. (2004). VALBU – Valenzwörterbuch deutscher Verben. Tübingen: Narr.

[30] Wörterbuch-Portal, http://www.woerterbuch-portal.de.

[31] Your Dictionary.com, http://www.yourdictionary.com.

Alle Webseiten wurden am 01.04.2009 zum letzten Mal überprüft.

Manuskripteingang: 1. Februar 2009

Dr. Carolin Müller-Spitzer arbeitet als Linguistin am Institut für Deutsche Sprache (IDS) in Mannheim und leitet dort das Projekt OWID. Sie koordiniert die Projekte hinsichtlich ihrer Integration im Portal, konzipiert die weitere Entwicklung von OWID sowie die XML-basierte Modellierung. Daneben leitet sie ein Projekt zu „Benutzeradaptiven Zugängen und Vernetzungen in *elexiko*", in dem die Benutzung elektronischer Wörterbücher auf breiter empirischer Grundlage untersucht und neuartige Zugriffsstrukturen auf die lexikografischen Inhalte entwickelt werden sollen.

Adresse: Institut für Deutsche Sprache, 68161 Mannheim,
E-Mail: mueller-spitzer@ids-mannheim.de

Dr. Roman Schneider leitet das Projekt „Texttechnologie und Datenbanken" am Institut für Deutsche Sprache (IDS) in Mannheim und verantwortet dort die Datenbanksysteme für digitale Wörterbücher, Bibliografien und Hypertexte. Neben der automatisierten Informationserschließung beschäftigt er sich mit Text Mining für semistrukturierte Daten, Fragen der Benutzeradaptivität, sowie mit semantischen Konzepten zur Wissensrepräsentation.

Adresse: s. o.,
E-Mail: schneider@ids-mannheim.de