A Secure Data Repository for Semantic Federation of Product Information

Sandro Reichert
Technische Universität Dresden
Faculty of Computer Science
Institute of Systems Architecture
sandro.reichert@tu-dresden.de

ABSTRACT

A large amount of today's product data is spread into office files and not efficiently searchable. With the integration of unstructured data from distinct sources, recent product information systems will evolve into federated product information systems. A repository is the heart of such systems, storing and managing extracted facts and information.

This paper presents methods to enhance the system's utility by integrating additional quality metadata, covering the information's freshness, origin and quality. Covered security aspects are access permissions on source documents which have to be propagated to the repository, and privacy issues, arising if personal data like user recommendations are stored. This is a position paper.

Categories and Subject Descriptors

E.2 [**Data Storage Representations**]: Linked Representations; H.3.6 [**Library Automation**]: Large text archives

General Terms

Design, Management, Security

Keywords

Semantic Repository, Metadata, Access Rights, Privacy

1. INTRODUCTION

Product related data and information occur in all steps of the product life cycle. Most of this data is unstructured and therefore not efficiently searchable and manageable. This is a major unsolved problem in IT [3]. Today's product information systems (PIS) and product lifecycle management (PLM) systems focus on the phases research, development, production, and sales. With the integration of data from multiple phases of the product lifecycle, data from different administrative domains has to be integrated, e. g., customer feedback from a product review website as well as technical product details from company-internal sources like databases or office documents. As can be seen, the source data's structure and quality is heterogeneous.

It is presumed that today's PIS evolve into federated product information systems (FPIS) by a) incorporating heterogeneous, i. e., unstructured data sources, and b) extending the coverage of the product lifecycle to phases not covered yet, e. g., customer demand analysis and maintenance. The goal of a FPIS is to get a holistic but simple view on data of all phases of the product life cycle. Figure 1 presents a simplified architecture of a FPIS. The bottom layer shows distinct data sources which may reside in another administrative domain than the FPIS. At the layer above, data is being extracted by several components, specialized on extracting data from un-, semi- and structured sources. The integration of existing, historically grown IT systems like PIS or master data management (MDM) systems is done by a legacy system integrator. Local files like office documents or emails are accessed by a document information extractor. Sources in the Web 2.0 are handled by a web information extractor. Data from wireless sensor networks or Radio Frequency IDentification (RFID) readers in the Internet of Things (IoT) are integrated by an IoT information concentrator. Subsequently, the extracted product data is federated and semantically harmonised. Domain specific knowledge and specialized vocabularies are mapped to a universal ontology and persistently stored in a repository.

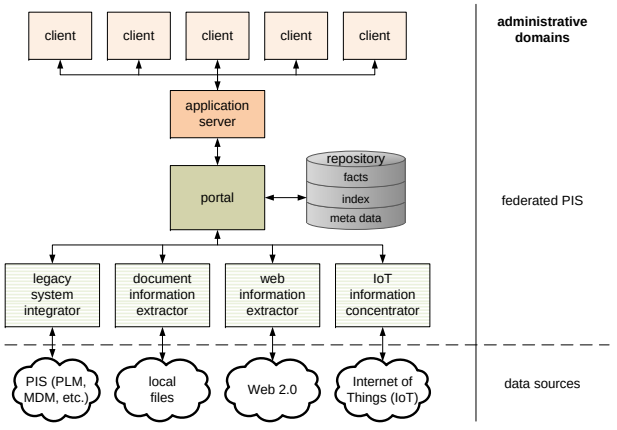

Figure 1: A federated product information system.

This paper focuses on three aspects of a secure data repository which is a central part of a semantic FPIS.

iiWAS2009, December 14-16, 2009, Kuala Lumpur, Malaysia

1. To enable users to assess the presented information's relevance, it is proposed to store additional metadata about the information's freshness, origin, and quality.
2. Existing access permissions on source documents have to be propagated to information in the repository.
3. Privacy issues arising from person related data to be stored are discussed.

Section 2 introduces two scenarios and the requirements on the repository, followed by a state of the art analysis on repositories. Research questions are raised in section 4, including an approach to solve them. Finally, the work plan for the Ph.D. thesis is presented.

2. SCENARIO AND REQUIREMENTS

This section presents two real life use cases from the project this work will be evaluated in, and the resulting requirements to the repository.

2.1 Service Technician Support

For companies developing and installing machines for industrial automatisation, one major branch of business is the long-term support and maintenance for their products. If a machine at customer's site has a malfunction, the customer gets in contact with the call center. His problem description is forwarded to a service technician Charlie who has to plan the actions to be taken, e. g., give detailed support via telephone, send a replacement part or plan a business trip to give on-site support. To do so, Charlie needs detailed information about the machines installed, e. g., geographical position, serial numbers, configuration, and status. Additionally, he needs instruction manuals, certification documents, and reports from past service or maintenance work. If his company had a FPIS providing this information in an all-embracing manner, he would have to spend less time in searching for them which would reduce the overall costs.

As it is not efficient to access all (company-wide distributed) data sources at the time the service technician has a request, a repository for storing the extracted information is required. Since the FPIS concentrates the company's knowledge, strong security restrictions have to be applied to the FPIS in general and the repository in particular. One aspect is to restrict the access to authorized persons only.

2.2 Smart Vendor

Alice: "I am looking for a dress that fits with my red shoes.", Barbara: "I have bought a dress that fits perfectly with my red Manolos.". If Alice and Barbara were friends going shopping together, Alice could easily trust in Barbara's recommendation. These things change, assuming that they do not know each other, Alice is a customer of a smart vendor and Barbara has published her recommendation at a fashion and lifestyle blog.

A smart vendor's business is to sell goods to its customers. Besides reducing the prices, an option to be competitive is to provide detailed information about offered products. To be able to answer Alice's question, he enriches the existing producer's information he already has with customer feedback from his own website as well as with information from other publically available data sources in the web, e. g., recommendation websites or a fashion and lifestyle blog. Using semantic technologies like ontologies and reasoning in the repository, the vendor's FPIS identifies that Manolos are an instance of the concept shoes, so it will present Barbara's recommended dress as a result to Alice's search request.

Several questions remain open: Are publically available data sources like the lifestyle and fashion blog trustworthy? Does Alice want to know whose recommendation she reads [21,22], or more general, are traceability mechanisms required to refer from facts within the repository to the source they are extracted from? If so, which privacy aspects have to be considered? Last but not least, where and how is this metadata stored in the repository?

2.3 Requirements

As seen in the scenarios, three different kinds of data sources exist: structured sources like a product database, semi-structured sources like XML-data, and unstructured sources like office files or pictures. Especially from unstructured sources, the (semi-)automatic information extraction is uncertain and results in fuzzy information. The repository has to be capable to store all different kinds of information: facts, fuzzy information, and a full text index for documents whose information can not be extracted. The following three requirements are the main focus of the author's work:

Information Quality: To store fuzzy information, additional quality information is required. Its possible realization as metadata is discussed in section 4.1.1.

Access Rights: Existing access rights on source documents have to be propagated to and continuously synchronized with the repository. The aspect is analyzed in section 4.2.1.

Privacy Conformance: The FPIS contains personal data so it has to be made sure that privacy policies are not violated. A detailed analysis follows in section 4.2.2.

Another important aspect is the scalability of the repository. Can the system deal with large amounts of data? What is the effect of including quality information (metadata) into the repository? Reasoning is required to make implicit knowledge explicit, e. g., to identify that Manolos are shoes (sec. 2.2). Last but not least, classical requirements like the integrity of the repository have to be considered for the overall system design but are out of focus of this paper.

3. REPOSITORY TOOLS

Basically, a repository is used to store information and objects. Three partially overlapping categories of repository tools can be identified. **Semantic repositories** extend the functionality of common repositories to cover semantic descriptions, i. e., to store statements about objects. A **reasoner** is used to infer logical consequences from a given set of axioms or facts. A tool which integrates front-end tools, back-end tools, and provides mechanisms to access data sources is called **middleware**. A recent, detailed analysis of state of the art repositories has been done in [2].

3.1 Semantic Repositories

Boca's [12] comprehensive list of features, including, e. g., role based access control and versioning, is impressive. The main drawback is its actuality, last modifications in early 2007 suggest that Boca is discontinued. OntoBroker RDF [15] does not support access rights. Oracle Spital 11.g [14] is an Oracle

DB supporting semantic features, use of metadata and access rights whereas their granularity is not named. Sesame [1] is a widely-used Resource Description Framework (RDF) triple store, optimized for storage and requests. Access Rights are provided per repository. YARS2 [9] is focusing on new indexing strategies. Its spare documentation lacks, e. g., in information about access rights. Since Boca seems to be discontinued, Sesame fits the proposed requirements best.

3.2 Pure Reasoners

FaCT++ [26] is a Description Logic reasoner based on tableaux decisions. It is sparely documented, no information about metadata or access rights could be found. As a pure reasoner, OntoBroker [15] is available for the languages F-Logic and OWL. OntoBroker F-Logic supports coarse-grained access rights per ontology, its OWL version does it not at all. Both, Pellet [23] and RacerPro [19] do not support access rights or metadata for reasoning. No pure reasoner satisfies the requirements.

3.3 Combined Repositories and Reasoner

AllegroGraph [8] supports the management of metadata, information on access rights could not be found. Jena [10] was initially developed within the HP Labs Semantic Web Programme. Access rights are not provided, the same applies to OWLIM [16]. Mulgara [18] is optimized to manage metadata but does not provide access rights. For a combined semantic repository and reasoner, AllegroGraph or Mulgara are the best choice in terms of metadata support.

3.4 Middleware

OntoBroker Server [15] integrates the aforementioned versions into one suite. Access rights can be applied on an ontology and per command basis. TopBraid Live [25] is a comprehensive middleware but mechanisms for access control could not be found. Virtuoso is the one and only tool that supports fine-grained access rights mechanism: row level security [17]. Virtuoso seems to be the best choice.

4. RESEARCH QUESTIONS & APPROACH

Today's repository tools lack in support for fine-grained access rights or managing metadata. This section elaborates research questions on how to integrate quality metadata into a repository and discusses resulting security issues.

4.1 Information Quality

The goal of a FPIS is to provide an all embracing view on product information, i. e., present facts and documents most relevant to the users' search requests. Metadata is machine readable data about other data and can be used to reflect the quality of the extraction from the source, or to refer from facts in the repository to source documents they are extracted from (traceability). Well-known specifications are, e. g., the Dublin Core Metadata Initiative [6] and the Exchangeable image file format Exif [24].

The first research question is: *How can metadata be used to improve the utility of information stored in the repository?* This question includes concepts and mechanisms for gathering and storing metadata. The author's goals are to a) enhance the ranking mechanisms used for the presentation of search results, and b) facilitate the user to rank the relevance of the presented information subjectively.

4.1.1 Metadata Classification and Examples

Metadata can be classified in explicit and implicit. Explicit metadata, like author or date of creation, are contained in the header of a source document. Implicit metadata are additional information generated by the FPIS. To improve the utility of information stored in the repository, the following metadata is proposed. The list does not claim to be complete and contains a subset of representative elements.

Source Metadata contains the URI of the information's source (origin), date of creation, and the FPIS' ratings of the source's dynamic and the FPIS' trust into the source. Since company-internal sources are more trustworthy than externals, they get a higher ranking.

Author Metadata is the name of the author and an author rating (reputation) in respect to topics or information sources. People trust into persons they know more than into others [21].

Extraction Metadata is information about used extraction techniques, extraction quality like precision of Natural Language Processing, and the date of extraction. By knowing date of extraction, date of creation and source's dynamic, one can infer the information's freshness/actuality.

Data History Versioning is a technique used in office documents or for source code. Integrating versioning into the repository enables the FPIS and users to access an information's history.

For the design of the repository, the two major focus areas will be the granularity and the management of metadata. For both, advantages and disadvantages of different approaches are discussed in the next two sections.

4.1.2 Metadata Granularity

Metadata can be processed at schema (document) or instance (data object) level: fine-grained or coarse-grained. Considering the metadata trust into a data source, there may be a significant difference between rating the trust into a specific extracted information versus rating its information source as a whole. In the smart vendor example (sec. 2.2), one information source is a public fashion and lifestyle blog. In Web 2.0, an increasingly large number of authors are publishing their ideas, whereas they have various knowledge about the products they are writing about and also various motivations for doing so. The advantage of using fine-grained metadata is the potential to rate information extracted from an expert higher than from a novice. The disadvantage is the complexity of rating every single information. Rating the source as a whole reduces this but results in an average rating for all information contained. The FPIS can not distinguish between information from an expert or novice, both have the same weight. On the other hand, in the service technician scenario (sec. 2.1), a product database is used company-internal only and therefore trustworthy—rating the source as a whole might be sufficient.

The research questions to be answerd are *methods for defining the required granularity of meta d on the information source, and algorithms for the rating of sources.*

4.1.3 Metadata Management

Another important research question is the management of metadata within the repository. Metadata can be stored together with the information they belong to, or be generated at runtime when requested.

An advantage of storing metadata and information together is, that requests on metadata can be processed immediately. The main disadvantage is the large amount of additional (meta)data that has to be stored in the repository. Resource consumption for managing both, product data and metadata may be more than twice as high as for managing product data solely. An opposite approach is to generate and store only a minimum amount of metadata required to compute the remainder when requested. The advantage is the reduction of storage requirements within the repository, resulting in the disadvantage of longer response times when requesting metadata which needs to be computed at runtime. Especially if a certain metadata is requested and computed repeatedly, scalability problems arise. Furthermore, dealing with fine-grained metadata intensifies scalability issues.

The challenge in designing methods which metadata have to be stored and which can be computed at runtime has some similarities to caching strategies. Both approaches will be analyzed and the most promising one will be implemented in the Ph.D. thesis. For the realization of metadata within a repository, the upcoming OWL2 standard [13] seems promising since it provides annotations on axioms.

4.2 Security Aspects

The three goals in Information Security are integrity, confidentiality and availabilty. Integrity means that information are correct, complete, and up to date or the opposite is clearly recognizable. Confidentiality means that access is granted to authorized subjects only, including means for protecting personal privacy. Availability ensures timely and reliable access to and use of information [7]. This paper focuses on confidentiality, more precisely on access control (sec. 4.2.1) and privacy aspects (sec. 4.2.2).

4.2.1 Access Control

A FPIS concentrates a company's knowledge. Access to an information in the repository has to be restricted to those who are authorized to access the source document the information has been extracted from. It is presumed, that the FPIS is integrated into the company's access rights management system, e. g., to use it for authentication purposes, and all documents to be included into the FPIS already have access permissions.

The third research question is: *How can access rights be lifted from source-level to information-level?* This includes the question how and where to store them in the repository, as well as keeping them consistent if access permissions on source documents change.

An apporoach for realising fine-grained access permissions on information is storing them as metadata, so an additional metadata element is added to section 4.1.1: Access Rights. Dublin core already provides entries for access control [6]. It is presumed, that the repository is secure in terms of information security, modifying metadata to obtain access by fraud has to be prevented by other mechanisms out of the paper's focus. Section 4.2.2 presents another application area where access rights are required: personal data.

4.2.2 Privacy

In the European Union, the protection of individuals with regard to the processing of personal data is done in the European directive 95/46/EG [11], in Germany, the Federal Data Protection Law [4] has to be abided. Other countries have similar—more or less restrictive—laws. Of cource, these laws affect the system design of a FPIS.

In the system design, the paper concentrates on protecting the privacy of authors and users inside the FPIS' administrative domain. If an author like Barbara (sec. 2.2) wants to stay anonymous on a public Web 2.0 blog, it is up to her/him to make use of an identity management system. Research in this area is done by projects like FIDIS [20] or PRIME [5].

Regarding personal data generated at runtime by the users of the FPIS or the system itself, scenario 1 is extended. In response to Charlie's request, the FPIS presents information extracted from three documents, annotated with the authors' names: his colleagues Dave and Eric. Charlie usually trusts in Dave's work and selects the related document for further reading. The FPIS logs his choice and increases its internal ranking score for the document to present more relevant/higher quality documents first. The problem concerning privacy is, that Charlie's boss can use several dummy requests to determine which employee works most efficient or is most favored by his colleagues—this is illegal in many countries. To overcome this issue, one approach to be analyzed is to store Charlie's choice not in the repository as metadata, but in his personal profile. This reduces the FPIS' utility since others can not profit from Charlie's indirect recommendation, but increases his privacy. Similar privacy issues arise if users rate presented information or documents actively. This approach is realized by, e. g., Amazon[1] where users write customer reviews and others rate these.

The fourth research question is *The contrast of metadata, i. e., functionality versus privacy: How to restrict access on metadata to protect personal data without loosing functionality?* This includes the usage of existing techniques to make personal data anonymous, e. g., to provide traceability from product usage to groups of persons but not to individuals.

5. CONCLUSIONS AND WORK PLAN

The paper presented an approach to solve the problem of managing unstructured data within a FPIS by integrating additional quality metadata into its repository and discussed arising security issues. So far, an analysis of today's repository tools has been done. It showed their lack in support for managing metadata or fine-grained access rights.

The research questions to be analyzed in the recently started Ph.D. thesis are: *1) How can metadata be used to improve the utility of information stored in a repository? 2) Methods for defining the required granularity of metadata based on the information source, and algorithms for the rating of sources. 3) How can access rights be lifted from source-level to information-level? 4) The contrast of functionality versus privacy: How to restrict access on metadata to protect personal data without loosing functionality?* For each, different approaches will be discussed in detail, the most promising will be used in the system design and prototypically implemented.

To prove the real world awareness, the system will be validated by use case partners in the German research project

[1] http://www.amazon.com

Aletheia[2]. Based on real data, domain experts manually select source documents that fit best to a real request. These documents represent the gold standard. Afterwards, the FPIS extracts information from the same sources, using different configurations: with and without using both, quality metadata or security restrictions developed in the Ph.D. thesis. The FPIS' responses to the aforementioned requests are compared to the gold standard, analyzing recall, precision and overall system performance.

Currently, state-of-the-art and related work are beeing investigated and should be finished within 6 month. The concept, covering all research questions, and the implementation should take 1 year each, followed by an evaluation within the last 6 month. The manuscript is beeing written in parallel.

6. ACKNOWLEDGEMENTS

This research was funded by the German Federal Ministry of Education and Research (BMBF) under grant number 01IA08001F. The responsibility for this publication lies with the author.

7. REFERENCES

[1] Aduna B.V. User guide for sesame 2.2, 2008. http://www.openrdf.org/doc/sesame2/users/, access 08-2009.

[2] A. Becker, M. Walther, S. Reichert, and J. Hladik. D.G4.2 Repository-Spezifikation. Technical report, 2009.

[3] R. Blumberg and S. Atre. The problem with unstructured data. *Information Management Magazine*, 2003.

[4] Bundesministerium der Justiz. Bundesdatenschutzgesetz in der Fassung der Bekanntmachung vom 14. Januar 2003 (BGBl. I S. 66), das zuletzt durch Artikel 1 des Gesetzes vom 14. August 2009 (BGBl. I S. 2814) geändert worden ist, Aug. 2009.

[5] J. Camenisch, A. Shelat, D. Sommer, S. Fischer-Hübner, M. Hansen, H. Krasemann, G. Lacoste, R. Leenes, and J. Tseng. Privacy and identity management for everyone. In *DIM '05: Proceedings of the 2005 workshop on Digital identity management*, pages 20–27, New York, NY, USA, 2005. ACM.

[6] Dublin Core Metadata Initiative Limited. The Dublin Core Metadata Initiative (DCMI) website, 2009. http://dublincore.org, access 08-2009.

[7] C. Eckert. *IT-Sicherheit: Konzepte, Verfahren, Protokolle*. Oldenbourg Wissenschaftsverlag GmbH, 2. auflage edition, 2003. ISBN 978-3-486-58270-3.

[8] Franz Inc. AllegroGraph RDFStore, 2009. http://www.franz.com/agraph/allegrograph/, access 08-2009.

[9] A. Harth, J. Umbrich, A. Hogan, and S. Decker. Yars2: A federated repository for querying graph structured data from the web. In *6th International and 2nd Asian Semantic Web Conference (ISWC2007+ASWC2007)*, pages 211–224, Nov. 2007.

[10] Hewlett-Packard Development Company, LP. Jena Semantic Web Framework, 2009. http://jena.sourceforge.net/, access 08-2009.

[11] K. Hänsch and L. A. Serna. Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, Nov. 1995. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:html, access 08-2009.

[12] IBM Adtech group. BocaUsersGuide, Feb. 2007. http://ibm-slrp.sourceforge.net/wiki/index.php/BocaUsersGuide-2.x, access 08-2009.

[13] B. Motik, B. C. Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz. OWL 2 Web Ontology Language Profiles, W3C Candidate Recommendation 11 June 2009, June 2009. http://www.w3.org/TR/owl2-profiles/, access 08-2009.

[14] C. Murray. *Oracle®Database Semantic Technologies Developer's Guide 11g Release 1 (11.1)*. Oracle, b28397-05 edition, July 2009.

[15] ontoprise GmbH. Ontoprise Helpsystem, 2009. http://www.ontoprise.de/help/index.jsp?topic=/com.ontoprise.ontobroker.help/html/start.html, access 08-2009.

[16] Ontotext AD. OWLIM Semantic Repository, 2009. http://www.ontotext.com/owlim/, access 08-2009.

[17] OpenLink Software. Virtuoso database row level security, Apr. 2006. http://virtuoso.openlinksw.com/Whitepapers/pdf/DMI_Virtuoso_RowLevel_Security.pdf, access 08-2009.

[18] Project Website. Mulgara Semantic Store, 2009. http://www.mulgara.org/, access 08-2009.

[19] Racer Systems GmbH & Co. KG. *RacerPro Reference Manual*, Oct. 2007.

[20] K. Rannenberg, D. Royer, and A. Deuker, editors. *The Future of Identity in the Information Society*. Springer, 2009.

[21] R. R. Sinha and K. Swearingen. Comparing recommendations made by online systems and friends. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001.

[22] R. R. Sinha and K. Swearingen. The role of transparency in recommender systems. In *CHI '02: CHI '02 extended abstracts on Human factors in computing systems*, pages 830–831, New York, NY, USA, 2002. ACM.

[23] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical owl-dl reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51–53, 2007.

[24] Technical Standardization Committee on AV&IT Network Technology. Exchangeable image file format for digital still cameras: Exif Version 2.2, Apr. 2002.

[25] TopQuadrant, Inc. Topbraid live^{TM}, 2009. http://www.topquadrant.com/products/TB_Live.html, access 08-2009.

[26] D. Tsarkov and I. Horrocks. Fact++ description logic reasoner: System description. In *Third International Joint Conference on Automated Reasoning, IJCAR 2006*, pages 292–297, 2006.

[2] http://www.aletheia-projekt.de