Knowledge Engineering - From Front-line Support to Preliminary Design

Sylvia C Wong, Richard M Crowder, Gary B Wills, and Nigel R Shadbolt
School of Electronics and Computer Science
University of Southampton, UK
sw2@ecs.soton.ac.uk, rmc@ecs.soton.ac.uk, gbw@ecs.soton.ac.uk, nrs@ecs.soton.ac.uk

ABSTRACT

The design and maintenance of complex engineering systems such as a jet engine generates a significant amount of documentation. Increasingly, aerospace manufacturers are shifting their focus from selling products to providing services. As a result, when designing new engines, engineers must increasingly consider the life-cycle requirements in addition to design parameters. To identify possible areas of concern, engineers must obtain knowledge gained from the entire life of an engine. However, because of the size and distributed nature of a company's operation, engineers often do not have access to front-line maintenance data. In addition, the large number of documents accrued makes it impossible to examine thoroughly. This paper presents a prototype knowledge-based document repository for such an application. It searches and analyzes distributed document resources, and provides engineers with a summary view of the underlying knowledge. The aim is to aid engineers in creating design requirement documents that incorporate aftermarket issues. Unlike existing document repositories and digital libraries, our approach is knowledge-based, where users browse summary reports instead of following suggested links. To test the validity of our proposed architecture, we have developed and deployed a working prototype. The prototype has been demonstrated to engineers and received positive reviews.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web-based services*; H.3.7 [**Information Storage and Retrieval**]: Digital Libraries; J.2 [**Computer Applications**]: Physical Sciences and Engineering—*Engineering*

General Terms

Design

DocEng'06, October 10–13, 2006, Amsterdam, The Netherlands.

Keywords

Intelligent Documents, Semantic Web, Service-Oriented Architecture

1. INTRODUCTION

The design and maintenance of large and complex engineering systems requires a significant amount of documentation, particularly if the system being considered is a turbofan jet engine used on the current generation of aircraft. The jet engine is amongst the most complex machine ever designed, incorporating a wide range technologies including high temperature materials, complex fluid dynamics and high speed rotating components.

A fundamental shift is currently occurring in the aerospace industry away from selling products to providing services. Companies such as Rolls-Royce aims to make half its engine fleet subject to long-term maintenance service agreements by 2010 [12]. Essential to the success of this market shift is the significant cultural change from *offering a service to support a product* to *designing a service and the product to support it* [12]. In other words, new products must be designed to provide lower and more predictable maintenance costs. To minimize maintenance costs through out the engine's life cycle, engineers must obtain knowledge gained from maintenance histories of similar products during the design phase of new products. This will help engineers identify parts most likely to be problematic throughout the engine's entire life cycle. It should be noted that engine design is typically undertaken by a number of teams who are responsible for individual engine modules, e.g compressor, turbine. Therefore it is impossible for any single member of a design team to access more than a fraction of available documentation. As is widely recognized, information systems usually develop over time into a set of heterogeneous resources. As a result, it becomes difficult for engineers to follow a trail through the resources [26]. The challenge for organizations is therefore to develop an information system that is both comprehensive and will satisfy the increasing demands from industry for up-to-date and easily accessible information.

In response to these challenges, we are implementing an intelligent, knowledge-based document repository to support engineers to design for the aftermarket. The intelligent document repository searches and analyses relevant maintenance records and design guidelines, and provides design engineers easy access to such information. It is hoped that the summary reports provided by the intelligent document repository will help engineers in creating design documents

that incorporate aftermarket issues into the design requirements.

This paper is organized as follows. Section 2 explains how knowledge can be re-used within a aeroengine manufacturing company. It also introduces a scenario explaining how maintenance records can be used to help to improve the reliability of both existing and new products. Section 3 gives a brief overview of knowledge management. Section 4 then discusses other works that aims to help user discover knowledge by integrating heterogeneous documents sources. Section 5 describes our proposed architecture for an intelligent document repository. Discussion can be found in Section 6 and conclusions in Section 7.

2. SCENARIO

As is well recognized in engineering design, the use of past experiences and previously acquired knowledge, either from the designer's own experiences or from resources within their organization forms an important part of the design process. It has been estimated that 90% of industrial design activity is based on variant design [9], while during a redesign activity up to 70% of the information is taken from previous solutions [15]. A cursory consideration of these figures identified two immediate challenges – how to capture knowledge, and how to retrieve it. The purpose of the work reported in this paper is to develop an intelligent document repository that can be used within a manufacturing organization for the retrieval of knowledge from across the organization to support the design activities.

Figure 1 shows the key information flow for the different stages in the life of an aero-engine. Concept design is the first stage of an engine's life cycle. Given a set of broad requirements, such as thrust, range of the target aircraft and fuel burn, engineers determines the approximate dimensions, weight, power and other physical characteristics for the engine design. The engineers also make estimates of the manufacturing costs of the engine. In the design stage, engineers transform the preliminary abstract design into a set of concrete plans that can be used in production. In production, engines are built according to the design plans. Traditionally, after production and sales, responsibility for the engine passed from the manufacturer, to the airlines, who own the engines. The airlines are responsible for maintaining the engines. This maintenance activity is supported by the manufacturer's technical support and operations team. To assist maintenance engineers to identify problems before a breakdown occurs, engines are commonly equipped with sensors for engine monitoring. This monitoring information can be analysed for abnormal operating conditions, such as temperature or pressure. However until fairly recently, the monitoring data was only used to support maintenance activities, even though it is a rich source of information for the designers of future engines.

As can be seen in Figure 1, there are interactions and information flow between neighbouring stages in the production maintenance process. This is due to the iterative nature of engineering processes. Design knowledge is also passed to operations in the form of 'owner's manuals'. Sometimes, information also pass between unconnected stages, for example, between operations and engine concept. However, the flow is weak and may take the form of informal and personal networking between engineer.

While the process works very well, it does have significant

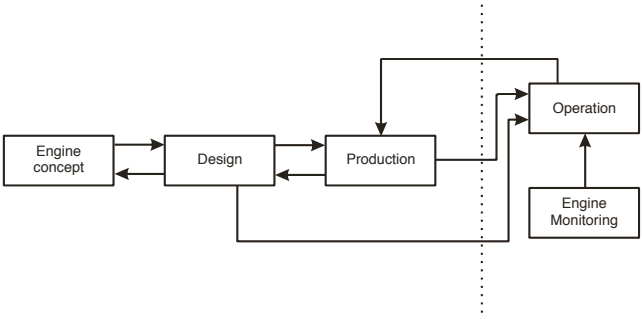

Figure 1: Information flow between the different stages in the life of an aero-engine. The vertical line between production and operation represents the transfer of the engine from its manufacturer and an airline. Operations is the generic term for maintenance and aftersale support.

disadvantages. In particular, design engineers are remote from the problems experienced in the field by operations. Due to the importance of increasing operational reliability and minimizing maintenance costs in the new market paradigm of product support, information gained in the operation of a fleet of engines needs to be fed back to the designers of subsequent engines. However, the current information infrastructure makes this difficult as concept design engineers do not have access to maintenance knowledge. Similarly, design engineers should consult existing maintenance documents to help design parts with more predictable maintenance costs.

As a result, we need to strengthen and help formalized the information flow between the company's aftermarket operations and the design teams. Figure 2 shows the information and knowledge flows our research aims to build. This would allow the knowledge gained during the design, production and operation of an engine to advice the design of the next variant.

The following scenario[1] illustrates the potential use and benefit from such a document repository. The scenario involves three separate and different groups of users that are involved in the life of a jet engine. Front-line maintenance engineers are involved in the day to day servicing of the engines, and thus responsible for populating the document repository with maintenance reports and other similar documents.

> *During the regular pre-flight checks, a flight crew reported a problem with the leak from an engine's bleed air system. Subsequent inspection which required the removal of the engine revealed that a duct had failed at a joint due to vibration. After repair, the engine was returned to service, and a full maintenance event report submitted to the document repository.*

The document repository can then be used by technical support and operation engineers, who are responsible for improving the performance of existing engines. They can

[1]The scenario is entirely fictitious and does not derive from any real event.

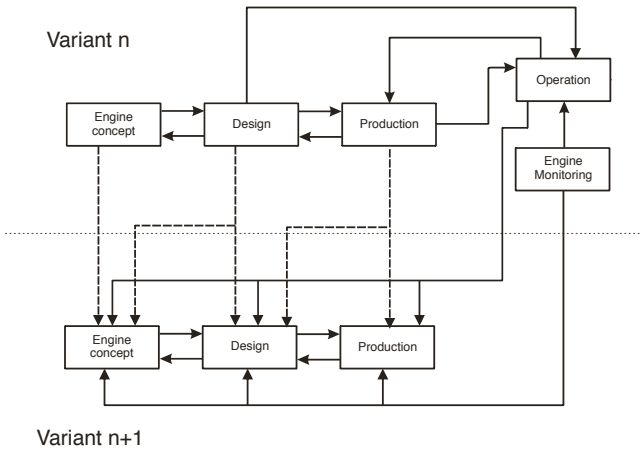

Figure 2: We aim to facilitate the flow of information gained during the life-cycle of one engine variant to inform the design of the next variant. The dotted arrows indicate the flow of design rationale and similar knowledge. The solid arrows represent all other information flows including real time engine information and design documentation.

use information collected in the repository to monitor trends that develop over a fleet of engines. Modification can then be designed to mitigate any problems found:

> *Following a review of the maintenance events relating to a specific engine fleet, a trend was noticed in the high than expected number of failure of an air duct joint due vibration. To maintain the reliability of the engine fleet, a modification was developed and implemented.*

The same information in the repository will also be used by design engineers working on a new engine:

> *The design team for the next variant of this engine reviews the performance of the air bleed system across the fleet to learn from previous design rationale and operational history. Finite element analysis showed that a joint failure could occur due to vibration if certain operational conditions were met. It was therefore decided that the future variant of the engine would both eliminate the joint and reroute the duct work. The revised design costs 50% more than the original. However, the saving over the life of the engine will be substantial due to lower likelihood of in-service failure.*

The goal of our work can thus be summarized as follows: To feedback and harvest knowledge gained from the aftermarket operations documents to help (a) operations engineers in designing modifications to existing engines, and (b) design engineers in designing the next variant engine for the aftermarket.

3. KNOWLEDGE ENGINEERING

Organizations have become increasingly concerned with knowledge management [24], amassing large amount of information into their corporate memory [13]. The aim is to use this repository of information to inform future discussions, decisions and activities. Figure 3 shows the four key activities in knowledge management – knowledge creation, knowledge mapping, knowledge retrieval and knowledge use.

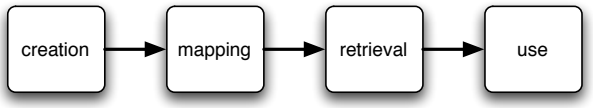

Figure 3: The four key activities in Knowledge Management.

The first step in creating a knowledge system is knowledge creation. In our scenario, this is the documentation written by engineers throughout the life time of an engine, from design to maintenance. In practice it can take many forms, including formal design reports, e-mails submitted by airlines and field service personnel, detailed inspection reports and the engine performance over individual flights recorded by engine monitoring systems.

Knowledge mapping involves creating an *ontology*, which is *a specification of a conceptualization* [10]. Gruber explains that a common ontology defines the vocabulary with which queries and assertions are exchanged among agents (people or software). The ontology sets out all the entities (objects or concepts) that we are interested in and the relationships that connect these entities together. This is intended to be a *pragmatic* definition, i.e. it defines the vocabulary that is actually *in use*, and the concepts that are *useful* in problem-solving. It does not give the deep underlying philosophical vision of the fundamental entities in the field. Hence, in knowledge management, an ontology is a tool, whose quality is entirely dependent on its usefulness.

Retrieval is the step that transform mere *information* into *knowledge*. We believe that knowledge is relevant information delivered at the right time and context [19]. To deliver this knowledge, information needs to be semantically enriched so that it can be better reused. When a knowledge system has a shared ontology for its disparate information resources, software agents can handle the semantically enriched resources consistently. Thus semantics and ontology together help deliver the right information at the right time, hence generating knowledge.

In our scenario, knowledge use occurs when engineers apply the knowledge gained from the repository in their work. For example, a design engineer applies the knowledge gained to create a better engine for the aftermarket. Or an operations engineer discovers a recurring, but minor problem with an existing engine, which would indicate a larger problem than each individual incident suggests.

4. RELATED WORK

The work described in this paper is an extension on our previous work with Rolls-Royce. In [7], we presented a

future vision for the working practices of designers within a manufacturing organization. We have found that engineering design environments are highly distributed in nature and are characterized by a large number of information sources, which together with the designers forms a complex sociotechnical system. It is concluded that a range of knowledge management tools would be required to support this future vision of engineering design environments [25]. Therefore one of the objectives of our current work is to define a future engineering design environment, with particular emphasis on the social and technical systems that will support designers in their day-to-day activities.

In [26], we created a document repository from distributed and heterogeneous engineering document resources. When an engineer searches for documents within the repository, the system generates a list of documents ordered according to the engineer's role and its related concepts. Thus, the document retrieval process is *intelligent* and adapts to the user's role within the company. However, we found that engineers actually want the knowledge that is buried within the documents, instead of the actual documents themselves. This is due to the large volume of documents available within the repository which makes it very time consuming to peruse thoroughly. In other words, engineers prefers to see *summary reports* of documents archived. For example, when searching for engine part failures, engineers want to see how many times the failure of a particular part leads to engine removal, but not the list of original maintenance documents. Another example of useful knowledge that can be extracted from engineering documents is an "expertise finder" [8]. The expertise knowledge can be obtained by integrating author information and the contents of documents. Thus, for our new engineering document repository, we aim to include the ability to provide analysis of information stored, in addition to simple document search.

The goal of this work is similar to that of the WISE (Web-Enabled Information Services for Engineering) project described by Boy and Barnard in [1]. However, very little information can be found on the scope and implementation of the project, which makes it difficult to conduct a meaningful comparison.

Our work in [26] can be seen as a digital library, with the extension where information presented is adapted to the role of the user. Digital libraries concentrate on the problem of searching for documents distributed over multiple repositories. For example, Priebe and Pernul [21] developed a portal over multiple document repositories by using an integrated metadata store. As a result, users can search on both the content of the documents and their metadata. In contrast, document index functionality does not form part of our proposed infrastructure. However, global document indexes can be provided to our knowledge repository as *services* that implement document indexes and metadata indexes.

Another area digital libraries concentrate on is dynamic links generation to relevant document resources [2, 20, 16]. Dynamic links are injected into documents automatically during presentation time, and does not alter the original documents. These dynamic links can point to related documents, or even services such as searching annotation and peer reviews [20]. In comparison, our proposed system does not perform dynamic link injection on existing documents. However, it can provide a list of suggested documents as a *service*.

Carr et al extended the concept of digital libraries to include the dissemination process that leads to publications [3]. They have developed the Digital Review Journal (DRJ), which is a knowledge base where users from different roles can collaborate through the entire process on writing an academic paper. It includes support for experiment creation, execution to publication. In comparison, our work concentrates on extracting existing knowledge generated from outside the framework. Also, the DRJ does not handle integrating knowledge from heterogeneous document sources.

Finally, there are also projects working on creating new semantically marked up data for large knowledge repositories, such as [4] and [6]. Creating new documents is outside the scope of our project, as we concentrate on the problem of delivering knowledge from existing documents. However, employing techniques to generate semantic information automatically for new documents to be deposited into the knowledge repository will improve document analysis and searching inside our framework.

5. ARCHITECTURE

Two key technologies are used in integrating the distributed and heterogeneous data sources available from the document repository – Service-Oriented Architecture (SOA) and the Semantic Web. SOA is a software architectural concept that defines the use of services to support the requirements of software users. In a SOA environment, functional components expose service behaviours accessible to other applications via loosely coupled standards-based interfaces. These components, called Web Services, interoperate based on a formal definition independent of the underlying platform and programming language. Due to the nature of loose coupling in a SOA, applications can be developed and deployed incrementally. In addition, new features can be easily added after the system is deployed. This modularity and extensibility make SOA especially suitable as a platform for an integrated knowledge repository within large engineering organizations.

The Semantic Web is an application of the World Wide Web aimed at computational agents, so that *programs*, and not just humans, can interpret the meaning of documents on the Web (or an intranet). This allows the Web to be used for more than a human-browseable repository of information. The basis of this interpretation is an ontology, a structure which forms the backbone of the knowledge interpretation for an application.

Figure 4 shows the interaction flows within our knowledge framework. The user forms a query via a user interface. In the current system design, this user interface is in the form of a web page provided by a web portal. To answer the user's query, the web portal finds and calls the web services that can provide it with relevant knowledge. These web services provide algorithms and functionalities that work on the underlying information set. There are no restrictions on the type of functionalities that the web services can provide, and it can range from an index of available documents, to finding the frequency of maintenance events.

The underlying information set is expressed as RDF triples [17]. RDF is the language for representing information about resources in the Semantic Web. RDF is used to represent both the data itself, and its associated metadata. An aerospace engineering ontology defined in OWL is used to describe the objects and concepts that appear within this

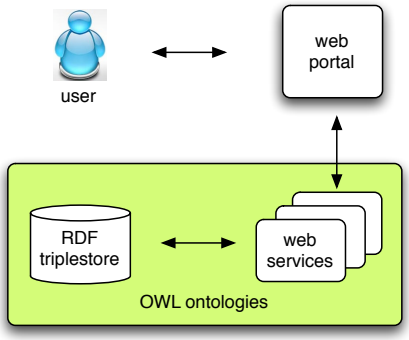

Figure 4: Interaction diagram for our intelligent document repository.

information set. OWL [18] is the standard web ontology language for use between web agents in the Semantic Web. The development of this ontology was greatly helped by the closely controlled language used in the aerospace industry. For example, irrespective of a part's description or name, it can be assigned a unique description based on its location within an engine using the ATA (Air Transport Association) descriptors.

Our intelligent document repository is organized as a SOA, as shown in Figure 5. Existing web standards are used wherever possible, to maximize tool reuse, compatibility and portability. A web browser access the web portal using the HTTP protocol. In turn, the web portal supplies the user interface to the web browser. The user interface is formed by a series of portlets. Portlets are reusable components that display relevant information to portal users. In general, they appear to end users as rectangular sections of a web page offering a limited set of information. In our implementation, the portal conforms to the Java Portlet API standard, JSR 168 (`http://jcp.org/jsr/detail/168.jsp`). To create the information for display, the portlets access one or more web services. We envisaged that these web services are provided by different departments within the company, and can be distributed across multiple sites. These web services will perform more than document indexing that are already available in all document repositories. For example, it can provide an interface to existing manufacturing systems such as PDM (Product Data Management), or allow access to analysis tools such as Life-Cycle Cost Models and Finite Element Analysis. The web service interfaces are defined in WSDL [5], which is the standard interface language for defining web service interfaces. Documents in the repository will be in the form of RDF triples. Both the RDF triples and associated OWL ontologies are stored in 3store [11], an efficient RDF storage implementation that provides query and access abilities over HTTP. As can be seen in the diagram, the architecture does not require all web services to access the underlying information sources via 3store. However, most, if not all, web services will operate on the semantically enabled data within 3store instead of the original heterogeneous documents.

The next three sections discusses each component within this architecture in greater details.

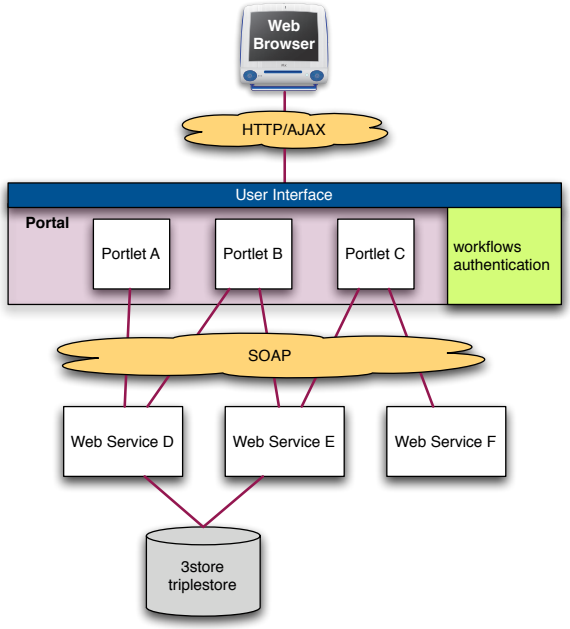

Figure 5: System Architecture

5.1 Documents and Ontology

For each maintenance event, maintenance engineers document information surrounding the event for later references. A maintenance event occurs whenever actions are performed on an engine. Usually, each maintenance event involves multiple actions. Information documented includes the circumstances of the event, actions taken, parts installed and removed and any other findings they observed. The engineers who created these documents are located in numerous sites around the world, in the manufacture's or third party repair bases or at airports. As a result, the maintenance documents are in a large variety of formats. Some of these reports are in the form of unstructured Word documents, with different sites using different templates. Other engineers record maintenance information using a centralized Service Data Manager (SDM).

To enable machines to interpret meanings stored within the documents, we need an ontology that captures all the terms and concepts used. Moreover, since the document repository is to be used by both design and service engineers, the ontology should capture concepts from engineers working in both areas. The ontology is created by analyzing existing documents and conducting knowledge acquisition interviews with engineers [26]. The engineers interviewed are carefully selected and are domain experts from several different specialization. During these interviews, the *card sort* technique [22] is used to help the engineer show how they used different document types and the relationships between these documents. The result of these interviews enabled us to identify, by specialism, the main concepts and the associated keyword for these concepts used by the particular type of engineer when searching for information.

The resulting ontology contains concepts ranging from engine failure mechanisms, engine models and parts to airport

locations. Figure 6 shows the simplified RDF graph describing a maintenance event using the application ontology. In the diagram, the eclliptical objects are concepts, or *classes*, in the ontology. In addition, *Component* contains a taxonomy of aero-engine parts, and *Engine* includes a list of exiting engine types. Each of these classes has one of more properties, as indicated by the edges in the graph. The rectangles represent values, or *literals*, in the RDF graph. In total, the ontology has 896 classes and 133 properties. Most of the classes represent parts within the engine taxonomy.

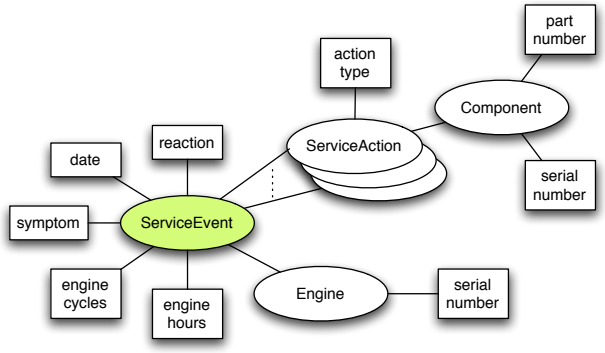

Figure 6: RDF graph for describing maintenance events. This is a simplified view and does not show all properties and classes defined in the ontology.

Using this ontology, we have populated 3store with maintenance records from both the SDM database and a small subset of reports provided as Word documents. Population from the SDM database is done using a script that maps the SDM database schema to the ontology. Population from Word documents is done manually, and is performed to test the validity of the ontology mapping. Automated extraction and population from Word documents is being investigated by one of our project partners that specializes in natural language processing. Currently, 3store contains approximately 3250 maintenance events, with around 31,000 number of actions. This information equates to 389,000 rows in a SQL database table. The space usage in the table is 12MB in data and 25MB in index (ie total of 37MB).

5.2 Web Services

Web services provide processing functionalities that can be access from anywhere on the network. The web services are decoupled from the portlets. In other words, multiple portlets can execute the same web service, and a single portlet can execute multiple web services. The interface of a web service is defined by its WSDL document. This document is in XML format, and lists the operations a web service supports. The document also defines the syntactic types of inputs and outputs for each operation.

We have developed several web services that perform common requests on the maintenance documents. For example, obtaining a list of parts that are involved in the highest number of unscheduled maintenance events, tracing the maintenance record of an engine or a part, or retrieving details of a maintenance event. The web services construct RDQL queries [23] according to their inputs. (RDQL is a query language for RDF data). They then obtain results from the triplestore using the constructed query. Thus, the web services allow RDQL queries to be reused across applications. Furthermore, they provide an abstraction over the data stored in the triplestore so developers who are unfamiliar with the ontology will still be able to perform queries. On the other hand, the use of an intermediary layer restricts the type of queries that can be performed. Therefore, for maximum flexibility, developers can access the triplestore directly by constructing their own RDQL query.

Besides web services that perform operations on the triplestore, we have also developed a graphing service that returns a histogram or a bar chart for a set of given data.

5.3 Web Portal

Users access our knowledge framework via the web portal. The portal uses username/password authentication, and role-based access control. Using role-based access control, we can customize the content of the portal according to the engineer's specialization. The roles defined in the system can reflect job functions of the engineers. Thus, engineers with different specialization can be served a different set of portal pages. For example, when studying deterioration mechanisms, support and operations engineers can be presented with information from individual engine parts, while design engineers will be presented with overall information for the entire fleet. This difference arises from the different tasks the engineers have to perform. Operations engineers are interested in locating and fixing problems of existing engines in service. On the other hand, design engineers are interested in overall performance to aid the design of future variants. Another possible customization can be used for navigation/drill down over the engine taxonomy. Since an engine contains a large number of parts, a simplified taxonomy for navigation will greatly speed up information browsing. The navigation taxonomy can be a subset of the entire engine taxonomy, based on the engineer's specialization.

A portal page consists of a series of presentation units, called portlets. Each portlet within the user interface performs a single presentation task. Figure 7 depicts the structure of the user interface for our intelligent document repository. Figure 8 shows a screenshot of the actual implementation.

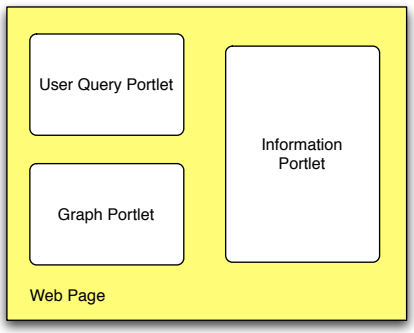

Figure 7: Portal-based user interface of the intelligent document repository.

The user query portlet presents the user with a set of canned queries. Each query is associated with a set of pre-

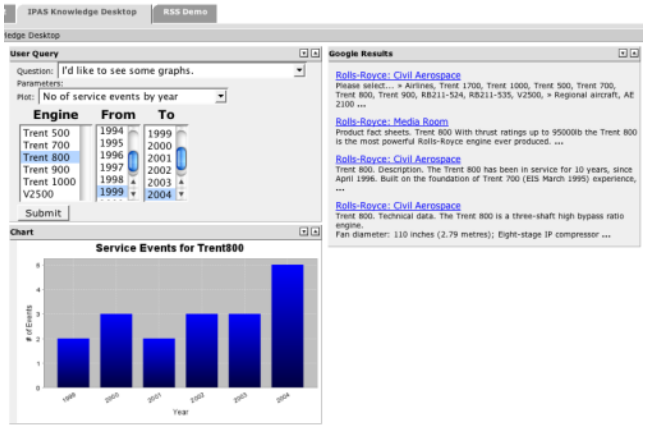

Figure 8: Screenshot from the portal implementation.

defined parameters that allows users to customize the query for their specific needs. The use of canned queries expedites the query process, and ensures users have entered all criteria needed to execute a successful query. They also provide a standardized perspectives on resources available.

The set of queries is derived from a questionnaire filled in by design engineers [14]. In the questionnaire, engineers are presented a list of questions relating to maintenance experience with a product. They are asked how often they might ask them when designing a new product. They are also asked what other questions they might want to ask. The result of this questionnaire tells us what are the most important and most common life cycle information design engineers seek from maintenance documents. The following are some of the questions that design engineers ranked highly:

1. What are the common failure mechanisms associated with this part?
2. What are the typical sequences of events which lead to this part's failure?
3. When this part fails, what other parts are typically affected as a result?
4. Which parts dominate the cost and reliability drivers in this engine?
5. What is the time between inspections/overhauls/repairs for this part?

After the user selects a new query and enters the appropriate parameters, the portal generates a new 'summary report' by propagating a change of viewing context to all visible portlets on the current page. This propagation of viewing context is achieved via *application scope* parameters in the Java Portlet API standard JSR 168. Each portlet decides how they will react to this change in the viewing context individually. If a change is needed, the portlet calls relevant web services to obtain information required for the updated view. For example, an engineer wants to know which parts dominate the maintenance statistics. To comply with this viewing context, the portal asks the appropriate web service

```
SELECT ?a
WHERE (?a <rdf:type> <i:ServiceActionPerformed>)
      (?a <i:hasActionPart> ?c)
      (?c <i:hasPartNumber> "ET10935")
```

Figure 9: RDQL query for obtaining all maintenance actions that involve the part ET10935.

```
SELECT ?a
WHERE (?a <rdf:type> <i:ServiceActionPerformed>)
      (?a <i:hasActionPart> ?c)
      (?c <rdf:type> <i:AirDuctJoint>)
```

Figure 10: RDQL query for obtaining all maintenance actions that involve air duct joints.

for a list of parts that is involved in the highest number of maintenance events. To generate this list, the web service has to consult two different information sources. The engine taxonomy and part names of a particular engine model are obtained from design documentation. Maintenance actions carried out through the engine's life are obtained from maintenance documentation. Within the maintenance documents, parts are only referred to using their part numbers, such as ET10935. Also, parts with different part numbers can be identical in functionality. Therefore, to generate results meaningful to the engineers, the web service must combine information from both design and maintenance. By correctly instantiating the parts involved in the *Component* hierarchy in the ontology, we can group functionally identical parts together within an RDQL query. As a result, by using the ontology, the web service can return a list of most serviced parts based on functionality groupings, and not simply distinct part numbers. This is illustrated in the query statements shown in Figure 9 and Figure 10. Using the information returned by the web service, the graph portlet redraws a new graph, and the information portlet shows the list of part names with links to design documents for those parts.

6. DISCUSSION

In this paper, we presented an intelligent document repository for an aero-engine manufacturer. However, the system architecture is generic and can be applied to organizations outside of aerospace engineering. Most large organizations generate a substantial amount of documents everyday. This is regardless of the sector the organization operates in, be it healthcare or finance. Our proposed architecture are applicable to document repositories in any industry, where the number of documents are too large to peruse, and that users are interested in knowledge extracted from documents deposited.

Heterogeneous document sources are integrated by a shared vocabulary – an ontology. To answer some of the questions from design engineers listed in Section 5.3, we need to combine information from multiple sources. For example, in Question 4, the engineer wants to know which parts dominate the cost drivers in the engine. To answer this, we combined reliability data from maintenance event reports with the engine taxonomy and costing information. These data are contained in different databases within the company.

Integration is made possible with the use of the ontology, which allows software agents to reason over the different resources.

Users access the document repository via a standard web browser. System components are hosted on distributed servers. As a result, the software can be deployed and updated centrally, without changing the configurations of thousands of desktop computers. Also, users can access the document repository without special software installed on their computers. The prototype has been demonstrated to Rolls-Royce engineers and has received positive reviews.

7. CONCLUSIONS

In this paper, we discusses the development of a knowledge-based document repository. The system allows users from across a company to access knowledge that they need to undertake their activities. In particular, we have developed a prototype for a large aero-engine manufacturer. A fundamental shift is occurring in the aerospace industry away from selling products to providing services. This shift in market focus means that new engines must be designed to provide lower and more predictable life cycle costs. To achieve this, engineers must obtain knowledge gained from the entire life of an engine.

However, design engineers often do not have access to front-line maintenance data due to the size and distributed nature of aerospace manufacturers. Moreover, an engine model remains in service for a long period of time and each engine in service generates a large number of maintenance documents during its life. As a result, it is impossible for design engineers to examine all information available thoroughly. Therefore, unlike existing document repository or digital libraries, our system is knowledge, and not information, based. It is based on services, and not document indexes or document linking. Services can search and analyze relevant maintenance records and design guidelines, and provides engineers with a 'summary report' to such information. It is hoped that this summary view provided will help engineers in creating design documents that incorporate aftermarket issues into the design requirements.

Two key technologies are used in integrating the distributed and heterogeneous data sources – the Semantic Web and Service-Oriented Architecture. The Semantic Web provides the framework allowing computer programs to interpret and reason over the heterogeneous document sources. The documents are integrated using an ontology, which captures the terms and concepts used in aerospace engineering. Service-Oriented Architecture allows knowledge extraction and analysis functionalities to be added to the system as modules called web services.

To test our proposed architecture, we have developed a working prototype. The prototype consists of a web portal, several web services and a RDF triplestore. Users access the document repository via a web portal, which is customized according to users' roles. The web services perform processing functionalities for queries we found engineers wanted to seek most from maintenance documents. The RDF triplestore contains real maintenance documents and the OWL aerospace ontology.

8. ACKNOWLEDGMENTS

This research was undertaken as part of the IPAS project (DTI Grant `TP/2/IC/6/I/10292`). The authors would also like to thank the project partners for providing us with data and ontologies. Specifically, we would like to thank Derek Sleeman and David Fowler from Aberdeen AKT for the ontology, and Alymer Johnson and Santosh Jagtap from Cambridge EDC for the user requirement analysis.

9. REFERENCES

[1] G. Boy and Y. Barnard. Knowledge management in the design of safety-critical systems. In D. Schwartz, editor, *Encyclopedia of Knowledge Management*. Idea Group, USA, 2005.

[2] L. Carr, W. Hall, S. Bechhofer, and C. Goble. Conceptual linking: Ontology-based open hypermedia. In *Proceedings of 10th International World Wide Web Conference (WWW)*, pages 334–342, Hong Kong, 2001.

[3] L. Carr, T. Miles-Board, G. Wills, and S. Grange. Extending the role of digital library: Computer support for creating articles. In *Proceedings of The 15th ACM Conference on Hypertext and Hypermedia*, pages 12–21, Santa Cruz, USA, 2004.

[4] L. Carr, T. Miles-Board, A. Woukeu, G. Wills, and W. Hall. The case for explicit knowledge in documents. In *Proceedings of ACM Symposium on Document Engineering*, pages 90–98, Milwaukee, Wisconsin, 2004.

[5] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web services description language (WSDL) 1.1. Technical report, W3C Note, `http://www.w3.org/TR/wsdl`, Mar. 2001.

[6] F. Ciravegna and Y. Wilks. Designing adaptive information extraction for the semantic web in amilcare. In S. Handschuh and S. Staab, editors, *Annotation for the Semantic Web*. IOS Press, 2003.

[7] R. Crowder, R. Bracewell, G. Hughes, M. Kerr, D. Knott, M. Moss, C. Clegg, W. Hall, K. Wallace, and P. Waterson. A future vision for the engineering design environment: A future sociotechnical scenario. In A. Folkeson, K. Gralen, M. Norell, and U. Sellgren, editors, *Proceedings of 14th International Conference on Engineering Design*, pages 249–250, Stockholm, 2003.

[8] R. Crowder, G. Hughes, and W. Hall. An agent based approach to finding expertise in the engineering design environment. In A. Folkeson, K. Gralen, M. Norell, and U. Sellgren, editors, *Proceedings of 14th International Conference on Engineering Design*, Stockholm, 2003.

[9] Y. Gao, I. Zeid, and T. Bardasz. Characteristics of an effective design plan system to support reuse in case-based mechanical design. *Knowledge-Based Systems Knowledge-Based Systems Knowledge-Based Systems*, 10(6):337–350, Apr. 1998.

[10] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[11] S. Harris and N. Gibbins. 3store: Efficient bulk RDF storage. In *Proceedings of 1st International Workshop*

on Practical and Scalable Semantic Systems (PSSS'03), pages 1–15, Sanibel Island, Florida, 2003.

[12] A. Harrison. Design for service – harmonising product design with a services strategy. In *Proceedings of GT2006, ASME Turbo Expo 2006: Power for Land, Sea and Air*, Barcelona, Spain, May 2006.

[13] I. Heath, G. Wills, R. Crowder, , W. Hall, and J. Ballantyne. Towards a new authoring methodology for large-scale hypermedia applications. *Multimedia Tools and Applications*, 12(2-3):129–144, Nov. 2000.

[14] S. Jagtap, A. Johnson, M. Aurisicchio, and K. Wallace. Pilot empirical study: Interviews with product designers and service engineers. Technical Report 140 CUED/C-EDC/TR140- March 2006, Engineering Design Centre, University of Cambridge, 2006.

[15] D. V. Khadilkar and L. A. Stauffer. An experimental evaluation of design information reuse during conceptual design. *Journal of Engineering Design*, 7(4):331–339, 1996.

[16] K. Malcolm, S. Poltrock, and D. Schuler. Industrial strength hypermedia: Requirements for a large engineering enterprise. In *Proceedings Hypertext '91*, pages 13–24, Seattle, 1991.

[17] F. Manola and E. Miller. RDF primer. Technical report, W3C Recommendation, `http://www.w3.org/TR/rdf-primer`, Feb. 2004.

[18] D. L. McGuinness and F. van Harmelen. OWL web ontology language overview. Technical report, W3C Recommendation, `http://www.w3.org/TR/owl-features`, Feb. 2004.

[19] D. E. Millard, F. Tao, K. Doody, A. Woukeu, and H. C. Davis. The knowledge life cycle for e-learning. *International Journal of Continuing Engineering Education and Lifelong Learning*, 16:110–121, 2006.

[20] N. Nnadi and M. Bieber. Lightweight integration of documents and services. In *DocEng '04: Proceedings of the 2004 ACM symposium on Document engineering*, pages 51–53, New York, NY, USA, 2004. ACM Press.

[21] T. Priebe and G. Pernul. Towards integrative enterprise knowledge portals. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 216–223, New York, NY, USA, 2003. ACM Press.

[22] G. Rugg and P. McGeorge. The sorting techniques: A tutorial paper on card sorts, picture sorts and item sorts. *Expert Systems*, 22(3):94–107, 2005.

[23] A. Seaborne. RDQL - a query language for RDF. Technical report, W3C Member Submission, `http://www.w3.org/Submission/RDQL`, Jan. 2004.

[24] N. Shadbolt and N. Milton. From knowledge engineering to knowledge management. *British Journal of Management*, 10(4):309–322, Dec. 1999.

[25] K. M. Wallace, C. Clegg, and A. Keane. Visions for engineering design: a multi-disciplinary perspective. In *Proceedings of 13th International Conference on Engineering Design*, pages 107–114, Glasgow, Scotland, 2001.

[26] G. Wills, D. Fowler, D. Sleeman, R. Crowder, S. Kampa, L. Carr, and D. Knott. Issues in moving to a semantic web for a large corporation. In *Proceedings of 5th International Conference on Practical Aspects of Knowledge Management (PAKM)*, volume 3336 of *Lecture Notes in Artificial Intelligence*, pages 378–388. Springer, 2004.