# Exploring Large Document Repositories with RDF Technology: The DOPE Project

**Heiner Stuckenschmidt and Frank van Harmelen,** *Vrije Universiteit Amsterdam*

**Anita de Waard, Tony Scerri, Ravinder Bhogal, Jan van Buel, and Ian Crowlesmith,** *Elsevier*

**Christiaan Fluit, Arjohn Kampman, and Jeen Broekstra,** *Aduna*

**Erik van Mulligen,** *Collexis and Erasmus University*

***This thesaurus-based search system uses automatic indexing, RDF-based querying, and concept-based visualization of results to support exploration of large online document repositories.***

Innovative research institutes rely on the availability of complete and accurate information about new research and development. Information providers such as Elsevier make it their business to provide the required information in a cost-effective way. The Semantic Web will likely contribute significantly to this effort because it facilitates access to an unprecedented quantity of data. The DOPE project (Drug Ontology Project for Elsevier) explores ways to provide access to multiple life-science information sources through a single interface.

With the unremitting growth of scientific information, integrating access to all this information remains an important problem, primarily because the information sources involved are so heterogeneous. Sources might use different syntactic standards (syntactic heterogeneity), organize information in different ways (structural heterogeneity), and even use different terminologies to refer to the same information (semantic heterogeneity). Integrated access hinges on the ability to address these different kinds of heterogeneity.

Also, mental models and keywords for accessing data generally diverge between subject areas and communities; hence, many different ontologies have emerged. An ideal architecture must therefore support the disclosure of distributed and heterogeneous data sources through different ontologies. To serve this need, we've developed a thesaurus-based search system that uses automatic indexing, RDF-based querying, and concept-based visualization. We describe here the conversion of an existing proprietary thesaurus to an open standard format, a generic architecture for thesaurus-based information access, an innovative user interface, and results of initial user studies with the resulting DOPE system.

## Thesaurus-based information access

Thesauri have proven to be essential for effective information access. They provide controlled vocabularies for indexing information and thereby help to overcome many free-text search problems by relating and grouping relevant terms in a specific domain. Thesauri in the life sciences include MeSH, produced by the US National Library of Medicine (www.nlm.nih.gov/mesh/meshhome.html) and EMTREE, Elsevier's life science thesaurus (www.elsevier.com/homepage/sah/spd/site).

These thesauri provide access to information sources (in particular document repositories) such as PubMed (http://pubmed.org) and EMBASE.com (http://embase.com), but currently no open architecture exists to support using these thesauri for querying other data sources. For example, when we move from centralized, controlled use of EMTREE within EMBASE.com to a distributed setting, we must improve access to the thesaurus with a standardized representation using open data standards that allow for semantic qualifications. RDF (Resource Description Framework) is such a standard.

Elsevier maintains the EMTREE thesaurus as a termi-


1094-7167/04/$20.00 © 2004 IEEE
*Published by the IEEE Computer Society*

nological resource for life science researchers. EMTREE is used to index EMBASE, a human-indexed online database. EMTREE currently contains the following information types.

- *Facets* are broad topic areas that divide the thesaurus into independent hierarchies.
- Each facet consists of a hierarchy of *preferred terms* used as index keywords to describe a resource's information content. Facet names are not themselves preferred terms, and they cannot be used as index keywords. A term can occur in more than one facet; that is, EMTREE is poly-hierarchical.
- Preferred terms are enriched by a set of *synonyms*—alternative terms that can be used to refer to the corresponding preferred term. A person can use synonyms to index or query information, but they will be normalized to the preferred term internally.
- *Links*, a subclass of the preferred terms, serve as subheadings for other index keywords. They denote a context or aspect for the main term to which they are linked. Two kinds of link terms, drug-links and disease-links, can be used as subheadings for a term denoting a drug or a disease.

EMTREE 2003 contains about 45,000 preferred terms and 190,000 synonyms organized in a multilevel hierarchy. The EMTREE thesaurus serves primarily as a normalized vocabulary for matching user requests against documents in the target sources. This project uses natural language technology provided by Collexis (www.collexis.com)[1] to automatically index documents in several different repositories with keywords from EMTREE. A Collexis *fingerprint* server houses the results and can be queried via a SOAP interface. (A Collexis fingerprint is very small representation of the characteristic concepts in a piece of source text.)

Natural language frequently refers to the same concept in several ways. The SOAP interface contains an indexing engine that uses EMTREE's synonym relations to return keywords most likely to be relevant to a given search input string. Also, EMTREE's hierarchical relations can identify keywords more specific than the target keyword, letting users expand their searches and thus gain much better recall. The results are ordered by relevance.

Among our challenges was identifying the minimal set of metadata (from each source) to be stored. The user interface assumes that several metadata are available for retrieval or display. The DOPE prototype uses indexes of the full content of ScienceDirect (full-text articles) and the last 10 years of Medline. These sources have different sets of metadata, and future DOPE versions will standardize them using the Dublin Core Metadata Initiative (http://dublincore.org). In general, however, DOPE permits easy inclusion of new data sources.

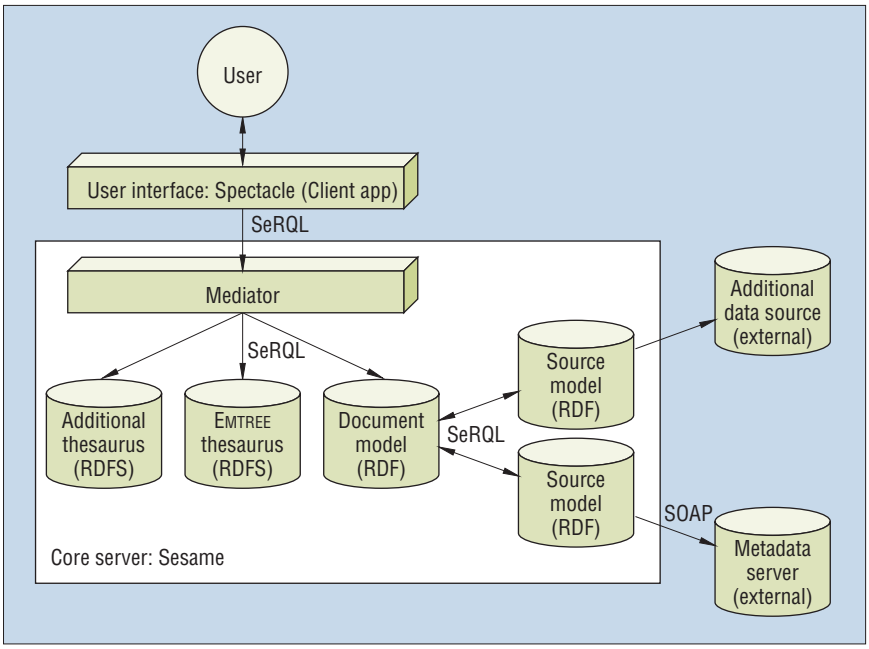


**Figure 1. Basic schematic of the DOPE architecture (protocols and data formats are in parentheses).**

## RDF-based information access

To provide this functionality, we needed a technical infrastructure to mediate between the information sources, thesaurus representation, and document metadata stored on the Collexis fingerprint server. We implemented this mediation in our DOPE prototype using the RDF repository Sesame.[2] Besides the technical integration, we also had to integrate the different information sources' representations on a syntactic and structural level. Figure 1 shows DOPE's architecture. First, we converted Elsevier's main life science thesaurus, EMTREE 2003, to an RDF schema format. Then, using EMTREE 2003 and the Collexis fingerprinting technology, we indexed several large data collections (five million abstracts from the Medline database and about 500,000 full-text articles from Elsevier's ScienceDirect).

In addition to the fingerprints (a list of weighted keywords assigned to a document), the Collexis server houses metadata about the document such as authors and document location. DOPE dynamically maps the Collexis metadata to an RDF model in two steps. The first transformation creates an RDF model, an exact copy of the data structure provided by the fingerprint server. The final model is a conceptual document model used for querying the system. An RDF database, using the SOAP protocol, communicates with both the fingerprint server and the RDF version of EMTREE. A client application interface lets users interact with the document sets indexed by the thesaurus keywords using SeRQL (an RDF rule language) queries sent by HTTP. The system design permits the addition of new data sources, which are mapped to their own RDF data source models and communicate with Sesame. We can add new ontologies or thesauri, which can be converted into RDF schema and also communicate with the Sesame RDF server.

We achieved syntactic interoperability by converting all relevant sources to RDF.[3] In particular, we produced an RDF version of the EMTREE thesaurus. Representing the thesaurus hierarchy as an RDF schema[4] class hierarchy lets us use Sesame's reasoning abilities to expand user queries to narrower keywords. We addressed the problem of structural heterogeneity among sources using transformations on the RDF information representation. We implemented these transfor-

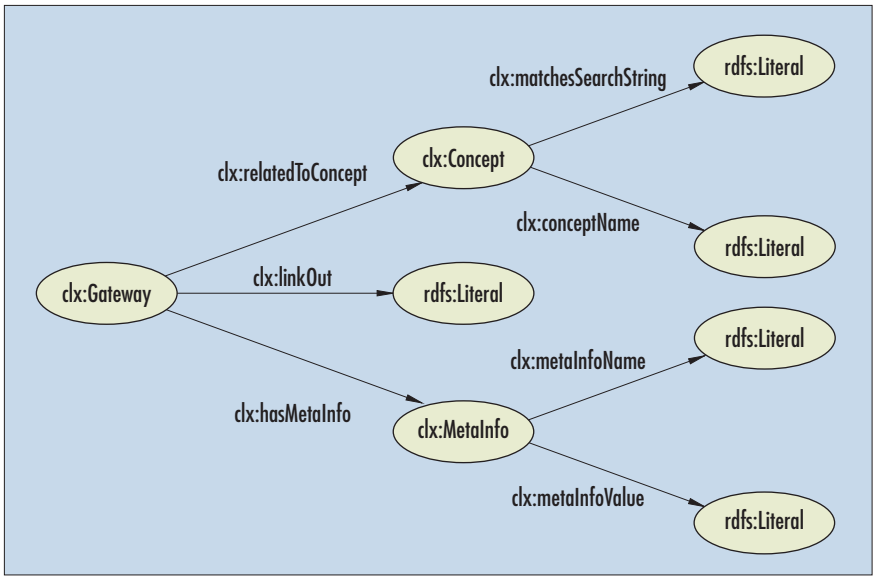


**Figure 2. The physical model: an ontology in Collexis terminology.**

mations using the Sesame query language SeRQL,[5] which also supports queries that output an RDF model differing structurally from the queried model. These "construct queries" serve, for example, to communicate with the Collexis server's fingerprint server, as we'll describe later.

The Collexis server is simply an information repository and isn't equipped with RDF-based input and output facilities. The DOPE prototype thus deploys an extractor component that uses the Collexis SOAP interface to convert the available information to RDF, creating a physical model (Figure 2) that is a one-to-one mapping to the original information.

Although the physical model is already in RDF, it isn't in the terminology in which the queries are formulated, and also isn't well suited to direct merging with different data sources. We therefore use the SeRQL query and transformation language to transform the physical model into a logical model (see Figure 3). The logical model is based on an adapted subset of the OntoWeb ontology (http://ontoWeb.aifb.uni-karlsruhe.de/Ontology). In particular, we simplified the author information representation and linked the model to the schema used to represent the EMTREE thesaurus. This link appears in the lower part of Figure 3. Each publication links to an RDF schema class that represents a preferred term in the thesaurus. Each publication is also annotated with a label and a relation to similar search strings that the Collexis server computes on the fly when it processes a query.

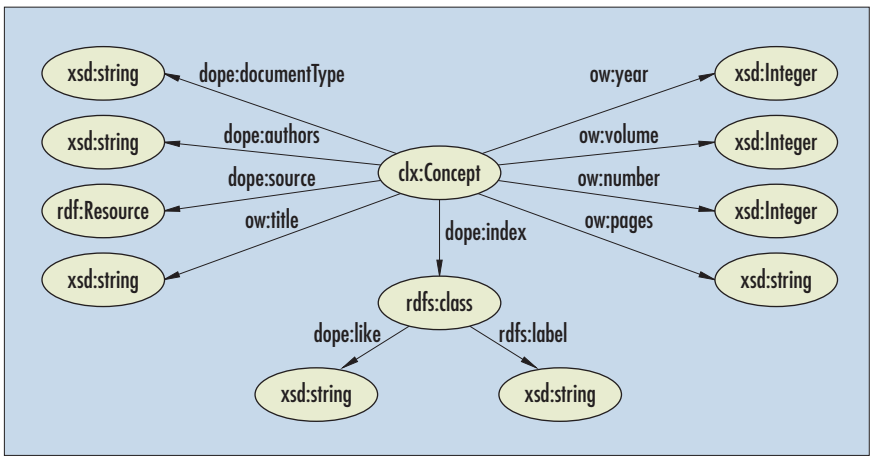


**Figure 3. The logical model: an abstracted ontology over multiple sources.**

For example, a user interested in documents about "AIDS" enters the search string in the DOPE client. To disambiguate the search string (that is, find the relevant thesaurus keyword concept), the client sends the following SeRQL query (uppercase terms denote variables as long as they don't occur in quotes[5]):

```
SELECT
  ConceptName, Concept
FROM
  {Concept} <dope:like> {"AIDS"};
    <rdfs:label> {ConceptName}
```

The query triggers the mediator from Figure 1 to start extracting keywords that, according to the Collexis server, match the phrase "AIDS." The Collexis server returns the keywords as an XML document, which the Extractor translates to an RDF model in the following form (note that, unlike XML query languages, the order of statements is irrelevant):

```
<emtree:35079> <clx:matchesSearchString> "AIDS".
<emtree:35079> <rdf:type> <clx:Concept>.
<emtree:35079> <clx:conceptName> "Acquired
   Immune Deficiency Syndrome".
<emtree:49320> <clx:matchesSearchString> "AIDS".
<emtree:49320> <rdf:type> <clx:Concept>.
<emtree:49320> <clx: conceptName > "Visual Aids".
…
```

This RDF model is a physical model and uses terminology from the Collexis RDF schema (Figure 2). The next step transforms the physical model into a new RDF model in terms of the logical schema. It performs this translation using a SeRQL CONSTRUCT query:

```
CONSTRUCT
  {Concept} <rdfs:label> {Name};
    <dope:like> {SearchString}
FROM
  {Concept} <clx:conceptName> {Name};
    <clx:matchesSearchString> {SearchString}
```

Applying this transformation query yields the following result:

```
<emtree:35079> <dope:like> "AIDS".
<emtree:35079> <rdfs:label> "Acquired Immune
  Deficiency Syndrome".
<emtree:49320> <dope:like> "AIDS".
<emtree:49320> <rdfs:label> "Visual Aids".
…
```

This data represents the same information but now in terms of the logical model in which the original SeRQL query was formulated. The logical model is stored in the repository, which now contains the information necessary to answer the DOPE client's query.

The DOPE client receives the requested list of keywords and presents it to the user, who chooses a concept. The DOPE client then sends a new query to retrieve the documents related to the chosen keyword and the documents' metadata:

```
SELECT
  Document, URL, Title, …
FROM
  {Document} <dope:index> {<emtree:35079>};
    <dope:source> {URL};
    <ow:title> {Title};
…
```

Again, the query engine decomposes this query, and the mediator forwards each subcomponent independently to the relevant source. Because this information hasn't been retrieved before, the mediator starts a new extraction process to retrieve it from the Collexis server and translates it in two steps into a logical model.

After retrieving the answer to the second query, the DOPE client needs, for each document, a list of related concepts. The third and final query retrieves these concepts:

```
SELECT
  RelatedConcept, ConceptName, Doc
FROM
  {DOC} <dope:index> {<emtree:35079>,
    RelatedConcept},
  {RelatedConcept} <rdfs:label> {ConceptName}
```

Since the previous query already retrieved and stored the <dope:index> property relations for each document, we can immediately evaluate the query against the logical model. The mediator forwards the call to the logical model directly instead of starting another extraction process from the Collexis server. The <rdfs:label> relations are available in the EMTREE repository, so the mediator forwards the subquery to the repository. The SeRQL query engine reintegrates the distributed results.

## The DOPE browser

Searching a large information space such as that used in DOPE requires more than a technical infrastructure to query available sources. The sheer volume of results will regularly overwhelm users, who often might not even know what to ask for. To address these common information disclosure problems, we had to provide an intelligent interface that guides users in exploring the information space and presents the query results in a structured way.

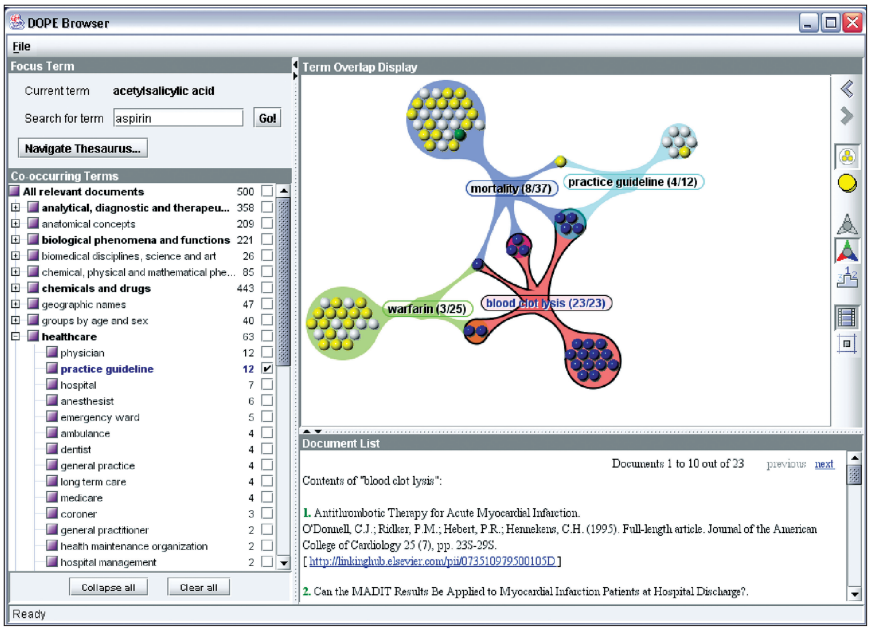


**Figure 4. The DOPE user interface. The left panel shows the current focus term and co-occurring keywords. The right side shows a visualization graph of the document sets for each keyword and their semantic distance, and a list of the documents occurring in the selected part of the graph.**

The user interface client prototype we designed, the DOPE browser, provides querying and navigation using thesaurus-based techniques while hiding back-end complexity such as the existence of multiple data sources or any thesaurus or ontology mapping that may occur. This system presents the user with a single virtual document collection made navigable using a single thesaurus (EMTREE). Each document includes typical document metadata such as title, authors, and journal information. This simplified view into the data makes the user interface easily reusable on other data sets and thesauri. The DOPE browser uses Aduna's thesaurus-driven, interactive visualization technology, the Spectacle Cluster Map,[6,7] for creating overviews of and navigating the available information.

In designing the browser, we assumed end users would find the EMTREE thesaurus too large to navigate directly. Researchers typically focus on an area that can be described by specific terms nested deep inside a thesaurus, but finding their way to these terms might prove difficult. Apart from the cognitive load, manually navigating the thesaurus might also be cumbersome simply because of its size.

Our approach thus lets the user quickly focus on topically related subsets of both the document collection and the thesaurus. First, the user selects a single thesaurus keyword. The system then fetches all documents indexed with this target keyword and also lists any others associated with these documents. These co-occurring keywords provide an interface for the user to explore the set of documents indexed with the focus keyword.

Suppose a user wants to browse through the existing literature on aspirin. He or she first enters the string *aspirin* in the browser's text field (upper left of Figure 4). The system then consults Sesame for all keywords related to this string. It responds with a dialog showing four possible EMTREE terms, asking the user to select one. (This dialog is omitted when only one exact match occurs with an EMTREE keyword.) If the user chooses the keyword **acetylsalicylic acid**, the chemical name corresponding to the brand name, this becomes the new focus keyword. The system consults Sesame again and retrieves up to 500 of the most relevant documents about acetylsalicylic acid, including their metadata fields (such as titles and authors) and the other keywords with which these documents are indexed. The browser presents the co-

occurring keywords in the tree at the screen's left side, grouped by facet keyword (the most generic, broader keyword—that is, the root of the tree they belong to). The user can now browse the tree and select one or more checkboxes that appear by the keywords to generate a visualization of their relations and contents on the screen's right side.

Figure 4 shows the interface after the user has checked the terms **mortality**, **practice guideline**, **blood clot lysis**, and **warfarin**. The visualization graph shows how their document sets overlap. Each sphere in the graph represents an individual document, with its color reflecting the document type such as full article, review article, or abstract. The colored edge between a keyword and a cluster of spheres indicates that those documents are indexed with that keyword. For example, among 25 documents on warfarin, 22 are labeled only with this keyword, two have also been labeled with **blood clot lysis**, and one is about warfarin, blood clot lysis, and mortality. This visualization also shows that within this document set about aspirin, significant overlap exists between the keywords **blood clot lysis** and **mortality**.

Various ways exist to further explore this graph. Users can click on a keyword or a cluster of articles to highlight their spheres and list the document metadata in the lower right panel. Moving the mouse over the spheres reveals the same metadata in a tool tip. Users also can export visualizations to a clickable image map that they can open in a Web browser.

The user starts a new query by typing in a search string. This empties the rest of the interface and loads a new set of documents and co-occurring keywords. The Thesaurus browser provides an alternate starting point for a next query: after selecting a focus keyword, the user can click the Navigate Thesaurus button at the upper left. This brings up a dialog box that lets the user select a new focus keyword by browsing through the thesaurus. The user can iteratively select a broader, narrower, or alternative keyword until arriving at a new focus keyword.

The visualization conveys several types of information. First, the user obviously sees document characteristics such as index terms and article types. Visualizing a set of keywords shows all Boolean combinations without needing to express them all separately. The graph also shows how these keywords relate within the selected document set's scope—that is, whether they overlap and, if so, which documents constitute that overlap. Consequently, the geometric distance between keywords or documents indicates their semantic distance: keywords that share documents appear close together on the graph, as do documents with similar keyword memberships.

To acquire reasonably relevant documents and keywords in a timely manner, the prototype applies thresholds on relevancies and number of results. Inefficient querying and network overhead currently cause some performance bottlenecks. We envision improvements in DOPE's Sesame implementation and browser query mechanism that could make the thresholds on maximum number of documents and keywords unnecessary, or at least orders of magnitude larger.

Most users liked DOPE's way of organizing terms in a two-level hierarchy, though the system lacks support for finding a specific co-occurring term.

Informal tests have indicated that this type of interface stimulates users to explore a large collection of documents. A user can easily create complex queries (visualizing a set of terms effectively visualizes all their Boolean combinations) and build a mental map of the available information.

## A user study

At a 2003 Drug Discovery conference, we gave the DOPE prototype to 10 potential end users, including six academic users (from undergraduate student to professor) and four industrial users (mainly in leading functions). We first gave users a brief overview of the prototype and then asked them to conduct tasks in their domain of expertise. To make the results comparative, we gave all users similar information and asked them to

1. Identify a search term
2. Explore the co-occurring terms
3. Explore the results using the visualization graph
4. Discuss and identify the prototype's potential benefits and problems

Users gave us valuable feedback at each step.

### Identifying a search term

We first asked users to select a term to focus the search and limit the results shown in the DOPE browser. Typical topics users looked for in the system included

- Genomic
- Glucagons
- Kinase
- Diabetes
- Serotonin
- MHC, peptide epitope
- COX-2 cyclo oxygenase
- Hypertension

We found, however, that enforcing a focus term unduly restricted users, who often wanted to use more than one term for focusing. We also observed that users don't always want to start their search using a domain term from the thesaurus but sometimes preferred to start with an author's or journal's name. The current prototype thus seems better suited for exploring the available information space than for searching for a specific article, which would require the option to define more search criteria.

### Exploring the co-occurring terms

Users generally reacted positively to the use of co-occurring terms for narrowing down the search result. They indicated that the additional terms provided useful context information to refine their initial queries. Most users liked DOPE's way of organizing terms in a two-level hierarchy, though the system lacks support for finding a specific co-occurring term.

Users mainly chose terms from the second level. We found that users felt the need for quite specific terms when narrowing down their queries, and some complained that the co-occurring terms were often too general given the quite specific focus term. We conclude that using co-occurring terms to narrow the search is a useful approach but that the interface needs more sophisticated mechanisms for selecting and ordering the terms.

### The visualization tool

Most users reacted positively to the visualization, and many referred to the graph as "this is what I'm thinking about" or "neat

way of jumping between categories." The two main points that emerged from the study were that the visualization provides

- *Richer contextual information about the articles.* Many users inferred that using the graph helped them see information they might otherwise have missed, enhancing serendipitous discovery.
- *Simpler article scanning.* Most users commented that when they scan a list of articles, they are looking for one or two keywords to appear in the article title; if the combination is more complex, it becomes increasingly cumbersome to scan the list effectively. The DOPE visualization tool acts as a reminder and map of their search criteria and alleviates cumbersome scanning.

We identified three issues for improving the visualization:

- *Interpreting the subset names.* In Figure 4, the user has selected four terms in a research area. When we asked users what information they get from this graph, most referred to the terms as labels rather than as the unions of the spheres. Only when we told users about the terms' role and what the spheres represent were they able to easily apply in same visual syntax.
- *Interpreting complex term overlaps.* As with Venn diagrams, the complexity of representation increases rapidly when applying more than three terms. Figure 5 shows how complex the visualization can become when the user selects four terms. No immediate way exists to resolve this in the current visualization, but one possibility might be to limit the number of available terms to alleviate complexity.
- *Manipulating the graph.* The interface permits keyword selection and deselection only on the tree at the graph's left side, but most users tried to interact directly with and manipulate the graph. This behavior indicates the need to support such direct manipulation. Also, users mainly scanned the titles with the graph's rollover feature and paid little attention to the list in the bottom frame.

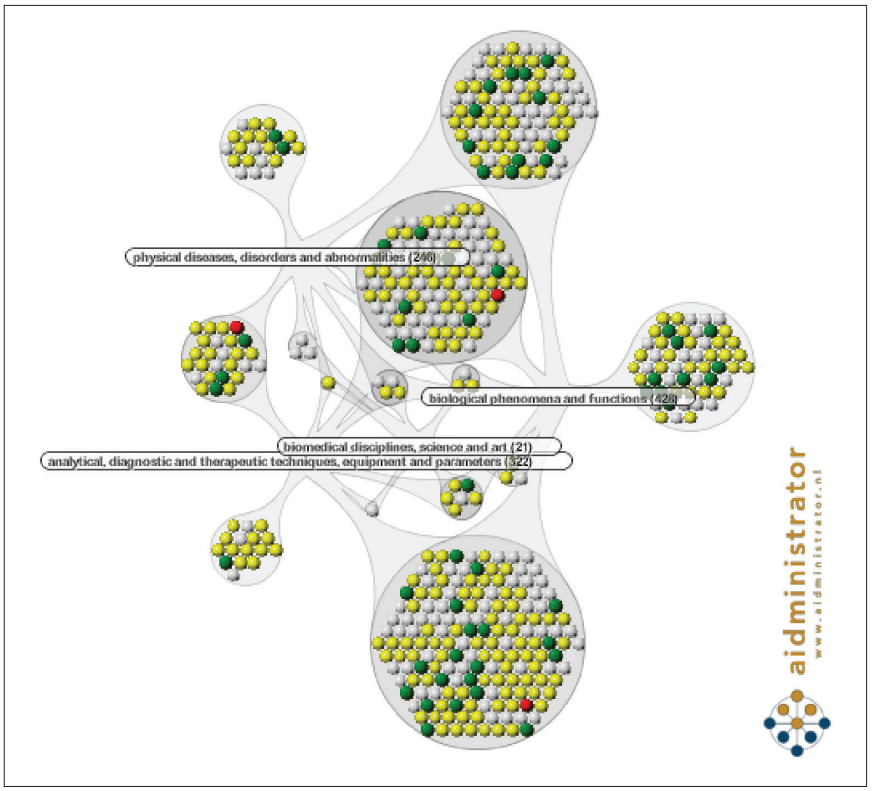


**Figure 5. In this visualization, the user has selected four terms.**

## Potential benefits

Discussions with users reveal that they found the tool useful for the exploration of a large information space. The users mentioned such examples as filtering information when preparing lectures on a certain topic and doing literature surveys (for example, using a "shopping basket" to collect search results). A more advanced potential application mentioned was to monitor changes in the research community's focus. This, however, would require extending the current system with mechanisms for filtering documents based on publication date as well as advanced visualization strategies for changes that happen over time.

The DOPE system is a useful, working implementation of Semantic Web technologies that allows for the inclusion of new distributed data sources and ontologies using the RDF data standard. Current performance problems stem mostly from query procedures between the Sesame system and the Collexis-SOAP interface. We plan to address these problems by expanding DOPE with other data sources and thesauri. Our visualization tool has proven useful for information discovery, although some improvements remain to be made.

To further build on this work, we are pursuing several directions. First, work currently underway at Aduna and the Vrije Universiteit is investigating making the architecture more generic to permit the inclusion of a distributed RDF database. We describe initial investigations of a more general scenario elsewhere.[8]

Second, Elsevier has tested the integration of Collexis fingerprints with a full-text index using the FAST search engine. Due to hardware limitations, the FAST index does not yet include enough records for productive user testing, but we're working to overcome this issue and hope to have results of a comparison between a thesaurus-based and a full-text search available shortly.

Finally, one of the most promising directions for further investigation is the extraction of semantic relations between records,[9] answering such questions as "What diseases does this drug treat?" or "What drugs treat this disease?" The RDF query engine can, in principle, answer such entity-relationship extractions (RDF queries are arguably underutilized in the current setup). We can compare the data with EMBASE records, which contain manually indexed drug and disease links that we can use as a training set. To investigate this promising route, we're exploring further collaboration between all DOPE participants. ■

## Acknowledgments

The Elsevier Advanced Technology Group funded this work. Aduna developed the system under its previous name, Aidministrator BV.

## References

1. E.M. Van Mulligen et al., "Research for Research: Tools for Knowledge Discovery and Visualization," *Proc. 2002 AMIA Ann. Symp.*, Am. Medical Informatics Assn., 2002, pp. 835–839.

2. J. Broekstra, A. Kampman, and F. van Harmelen, "Sesame: An Architecture for Storing and Querying RDF and RDF Schema," *Proc. 1st Int'l Semantic Web Conf.*, LNCS 2342, Springer-Verlag, 2002, pp. 54–68.

3. D. Beckett (ed.), *RDF/XML Syntax Specification (Revised),* W3C Recommendation 10 February 2004, http://www.w3.org/TR/rdf-syntax-grammar.

4. D. Brickley (ed.), *RDF Vocabulary Description Language 1.0: RDF Schema,* W3C Recommendation 10 February 2004, http://www.w3.org/TR/rdf-schema.

A Fill here?

## The Authors

**Heiner Stuckenschmidt** is a senior researcher in Vrije Universiteit Amsterdam's Knowledge Representation and Reasoning Group, which provides scientific and technical consultancy on Semantic Web language, thesaurus representation, and data integration in the DOPE project. He received his PhD in computer science from the Vrije Universiteit. Contact him at Vrije Universiteit Amsterdam, de Boelelaan 1081a, 1081HV Amsterdam; heiner@cs.vu.nl.

**Frank van Harmelen** is head of the Knowledge Representation and Reasoning Group at Vrije Universiteit Amsterdam. He received his PhD in computer science from the University of Edinburgh. Contact him at Frank.van.Harmelen@cs.vu.nl.

**Anita de Waard** is a project manager in the Advanced Technology Group at Elsevier, which is responsible for DOPE project management, requirements analysis, and evaluation of the results. She received her MSc from the University of Leiden. Contact her at a.dewaard@elsevier.com.

**Tony Scerri** is Senior Software Architect in the Advanced Technology Group at Elsevier. He received his BSc in artificial intelligence from the University of Westminster. Contact him at a.scerri@elsevier.com.

**Ravinder Bhogal** is member of Elsevier's User-Centered Design Group, where he is in charge of evaluation of the DOPE project. He received a BEng in information systems engineering from Imperial College London and a MA in interactive media from the Royal College of Art. Contact him at r.bhogal@elsevier.com.

**Jan van Buel** is member of Elsevier's Bibliographic Databases Department, where he is in charge of DOPE project thesaurus management. He received his MSc from Radboud University Nijmegen. Contact him at J.van.Buel@elsevier.com.

**Ian Crowlesmith** is a product manager in the Bibliographic Databases Department at Elsevier, where he is responsible for the development and maintenance of the EMTREE thesaurus. He received his PhD in bacterial genetics from the University of Bristol. Contact him at I.Crowlesmith@elsevier.com.

**Christiaan Fluit** is Lead Visualization Engineer at Aduna, a technology company specializing in tools for intelligent Information management. He is responsible for visualization in DOPE. He received his MSc from the Vrije Universiteit Amsterdam. Contact him at Christiaan.Fluit@aduna.biz.

**Arjohn Kampman** is Senior Developer at Aduna. He is responsible for storage and retrieval on the project. He received his BSc in computer science from Saxonian Universities. Contact him at Arjohn.Kampman@aduna.biz.

**Jeen Broekstra** is Knowledge Engineer at Aduna. He received his MSc in artificial intelligence from the Vrije Universiteit Amsterdam. Contact him at Jeen.Broekstra@aduna.biz.

**Erik van Mulligen** is CTO of Collexis, a technology company specializing in natural language technology for automatic indexing and document retrieval. He is also an assistant professor at the Erasmus University Rotterdam. He received his PhD in medicine from the Erasmus University Rotterdam. Contact him at mulligen@collexis.com.

5. J. Broekstra and A. Kampman, "SeRQL: Querying and Transformation with a Second-Generation Language," technical white paper, Aduna/Vrije Universiteit Amsterdam, Jan. 2004.

6. C. Fluit, M. Sabou, and F. van Harmelen, "Ontology-Based Information Visualization," *Visualizing the Semantic Web*, V. Geroimenko and C. Chen, eds., Springer-Verlag, 2003, pp. 36-48.

7. C. Fluit, M. Sabou, and F. van Harmelen, "Supporting User Tasks through Visualization of Lightweight Ontologies," *Handbook on Ontologies in Information Systems*, S. Staab and R. Studer, eds., Springer-Verlag, 2003, pp. 415-432.

8. H. Stuckenschmidt et al., "Data Structures and Algorithms for Querying Distributed RDF Repositories," *Proc. 13th Int'l World Wide Web Conf.* (WWW 04), ACM Press, 2004.

9. M. Weeber et al., "Generating Hypotheses by Discovering Implicit Associations in the Literature: A Case Report of a Search for New Potential Therapeutic Uses for Thalidomide," *J. Am. Med. Informatics Assoc.*, vol. 10, no. 3, May–June 2003, pp. 252–259.