INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

INFORMING PRELIMINARY DESIGN BY INCORPORATING SERVICE KNOWLEDGE

Sylvia C Wong, Richard M Crowder, Gary B Wills and Nigel R Shadbolt

School of Electronics and Computer Science, University of Southampton, Southampton, UK

ABSTRACT

As manufacturers shift their focus from selling products to providing services, designers must increasingly consider the life-cycle requirements, in addition to conventional design parameters. To identify possible areas of concern, engineers must consider knowledge gained through the life cycle of a related product. However, because of the size and distributed nature of a company's operation, engineers often do not have access to front-line maintenance data. Additionally, the large number of documents generated during the design and operation of a product makes it impractical to manually review all documents thoroughly during a design task. This paper presents a prototype knowledge-based document repository for an aeroengine manufacturer; the aim of which is to aid engineers to create design requirements that incorporate maintenance issues.

Keywords: Intelligent documents, Semantic Web, Service-Orientated Architecture, Design

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 an aeroengine used on the current generation of aircraft. A fundamental shift is currently occurring in the aeroengine industry away from selling products to providing services – commonly termed *power by the hour*. Companies such as Rolls-Royce aim to make half its engine fleet subject to long-term maintenance service agreements by 2010, [1]. 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*, [1]. To ensure that the commercial risk to the manufacturer is minimised, any new engines must be designed to provide lower and more predictable maintenance costs. To minimize maintenance costs through out an engine's life cycle, design engineers must obtain knowledge gained from maintenance histories of similar products during an engine's design phase. This will help engineers identify engine parts most likely to be problematic throughout the engine's life cycle. As engine design is typically undertaken by a number of teams, who are responsible for individual engine modules, e.g. compressor, turbines, it is impossible for any single member of a design team to access more than a fraction of the available documentation. As is widely recognized, information systems usually develop over time into a set of heterogeneous resources. Moreover these document resources usually incorporate ill-defined metadata, which makes it difficult to find relevant information. As a result, it becomes difficult for engineers to follow a trail through the resources [2]. 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 new culture. The intelligent document repository searches and analyses relevant maintenance records and design guidelines, and provides design engineers with easy access to such information. It is expected 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.

The objective of our work can be summarized as follows: to feedback and harvest knowledge gained from the maintenance and related documents to help the operations engineers in designing modifications to existing engines, and the design engineers in designing the next engine variant.

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

2 KNOWLEDGE REUSE IN 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 [3], while during a redesign activity up to 70% of the information is taken from previous solutions [4]. A cursory consideration of these figures identified two immediate challenges - how to capture knowledge, and how to retrieve it. An intelligent document repository that can be used within a manufacturing organization for the retrieval of knowledge from across the organization to support design activities will have a significant impact on the design process.

Figure 1 shows the key information flow for the different stages in the life of an 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 determine the approximate dimensions, weight, power and other physical characteristics for the proposed engine. 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, where engines are built. After delivery, the maintenance of an engine is undertaken by the airline or a third party, within the requirements of a type certificate. The type certificate requires the manufacture to provide the maintainer with instructions and parts, so that the engine can be maintained in an airworthy condition. In addition the maintainer has to prove to the certificating authorities that they are capable of maintaining to the required standards. To support the maintenance activity the manufacturer's provide technical support and operation teams. 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 analyzed 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 [5].

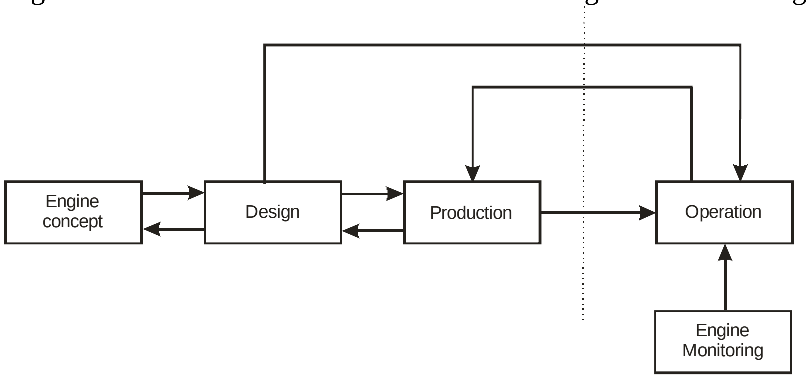

Figure 1. Information flow between the different stages in the life of an engine. The vertical line between production and operation represents the transfer of the engine from its manufacturer to its user.

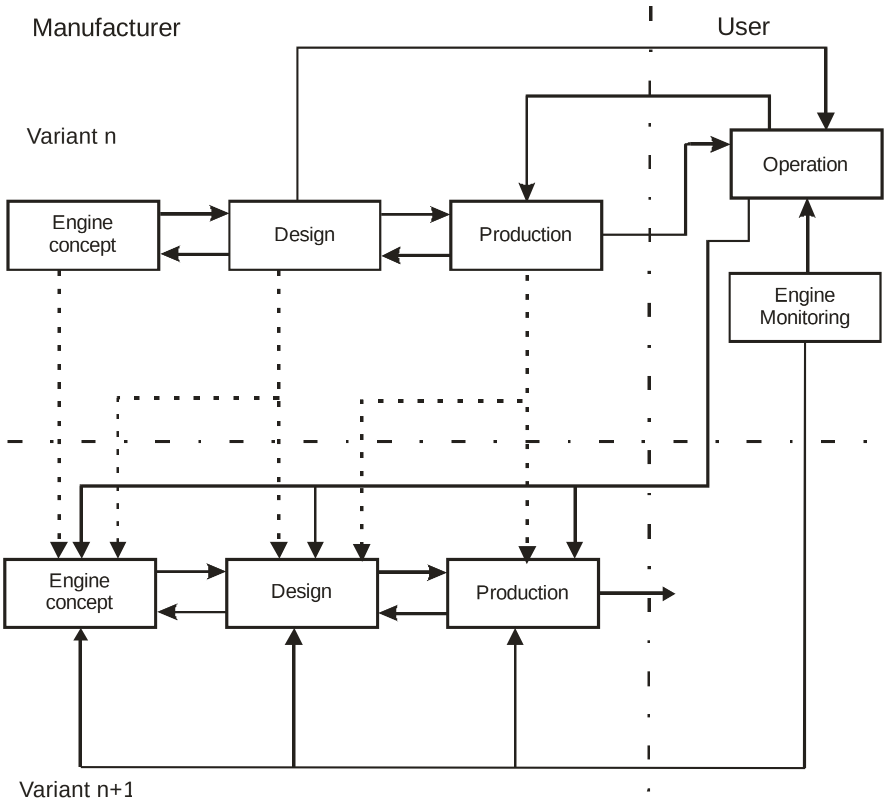

Figure 2. We aim to facilitate the flow of information gained during the life-cycle of one engine variant to inform the design of the next variant. The dotted arrows indicate the flow of design rationale and related knowledge. The solid arrows represent all other information flows including design documentation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

While this process works very well, it does have significant 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, 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.

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 that 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 are responsible for populating the document repository with maintenance reports and related documents.

During the pre-flight checks, a flight crew reported a problem with an engine's bleed air system. Inspection of the engine revealed that a duct had failed at a joint due to vibration. After repair, the aircraft was returned to service, and a full maintenance event report submitted to the document repository.

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

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

> *Following a review of the maintenance events relating to a specific engine fleet, a trend was noticed in the higher than expected number of failure of a duct joint due to 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 variant of the engine:

> *The design team for the next variant of this engine reviews the performance of the air bleed system across all engine fleets to learn from previous design rationale and operational histories. The large numbers of failures were noted and detailed 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 40% more than the original. However, the saving over the life of the engine will be substantial due to lower likelihood of in-service failure and consequential impact on operations.*

With the concepts associated with the Semantic Web it is possible to enhance still further the range of information brought into the design process. It is for example possible to review how a design teams options of a specific approach to a solution has changed over a number of years. As metadata is present to a documents' section level, a designer will be able to locate reports delivered by other design teams that contain references to the original design team's documentation, or to the parts designed by that team.
As part of this search it may becomes apparent that some reports originated from outside the company's engineering domain, in particular purchasing, this will allow the designer to get a complete overview of both the commercial and technical decisions taken during the design and development of the air system. As will be expected the search will return many more 'hits' than a person can reasonably deal with. To aid the review process the designer will be able to visualize these relationships in two or three dimensions through the Semantic Web browser, which permits easy navigation through the information space. This not only allows understanding of these relationships and how they've influenced with engineering developments, but also identifies events clusters.

3 KNOWLEDGE ENGINEERING

Organizations have become increasingly concerned with knowledge management [6], amassing large amount of information into their corporate memory [7]. The aim is to use this information repository 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.

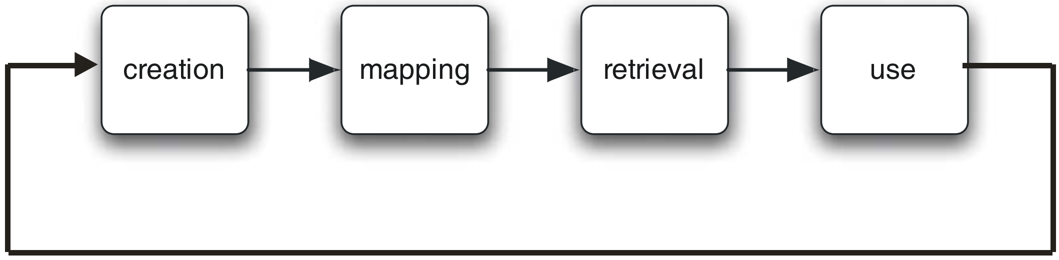

Figure 3. The four key activities in the knowledge management cycles.

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* [8]. 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 transforms *information* into *knowledge*. We believe that knowledge is relevant information delivered at the right time and context [9]. 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 [10], 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 [11]. 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.

The approach to aeroengine provision discussed in [1] has considerable similarities to the “Total Care” concept described in [12], which identifies benefits to both the manufacturer and the customer. The design process discussed in the paper requires the integration of both operational and service information to maximise the benefits to both parties.

In [2], 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 prefer 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 [13]. 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.

Our work in [2] 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 [14] 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 [15] [16] [17]. 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 [16]. 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*.

Finally, there are also projects working on creating new semantically marked up data for large knowledge repositories, such as [18] and [19]. 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, termed 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.

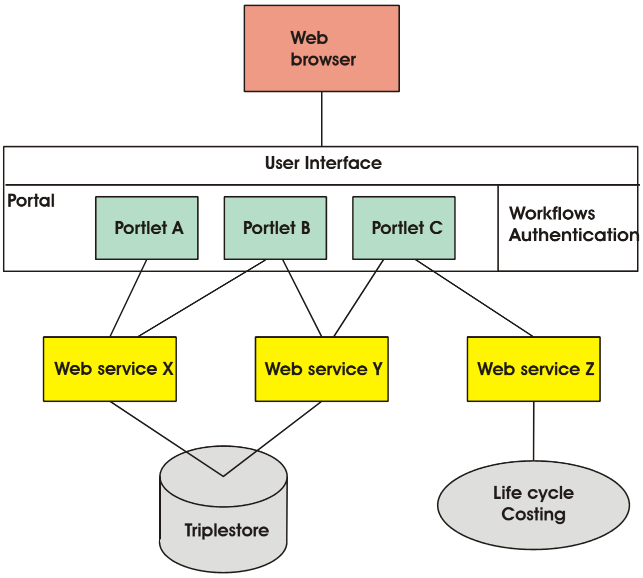

Figure 4. A conceptual view of the prototype's architecture, showing three typical web services, two for search and retrieval, and the third is the interfacing to a life cycle cost modeler.

Our intelligent document repository is organized as a SOA, as shown in Figure 4. The user forms a query via a user interface. In the current system design, this user interface is in the form of web pages 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 [20]. 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 information set. OWL [21] 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. Existing web standards are used wherever possible, to maximize tool reuse, compatibility and portability. A web browser accesses 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[2] . 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, which are important elements in the design concept being introduces. The web service interfaces are defined in WSDL [22], 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 Sesame[3], an open source RDF framework with support for RDF Schema inferencing and querying. It should be noted that the architecture does not require all web services to access the underlying information sources via the triplestore. However, most, if not all, web services will operate on the semantically enabled data within the triplestore instead of the original heterogeneous documents.

5.1 Documents and Ontology

As expected in this domain, there is a considerable number of information sources, included Event Reports (completed by front line maintenance engineers), and Strip Reports (created when an engine is subjected to a major overhaul). In addition the operation side of the company is supported by systems including Service Data management that hold a considerable number of event reports. In an engine type is modified in service, this is recorded in a Service Bulletin. This is complemented by information including PDM, Design Reports from the manufacturing side of the organisation.

To enable machines to interpret meanings stored within these documents, we created 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 captures concepts from engineers working in both areas. 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, allowing the ontology to be created [2].

The resulting ontology contains concepts ranging from engine deterioration mechanisms, engine models and parts, to airports. Figure 4 shows a simplified RDF graph for the concepts associated with maintenance events in the application ontology. In the diagram, the ellipses are concepts, or *classes*, in the ontology. In the ontology, a UML attribute is modelled as an OWL property which has the specified class as its domain. A UML association is modelled with an OWL property which has the linked classes as its domain and range. In addition, the *Part* class contains a taxonomy of aero-engine

[2] For further details see http://jcp.org/jsr/detail/168.jsp
[3] For further details see http://www.openrdf.org/about.jsp

parts, and the *Engine* class includes a list of exiting engine types. In total, the ontology has 900 classes and 150 properties.

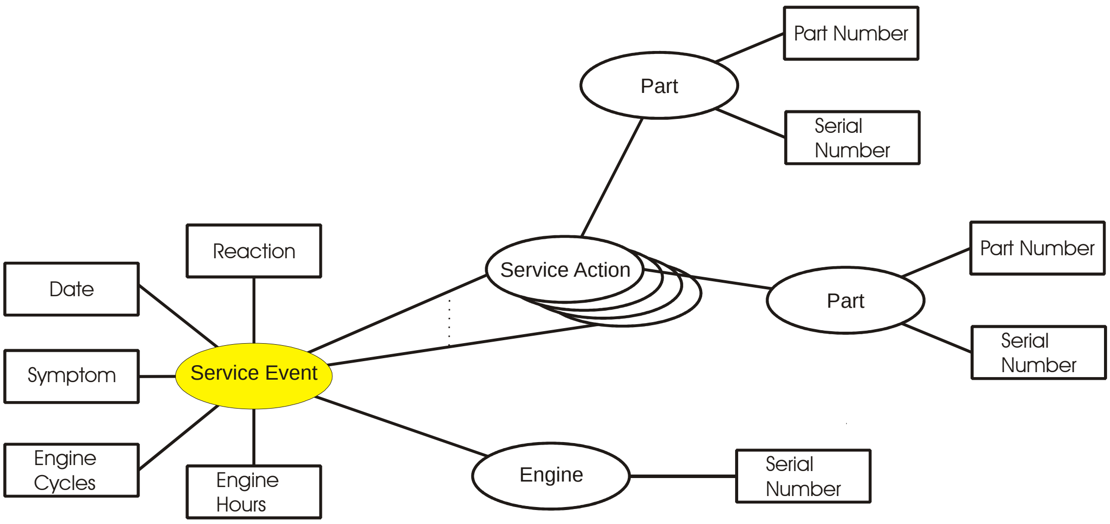

Figure 4. The RDF graph used to describe maintenance events. This is a simplified view and does not show all properties and classes defined in the ontology.

Using the developed ontology, we have populated a triple store with maintenance records extracted from the company's SDM database, together with a small subset of the strip and event reports which were provided as Word documents. Population from the SDM database was achieved using a script that maps the SDM database schema to the ontology. Population from Word documents is currently undertaken 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 specialize in natural language processing. Currently, the triplestore contains approximately 3300 maintenance events, which equates to around 31,000 actions. The populated triplestore contains 316 000 ground triples and 213 000 inferred triples generated through reasoning. The current memory requirements are 12MB for data and 25MB for the index.

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 SPARQL queries [23] according to their inputs. They then obtain results from the triplestore using the constructed query. Thus, the web services allow SPARQL 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 SPARQL query.

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 information across the entire fleet. These different requirements 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 is the use of navigation/drill down over the engine taxonomy. Since an engine contains a very 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.

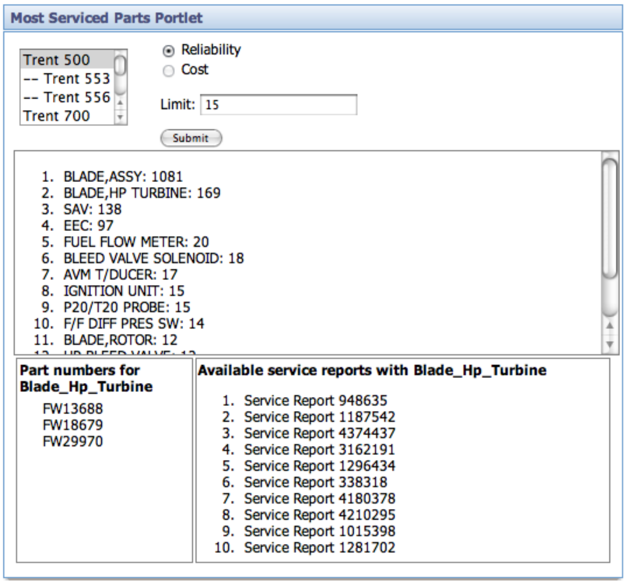

Figure 5 Screenshot from the current portal implementation.

A questionnaire based study was carried out to better understand what kind of knowledge design engineers would like to discover from the document repository [24]. In the questionnaire, engineers were presented with 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. Typical questions that engineers ranked highly included:

1. What are the common deterioration mechanisms associated with this part?
2. What are the typical sequences of events which lead to this part's deterioration?
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?

Figure 5 shows a screenshot of one of the portlets we developed. This portlet shows engineers the list of parts that are involved in the highest number of service actions in the selected engine. Users can obtain further information about any of the parts in the list. For example, they can see what part numbers make up a functional part. They can also view service reports that mention a part. To generate the list of most serviced parts, 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, 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 *Part* hierarchy in the ontology, we can group functionally identical parts together within a 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.

It should be noted five questions can be broadly classified as belonging to two categories – ranking and scoring. To answer questions 1 to 3, we need to first find the occurrence of a pattern within the documents. Afterwards, we rank the occurrences to obtain a list in ascending or descending order. Specifically, in a SPARQL query, the pattern is a RDF triple, or a series of RDF triples. Hence questions 1 to 3 can be considered to be a ranking problem. Question 4 contains two sub-questions, sorting parts that are serviced most by frequency and by cost. Obtaining a list of parts by service frequency is a ranking problem. Sorting by cost requires the multiplication of unit costs of parts with their service frequencies. Therefore this question can be considered to be a scoring problem in addition of being a ranking one. For question 5, criticality is measured by the reaction taken by front-line maintenance engineers. Each reaction is assigned a criticality score, as defined by the manufacturer. Therefore, question 5 is also a scoring problem. One feature that is clear from the questions that design engineers need answering are more likely to require a statistical approach as opposed to extracting a highly specific document.

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 is applicable to document repositories in any industry, where the number of documents is 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 design engineers and while it has limited functionality the approach taken has received positive reviews. The design community recognizes the need for this approach as we are able to summarize large document collections (both in size and type). The customisation of the portal is highly relevant to both the user and the company, as it bring clear benefits. The prototype is in the process of being extended to incorporate further documents collects, allowing more summarization and questions to be addressed.

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.

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 Professor Derek Sleeman and Dr David Fowler from Aberdeen AKT for the ontology, and Dr Alymer Johnson and Santosh Jagtap from Cambridge EDC for the user requirement analysis.

REFERENCES

[1] 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, 2006.

[2] 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)*, 2004, pp. 378-388.

[3] 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,* vol. 10, pp. 337-350, April 1998.

[4] D. V. Khadilkar and L. A. Stauffer, "An Experimental Evaluation of Design Information Reuse During Conceptual Design," *Journal of Engineering Design,* vol. 7, pp. 331-339, 1996.

[5] T. Jackson, J. Austin, M. Jessop, B. Linag, A. Pasley, M. Ong, G. Allan, V. Kadirkamanathan, H. Thompson, and P. Fleming, "Distributed Health Monitoring for Aero-Engines on the Grid: DAME," in *Proceedings of IEEE Aerospace*, Montana, 2005.

[6] N. Shadbolt and N. Milton, "From Knowledge Engineering to Knowledge Management," *British Journal of Management,* vol. 10, pp. 309-322, December 1999.

[7] I. Heath, G. Wills, R. Crowder, a. W. Hall, and J. Ballantyne, "Towards A New Authoring Methodology for Large-Scale Hypermedia Applications," *Multimedia Tools and Applications,* vol. 12, pp. 129-144, November 2000.

[8] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition*, vol. 5, pp. 199-220, 1993.

[9] 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,* vol. 16, pp. 110-121, 2006.

[10] 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 *Proceedings of 14th International Conference on Engineering Design*, Stockholm, 2003, pp. 249-250.

[11] 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*, Glasgow, Scotland, 2001, pp. 107-114.

[12] T. Alonso-Rasgado and G. Thompson, "A rapid design process for Total Care Product creation," *Journal of Engineering Design,* vol. 18, 2007.

[13] A. L. Cohen, P. P. Maglio, and R. Barett, "The expertise browser: How to leverage distributed organizational knowledge.," in *Workshop on collaberative and cooperative*

information seeking in digital information environment at the International Conference on Computer Supported Cooperative Working Seattle, WA, 1998.

[14] T. Priebe and G. Pernul, "Towards integrative enterprise knowledge portals," in *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, New Orleans, LA, USA, 2003, pp. 216-223.

[15] 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)*, Hong Kong, 2001, pp. 334-342.

[16] N. Nnadi and M. Bieber, "Lightweight integration of documents and services," in *DocEng '04: Proceedings of the 2004 ACM symposium on Document engineering*, Milwaukee, Wisconsin, USA, 2004, pp. 51-53.

[17] K. Malcolm, S. Poltrock, and D. Schuler, "Industrial Strength Hypermedia: Requirements for a Large Engineering Enterprise," in *Proceedings Hypertext '91*, Seattle, 1991, pp. 13-24.

[18] 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*, Milwaukee, Wisconsin, 2004, pp. 90-98.

[19] F. Ciravegna and Y. Wilks, "Designing Adaptive Information Extraction for the Semantic Web in Amilcare," in *Annotation for the Semantic Web*, 2003.

[20] F. Manola and E. Miller, "RDF Primer," W3C Recommendation, http://www.w3.org/TR/rdf-primer February 2004.

[21] D. L. McGuinness and F. v. Harmelen, "OWL Web Ontology Language Overview," W3C Recommendation, http://www.w3.org/TR/owl-features February 2004.

[22] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL) 1.1," W3C Note, http://www.w3.org/TR/wsdl March 2001.

[23] A. Seaborne, "RDQL - A Query Language for RDF," W3C Member Submission, http://www.w3.org/Submission/RDQL January 2004.

[24] S. Jagtap, A. Johnson, M. Aurisicchio, and K. Wallace, "Pilot Empirical Study: Interviews with Product Designers and Service Engineers," Engineering Design Centre, University of Cambridge 2006.

Contact:

Dr Richard Crowder
School of Electronics and Computer Science
University of Southampton
Southampton
SO17 1BJ
UK
Phone: +44 23 8059 3441
Fax:
rmc@ecs.soton.ac.uk.