Creating and Using Organisational Semantic Webs in Large Networked Organisations

Ravish Bhagdev, Ajay Chakravarthy, Sam Chapman, Fabio Ciravegna and Vita Lanfranchi

Department of Computer Science, University of Sheffield, Regent Court,
211 Portobello Street, S1 4DP Sheffield, United Kingdom
{*N.Surname*}@shef.ac.uk

Abstract. Modern knowledge management is based on the orchestration of dynamic communities that acquire and share knowledge according to customized schemas. However, while independence of ontological views is favoured, these communities must also be able to share their knowledge with the rest of the organization. In this paper we introduce K-Forms and K-Search, a suite of Semantic Web tools for supporting distributed and networked knowledge acquisition, capturing, retrieval and sharing. They enable communities of users to define their own domain views in an intuitive way (automatically translated into formal ontologies) and capture and share knowledge according to them. The tools favour reuse of existing ontologies; reuse creates as side effect a network of (partially) interconnected ontologies that form the basis for knowledge exchange among communities. The suite is under release to support knowledge capture, retrieval and sharing in a large jet engine company.

Keywords: Semantic Web-based Knowledge Management, Knowledge capture, search and retrieval, application of semantic web technologies.

1 Introduction

The classic Knowledge Management environments aim at creating large homogeneous knowledge or document repositories where corporate knowledge is collected and organised according to a single conceptual schema. This schema represents the official agreed view of the organisation with the intent of supporting communication between its different parts and is generally used in an Enterprise Knowledge Portal providing unique standardised access to proprietary knowledge [1]. However, it is a well-known issue that many of these portals are deserted by users, who continue to capture and share knowledge in ways quite different from those provided by the corporate-wide systems. For instance workers use non-official tools such as shared directories, personalised and local databases, etc. [2]. The reason for deserting the central knowledge portals can be summarised as the difficulty in adopting models, schemas and procedures that are unsuitable to specific communities of users. From experience in the manufacturing industry, users tend to organise their cycles of knowledge acquisition and capturing around unstructured (e.g. textual documents) or semi-structured documents (e.g. forms) created with tools like word processors or spreadsheets, rather than using centralized (or even local) databases. Usually this type of document is retrievable via keyword searching and knowledge is individually acquired by reading the document. This problem is amplified by the large amount of unstructured material produced in large organisations, which increases the unmanageability issue. Also, modern organisations are very dynamic and often favour the creation of communities of practice, i.e. more or less formal groups of people with a common interest in some subjects and who collaborate over an extended period to share ideas, mental models, practices, find solutions, and build innovations [4]. As these communities tend to exist across the traditional organisational boundaries, they ill fit pre-determined standard schemas and require flexible customisable knowledge for specific ad-hoc uses.

In summary, modern knowledge management is moving away from the idea of a large centralised schema to suit all situations (*absolute knowledge*) to a more localised approach (*local knowledge*) that "is a partial, approximate, perspectival interpretation of the world" [2].

A similar approach has been adopted by the Semantic Web community, where the concept of small scale distributed interconnected ontologies [3] has replaced the idea of large comprehensive all-encompassing ontologies. The Semantic Web can therefore help such a change in knowledge management with tools and techniques supporting: 1) definition of community-specific views of the world; 2) capture and acquisition of knowledge according to them; 3) integration of captured knowledge with the rest of the organisation's knowledge; 4) sharing of knowledge across communities.

In this paper we introduce K-Forms and K-Search, a suite of tools supporting distributed networked knowledge acquisition, capture, retrieval and sharing. They enable communities of users within or across organisations to define their own views, while at the same time maintaining connections with other communities' views and (if required) with a central schema.

K-Search enables retrieval and sharing of documents and knowledge. It enables accessing multiple repositories (semantic or traditional) using either directly their local reference ontology or other connected ontologies. Queries can return 1) the original documents annotated with the extracted knowledge, or 2) documents generated using metadata for specific user needs, or 3) a summary of the knowledge in the form of graphs or statistics or 4) triples for further elaboration (e.g. by an external Semantic Web Service). Furthermore extracted triples can be exported in CSV or RDF formats for further analysis using external tools.

K-Forms and K-Search enable management modalities that satisfy two main principles claimed by [2] as essential for modern knowledge management:

- *Principle of Autonomy* where each unit is granted a high degree of autonomy to manage their local knowledge;
- *Principle of coordination* where units are enabled to exchange knowledge with other units through a mechanism of mapping other units' context onto their local context.

New communities can be supported by rapidly defining schemas and modalities to capture and share knowledge. As they tend to evolve rapidly, change is supported by enabling easy networked modification of the knowledge schema via a standard browser. Moreover this approach supports the definition and reuse of different views on the same data, particularly useful when communities of users with different tasks and information needs are involved.

In this paper we first introduce a general architecture from a functional point of view, describing how knowledge can be represented, captured, retrieved, shared and visualised using K-Forms and K-Search. Then we will go into the details of system components. An evaluation will show how and in what sense the methodology is appreciated and understood by real users. We will then describe an application developed for an industrial environment to support capture and sharing of knowledge about jet engines. A comparison with the state of the art will follow. Finally we draw conclusions and highlight future work.

2 A Form-oriented approach to Knowledge Acquisition and Capture

The architecture is composed of two modules: K-Forms and K-Search. K-Forms supports knowledge acquisition and representation (the definition of the knowledge structures, e.g. the ontology) as well as capture (the creation of instances). K-Search enables searching and sharing of information and knowledge.

Both tools use the concept of forms as the interaction paradigm. Form-based solutions are very familiar to users, as they are very common in knowledge management and in every-day life, therefore they are very easy to understand and use.

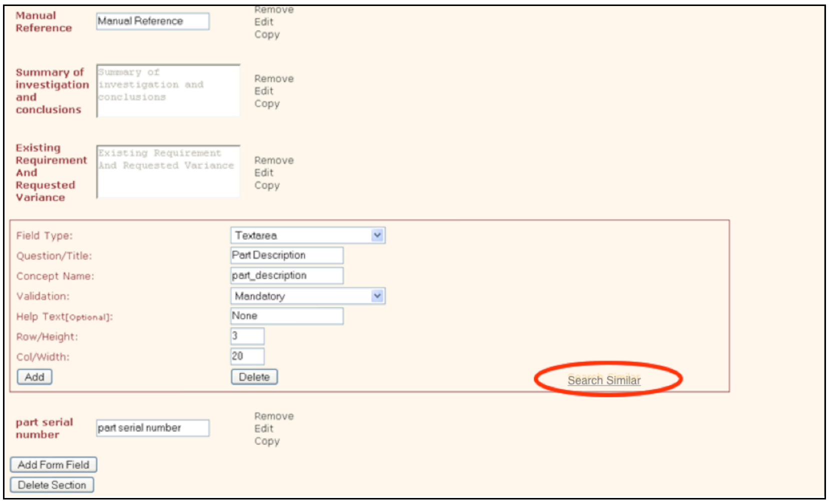
2.1 K-Forms: the Final User View

For knowledge acquisition, many organisations adopt forms as a way to capture knowledge, either in the form of static Web forms or as a template in a word processor (e.g. MS Word) or in a spreadsheet. Both solutions have issues.

Static forms are largely unsuitable to dynamic emerging communities as they require a computer expert to be implemented or modified and such expert is rarely available in a community of practice. Therefore Web forms tend to be used mainly in centralised services and changes to the forms are unlikely to happen. This is a serious drawback as it has been noted how users who do not find the appropriate field in a form, start using inappropriate fields for storing the information they need [7], hence causing the consistency of data to be corrupted.

The use of flexible forms (e.g. MS Word) has the opposite disadvantage. Users feel entitled to modify the forms at any time, especially when a centralised consistency check is difficult or impossible. In this case, the change in format causes missing data (e.g. because fields disappear) and/or the insertion of inappropriate data (because these tools do not allow rigid type control on the input). Also the information contained is unstructured and therefore it requires capturing using other means like IE from Text. The extraction is made even more complex when the forms are modified.

As noted in [8], in a corpus of 18,000 reports on jet engine diagnostic issues, forms were so disrupted that the application of complex machine learning techniques was required in order to normalise and capture knowledge.



Manual Reference: Manual Reference

Summary of investigation and conclusions: Summary of investigation and conclusions

Existing Requirement And Requested Variance: Existing Requirement And Requested Variance

Field Type: Textarea

Question/Title: Part Description

Concept Name: part_description

Validation: Mandatory

Help Text[Optional]: None

Row/Height: 3

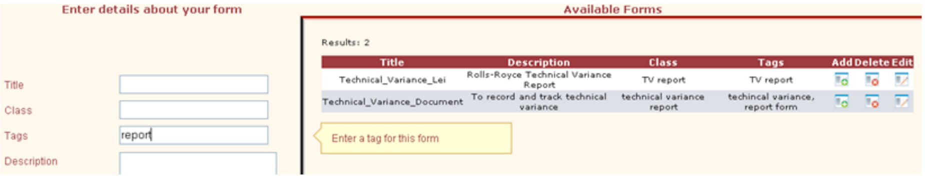
Col/Width: 20

part serial number: part serial number

Figure 1 - K-Forms interface for new form creation: highlighted the possibility to search for similar concepts when creating a new one

K-Forms is designed to provide an intermediate semantic solution between rigid web forms and unstructured documents: easily deployable Web forms. K-Forms supports knowledge acquisition, modelling and capture by enabling an easy creation and deployment of Web forms via a graphical user interface. K-Forms typically cannot be modified arbitrarily by whoever fills them but rights can be given to do so should it be appropriate. To encourage semantic interconnections, a functionality for identifying and reusing relevant templates is provided so that the definition of new forms can always start from existing ones (see Figure 1, Figure 2 and Figure 3). This is an important requirement because communities in the same organisation generally tend to need some minor modifications of a similar form within the same domain. For example, components of a jet engine are generally defined via *name*, *model* and *serial number*. A specific community investigating the condition of specific modules during engine overhaul will still need the same information, but will need to add some additional parts (e.g. the name of the module investigated and the engine's number of cycles, etc.). So importing from existing forms not only saves a users time avoiding the redefinition of existing structures but also provides schema linkage between forms and the captured concepts.

K-forms enables easy definition of confidentiality gates, distinguishing between local knowledge (which may be confidential and must not be necessarily shared with others) and shareable knowledge. Knowledge acquired for each domain is stored on a community server. Every user (including external ones) must access the community store in order to access the knowledge. Therefore each community is always in full control of the information and can decide what parts are made available to external users and what parts are not.



Enter details about your form

Title	
Class	
Tags	report
Description	

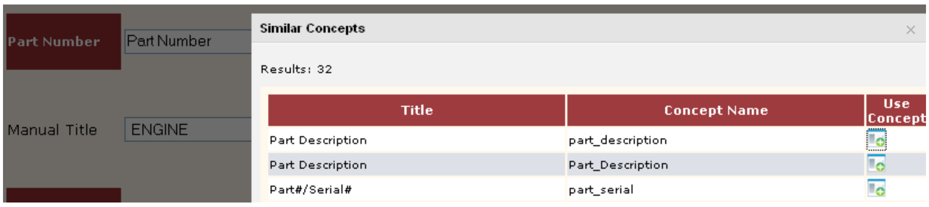
Available Forms

Results: 2

Title	Description	Class	Tags	Add	Delete	Edit
Technical_Variance_Lei	Rolls-Royce Technical Variance Report	TV report	TV report			
Technical_Variance_Document	To record and track technical variance	technical variance report	techincal variance, report form			

Enter a tag for this form

Figure 2 - K-Forms: search and reuse of similar <u>forms</u> before creating a new one



Part Number	Part Number
Manual Title	ENGINE

Similar Concepts

Results: 32

Title	Concept Name	Use Concept
Part Description	part_description	
Part Description	Part_Description	
Part#/Serial#	part_serial	

Figure 3 - K-Forms: search and reuse of similar <u>concepts</u> before creating new ones (here search for concepts similar to part_description is shown).

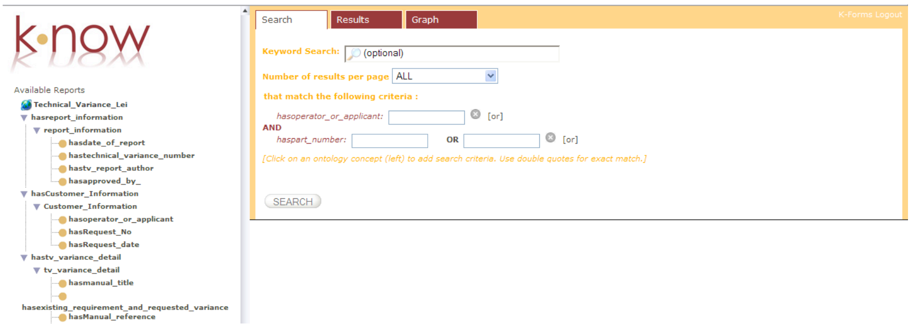
2.2 Forms: the Technical View

From a technical point of view, K-Forms is a methodology for knowledge acquisition and capture. It is based on semantic technologies (invisible to the users) that provide knowledge sharing and reuse capabilities beyond what is possible with standard technologies.

2.2.1 Forms as ontologies

When a new form template is created, K-Forms seamlessly translates it into an explicit OWL ontology (see Figure 4); the template concepts and fields and associated constraints are translated into OWL statements. Data inputted into forms based on those templates is transformed into RDF triples that can be searched using standard query languages such as SPARQL and SERQL.

Forms can be divided into sections and fields. Sections can have subsections and fields, e.g. the section designed to hold data about a person will be a concept; a person will have a series of properties e.g. name, address and date of birth. Sections are presented as sub-forms to be filled.
Fields are typed (e.g. text field, integer field, text area, checkbox, option list, etc.). Fields represent meta-properties of the document (e.g. author, date, etc.) or its content (e.g. an issue to be reported). Simple fields will require just the inputting of a value at filling time (a number, a string, a text, a date, etc.). When a field is defined, the user selects a field type and must input appropriate properties for the specific field (e.g. size, validation constraints, default value, help text, etc.). If parts of an existing form are imported, the system automatically imports any associated constraint (e.g. on data types).



Keyword Search: (optional)

Number of results per page ALL

that match the following criteria :

hasoperator_or_applicant: [or]

AND

haspart_number: OR [or]

[Click on an ontology concept (left) to add search criteria. Use double quotes for exact match.]

Figure 4 - The Ontology produced by K-Forms after a new form is created can be loaded in K-Search for searching the newly generated knowledge

When a form template is saved, K-Forms translates a template into an OWL Ontology The form is represented as an OWL class (`<FormName> Class`) with its own NameSpace to avoid name conflicts. Form sections are represented as OWL classes (`<ConceptName> Class`) that can have subsections (related classes) or individual fields (properties).

Relations among concepts are represented as OWL relations between classes and properties (using owl:equivalentClass and owl:equivalentProperty).

Adding relations between parents (section or entire form) and the contained subsections (object type properties) enables reuse of entire sections. Relational tables can be represented as advanced sections. The domain of some relations may be the overarching `<FormName> Class`. When concepts are introduced at the top level, a relation is formally created domain `<FormName> Class` and range `<ConceptName> Class`.

Individual fields (such as text box, text area, check box) can be added as a property of each section, subsection, or directly in form classes; they are represented as OWL properties. Some of them (as `FormCompilationDate`) are properties of the overarching `<FormName> Class`. Restrictions can be set for the possible values of the individual fields using xml datatype schema (xsd:types) (for example declaring a type to be a positive integer). Again individual fields can be reused between forms and classes.

When concepts are reused across forms, owl:equivalentClass is created. This is due to the fact that users may reuse the concept but change non-conceptual details regarding its visualisation.

2.2.2 Form reuse as implicit ontology mapping

When (part of) a form is reused, an explicit mapping between (parts of) the two underlying ontologies is established. This means that instead of creating new concepts into the new form namespace, the new ontology will use the concept in the other form's namespace. If the concept is modified, then a link between the two concepts is maintained by automatically creating a meta-class capturing the common parts. This may happen either at the single concept level or at the whole form level.

This creates automatically a network of interconnected ontologies.

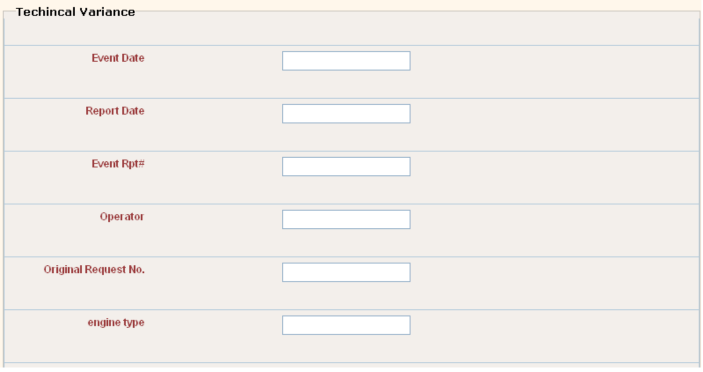
The ontology mapping can also be controlled by a system manager who can manually establish further mapping or suggest modifications to the forms in order to make it compatible to other existing forms if possible. This will increase the quality and quantity of interconnections.

Although these connections are far from being complete and ideal, they enable the creation of a web of interconnected ontologies where knowledge can be and shared, much in the way recent Semantic Web developments recommend [5][6].

2.2.3 Knowledge capture in K-Forms

When a form design is complete (after community validation, if needed), the form can be published for use via Web or Intranet (by simply pressing a "publish" button).

Knowledge capture using the form can then begin (Figure 5). All the inputted values are transformed into RDF statements related to the form ontology.



Techincal Variance

Event Date

Report Date

Event Rpt#

Operator

Original Request No.

engine type

Figure 5 - The form corresponding to the simple template in Fig. 1.

The target users are knowledge workers, for example an engineer diagnosing a fault on a car who has to report about the fault, its symptoms, causes, etc. Form filling is a natural approach to knowledge capture in such environments. The constraints posed on the form (e.g. via selected validation methods and strict data typing) ensure a degree of knowledge quality. Although forms can be modified easily, policies are set to prevent adjustments, as template alterations must be a community decisions and not individual decisions. After data entry, the system generates a document summarising the inputted information. The document can alternatively be sent by email or printed. The format of the summary can be customised with a simple HTML editor during form design. The user can preview the completed form and decide to modify or publish the knowledge. As knowledge capture can occur in places where a Web connection is unavailable, gathered knowledge can be held in a local store and later published to a central server.

2.3 K-Search: Sharing Knowledge Across Communities

Once the knowledge has been thus captured, it can be searched using **K-Search**, a semantic tool for documents and knowledge retrieval and sharing[8]. To guarantee consistency in the framework, K-Search has been designed following the same form-based interaction paradigm, providing an interface that allows users to combine conditions on both metadata and keywords within the same query. K-Search also supports multiple, de-centralised, dynamic knowledge communities by enabling access to knowledge stored in multiple repositories (e.g. triple stores) with multiple ontologies.

The ontology associated to a template is made available to a search mechanism (K-Search) that enables hybrid searches [8]. Hybrid Search (HS) combines the flexibility of keyword-based retrieval with the ontology and its reasoning capabilities, making a synergistic use of both strengths. In HS, users can combine within the same query: (i) ontology-based search; (ii) keyword-based search and (iii) keyword-in-context based search. Keyword-in-context searches for keywords only in the text annotated with a concept in the ontology; in case the document has been created with K-Forms, it searches the content of the form field values associated to the concept.

K-Search interface and modality of interaction is illustrated in Figure 3. The interface works in a standard Web browser, enabling the definition of complex hybrid queries in an intuitive way. Keywords can be inserted into a default form field in a way similar to what required by search engines; Boolean operators AND and OR can be used in their combination. Conditions on the metadata can be added to the query by clicking on the ontology graph (left side of interface in Fig. 3). This creates a form item to insert conditions on the specific concept. Here a unique identifier or a substring of the field can be inserted as condition. AND and OR can be inserted among ontological conditions.

The query output of K-Search is a set of ranked documents displayed as a list on the mid-right panel of the interface; each item in the list is identified by the values in the metadata that satisfy the ontology-based search. Clicking on one item causes the corresponding document to be shown on the bottom right. The document is presented with added annotations via colour highlighting; advanced features or services are associated to annotations [12, 13]: for example right clicking on a concept enables – among other things - query refinement with the selected term. Moreover, K-Search provides ways of inspecting the results of the query as bi-dimensional graphs (pie or bar chart) according to two elements e.g. issue and component); items in the graph (e.g. pie chart slices) are clickable to focus on the associated subset of data (e.g. associated documents or data). The retrieved triples can be exported if needed (in RDF, CSV or XLS formats) for external analysis or automated processing. K-Search is described in details in [8].

2.3.1 Querying across ontologies

The Web of interconnected ontologies is exploited in searching. K-Search enables searching multiple repositories at once using one of the available ontologies. This means that a user can decide to:

(i) Query a specific resource via the original ontology
(ii) Query a resource using a different ontology interconnected to the original one
(iii) Query multiple repositories using one specific ontology.

When an ontology different from the original is used (cases *ii* and *iii*), the original query is mapped to the original ontology via the formal links. For the common part of the information, there is no issue. For the parts that are not mapped the restrictions on the unmapped parts can be turned into keywords to be matched on the whole document generated from the filled form.

3 Evaluation

An evaluation was carried out to show how using K-Forms users can create a network of interconnected ontologies representing distributed communities and views without any user effort or specialised knowledge. In particular the aims were to evaluate:

- Knowledge reuse: can people look for knowledge they were unaware of and reuse it? Can users from distributed communities create interconnected knowledge in a decentralised way?
- Knowledge conceptualisation: can users conceptualise in terms of forms and use forms as a way to capture knowledge?

6 users tested the system individually. Each session lasted about 90 minutes and users were all academic persons with different degrees of familiarity with semantic technologies and ontology development. The data collected was both objective (logs of the interaction, screen activity) and subjective (questionnaire, interview). Questions were mainly on a 5 points Likert scale measuring the rate of agreement with a specific statement. Upon arrival, users filled in a personal profile questionnaire and were shown a short video (2 minutes) on the system and its use.

Two tasks, written as work task simulations [14] derived from real-world engineering tasks, were then presented to the users. The first task was to analyse a set of technical documents and create a form that could capture the contained knowledge, first on paper and afterwards using K-Forms. The second task was to analyse a different set of documents (that had some semantic intersection with the previous set) and create a form that could capture the knowledge, having as input also the ontology of the previous corpus (this was done in order to test the reuse of concepts). Half of the users received their own ontology (created in Task 1), while half received someone else's ontology (to test the difference between reusing personal knowledge and shared knowledge).

When both tasks were accomplished participants were asked to fill in a user satisfaction questionnaire that covered several aspects of the system, e.g. ease of use, ease of learn, perception of speed and accuracy, etc. The user evaluation ended with an interview aiming at eliciting explanations on participants' behaviour, impressions on the system and personal opinions.

Results for Knowledge Reuse

The ontologies created in task 1 were evaluated to check the degree of similarity when no reuse possibility was offered: users created unrelated ontologies with similar concepts but the similar concepts were not linked in the background.

In average 5 matching concepts between the ontologies were identified. Users created concepts with the same or very similar names to capture similar knowledge but the ontologies being unrelated impairs the possibility of reusing the concept or of searching across the different ontologies and instances.

When given the possibility to reuse an ontology (Task 2) users reused in average 60% of the possible concepts. The users' behaviour had many individual variations, with a peak of 80% and a minimum of 30% concepts reuse. In some cases users reused entire sections, modifying them to better suit their needs.

K-Forms knowledge capture method enabled people to discover new resources they were unaware of (as for ontologies created by other users given as input) and to directly connect what they thought was relevant. This bottom-up decentralized approach created a network of interconnected ontologies with no user effort and allows for a considerable reuse and ability to search across.

From the user interviews more details emerged on the sharing and reuse of concepts.

All the users were happy to reuse their own concepts as they knew not only the name of the concept but also the format they wanted it represented in (i.e. list or textfield) but they were more weary of reusing other people concepts as they may have been differently visualised. The percentage of reuse of own concepts is 68% while the reuse of someone else's concepts is 45%. This leads to the definition of a new requirement: enabling reuse of a concept whilst changing the visualisation manner.

During the interviews reuse strategies emerged: in general users were creating more generic concepts when aiming to reuse them while more specific when they thought they were peculiar to just one document type. Reuse of concepts was appreciated by 90% of the users as it saves time while one user still preferred to type his own concepts. When interviewing the user that did not want to reuse concepts, he commented about not being confident that the concept was exactly matching what he wanted to describe: this issue could be partially solved by showing instances to the users, so he could know whether the concept he is thinking to reuse meets his needs.

Results for Knowledge Conceptualisation

More detailed questions and task observations allowed us to evaluate the conceptualisation feature. When analysing the paper sketches of the corpora provided, all the users managed to easily sketch a form that could capture at least 90% of the existing concepts – some information was ignored as the meaning was not clear or as it was present only in part of the corpus. When translating the paper form into a K-Forms form all the users managed to translate it as they wished. Only 1 user found issues in translating a paper design as a required field type (list) was not available for use inside a table field. The user had to modify the form design to fit with system capabilities.

Overall K-Forms proved successful in supporting Knowledge Conceptualisation: users found easy or very easy (66.7%) to design a form using the system (33.3 % rated it average). Selecting sections, subsections and form fields was considered very clear by 66.6% and clear by 33%.

4. MCR: a K-Forms application

As a follow-up of this experiment, an experimental application was developed for Rolls-Royce plc for capturing and sharing information about inspections to jet engine components. An existing electronic form requiring the input of some 80 fields has been re-implemented using K-Forms. The form required input of text as well as uploading and annotation of images. Then a number of existing Module Condition Reports (MCR) have been used to fill the repository. The architecture supports a task where users have to visit overhaul shops around the world where no Intranet connection was available. Therefore they save the filled forms locally and upload them when go back to their offices at a future point. As soon as the triples are uploaded they become available to the rest of the community via K-Search.

The change in upload method from standard word processor documents to K-Forms was seamless and without consequences for the users. The possibility to query the knowledge contained in the forms through K-Search, however, makes the knowledge immediately searchable and reusable, and therefore we expect that many hours will be saved for knowledge workers who no longer need to read documents to access the contained knowledge.

Moreover, the knowledge about the condition of modules can be now correlated and connected to other knowledge contained in other repositories about the same topic. Other important services are now available for querying such as providing images of a module sorted by number of cycles (so to show the relation between condition and the amount of use). All these services where unavailable before the introduction of K-Forms: the only service available was a database allowing document access via some limited metadata (date, engine, component, etc.).

Results in module condition reports can now be compared to the results in two other repositories created for Rolls-Royce containing 18,000 event reports and about 11,000 technical variance documents created using Information Extraction from textual documents [8].

5. Comparison with the State of the Art

A number of previous works are relevant to K-Forms + K-Search. As for acquisition and representation of knowledge, standard technology like ontology editors can be used. However ontology editors like Protégé[1] require specialised knowledge and are generally unsuitable to a community of users who are primarily experts in their domain. The capture phase via Web forms was envisaged in [7] and [9] to capture knowledge and to connect the forms to an ontology. However, they did not envisage the possibility to effectively and efficiently enable the definition of user-defined forms. They still relied on technologies like X-Forms and the manual handcrafting of the forms. This limits the usefulness and flexibility of the tool for supporting distributed dynamic communities because they require the intervention of a computer scientist. Moreover, they do not provide anything like semantic searching mechanism

[1] http://protege.stanford.edu/

to access information, which has proven very effective in the three applications of K-Search. Form-based capture is also used in ontology-based knowledge portals [17]. These are system that use the ontology schema and its instances to capture to produce different views on the knowledge (e.g. to enable management of a large distributed project. Knowledge portals presented in literature require complex ontological modelling and therefore it misses the impact of the flexibility of K-Forms.

A technology similar to K-Forms is Semantic Forms within Semantic Media Wiki[2]. These semantic forms can be defined to capture specific semantic knowledge but lack graphical support to development. Knowledge within these semantic forms is captured without optimised storage for hybrid querying of the semantic information contained. Also Semantic Forms do not encourage reuse and interrelation across forms meaning gathered semantic knowledge is mostly unrelated. K-Forms differs by design to create and modify forms easily by users and communities to support emerging and rapidly evolving environments by non-technical users. Annotations are stored in a triple store enabling easy hybrid search. Moreover K-Forms enables to interconnect the underlying ontologies in an easy way in order to support search across form repositories.

The intuitive mixing and reuse of knowledge in existing ontologies has been explored in Potluck [5]. The system enables the mashing up of different pieces of ontological content. Their approach is data oriented, i.e. it involves the correlation of instances for mashing up, not the matching of the ontological schema as we enable in K-Forms. However, the spirit is very similar in the two systems.

A number of works have focused on ontology-based search. Most of the approaches, however, take the view that the user interaction should happen via keyword-based queries that are converted automatically into queries to the ontologies [10, 11, 15, 16]. In our view, the keyword-based approach has a number of issues. First it contributes to the disorientation of the users who do not know what to ask for if they are unfamiliar with the domain. This is visible for example in [16] where in the evaluation they had to remove a number of queries outside the coverage of the ontology. Moreover, keywords may have multiple interpretations (ambiguity) or no semantic interpretation. This requires sometimes sessions of refinements that can be disappointing for the users. Also it is unclear how to mix keywords and ontological queries, as it is difficult for the system to understand if part of the query is out of scope of the ontology or some parts must be interpreted as keyword matching. The use of Boolean operators *OR* and *AND* are difficult in these kind of interfaces, to the point that to our knowledge no one supports them. All the conditions are expected to be in AND. In K-Search, the explicit use of the ontology for query avoids the disorientation as it makes explicit the possible queries. It also enables the sophisticated use of Boolean operators. It enables to clearly define what is to be used as keywords and in what context (the whole document or just the sections annotated with a specific context). This enables very precise queries. Moreover, the possibility to specialize the query by right clicking on information in displayed documents enables a mixture of exploration and browsing which is mostly effective and liked by the users [8].

[2] http://www.mediawiki.org/wiki/Extension:Semantic_Forms

6. Conclusions and Future Work

In this paper we have described a suite of tools for knowledge management that enables easy and flexible knowledge acquisition, representation, capturing, retrieval, visualisation and sharing in large distributed organisations. K-Forms enables the intuitive design and deployment of web-based forms that capture semantic information. K-Forms provides a solution to the formalisation of knowledge capture for new cases, but does not provide a solution for legacy data, i.e. Word and Excel forms previously created. In these cases, information extraction from text can be applied; in [8] we have described two applications to two Rolls-Royce datasets that have been extremely successful. Now it is possible to search both new form-based data and legacy documents seamlessly. A user evaluation proved that K-Forms can successfully support users in conceptualising and reusing Knowledge: the users managed to create exactly the form they wanted in 90% of the cases and rated the process as very easy; in some cases they reused up to 80% of already existing concepts. K-Forms uses Semantic Web technologies in two ways: as a way to support the creation of forms (an ontology of form components and constraints guides the template creation) and as a way to create a domain ontology to support search.

K-Search enables accessing multiple repositories using either directly their local reference ontology or portions of other connected ontologies. Queries can either return a document generated with the retrieved knowledge (hence enabling user-specific information presentation), or a summary of knowledge in the form of graphs or statistics. Extracted triples can also be exported to an RDF repository or to a spreadsheet for further elaboration.

K-Forms and K-Search have been used for defining applications in real world environments in the aerospace domain.

Future work will concern further development of the concept of the networked ontologies and their impact on knowledge management. So far, that aspect has been only partially exploited as a way of searching across repositories. However, the creation of partially connected ontologies opens very interesting perspectives for knowledge management that go beyond just searching with a different perspective. Moreover, we need to explore the impact of changes to the existing form schema when some forms have been already filled. At the current point in time, the two versions of the same ontology are treated as different ontologies: as the new version is created by importing parts of the old one, search across the two is possible for the common parts. However, we feel that more sophisticated approach can be taken to address the issue.

Finally, a spin-out company (k-now.co.uk) has been created to commercialise the technology.

Acknowledgments. The work was supported by 1) IPAS, a project jointly funded by the UK DTI (Ref. TP/2/IC/6/I/10292) and Rolls-Royce plc and 2) X-Media, an Integrated Project on large scale knowledge management across media, funded by the European Commission as part of the IST programme (IST-FP6-026978), (www.x-media-project.org).

References

1. Thomas H. Davenport and Laurence Prusak: Working Knowledge: How Organizations Manage What They Know. Harvard Business School Pr, Boston. 1997.
2. Matteo Bonifacio, Paolo Bouquet and Roberta Cuel. Knowledge Nodes: the Building Blocks of a Distributed Approach to Knowledge Management. Journal for Universal Computer Science, Vol. 8/6, pp 652-661. Springer Pub. & Co. 2002.
3. Jim Hendler: Agents and the Semantic Web. IEEE Intelligent Systems, March/April 2001, Vol. 16, No. 2
4. Etienne Wenger: *Communities of Practice: Learning, Meaning, and Identity*, Cambridge University Press, 1998.
5. David Huynh, Robert Miller and David Karger: Potluck: Data Mash-Up Tool for Casual Users. Proceedings of the 6th ISWC and the 2nd ASWC, Busan, Korea, November 2007
6. Marta Sabou, Jorge Gracia, Sofia Angeletou, Mathieu D'Aquin and Enrico Motta: Evaluating the Semantic Web: A Task-based Approach. Proceedings of the 6th International Semantic Web Conference and the 2nd ASWC, Busan, Korea, November 2007
7. Swapna Gupta J. Scott Hawker and Randy K. Smith: Acquiring and Delivering Lessons Learned for NASA Scientists and Engineers: A Dynamic Approach. In Proceedings of the 43rd Annual Association of Computing Machinery Southeast Conference, March 2005, Kennesaw, GA.
8. Ravish Bhagdev, Sam Chapman, Fabio Ciravegna, Vitaveska Lanfranchi and Daniela Petrelli: Hybrid Search: Effectively Combining Keywords and Semantic Searches. In Proceedings of the 5th ESWC, Tenerife, June 2008
9. Marlon Dumas, Lachlan Aldred, Mitra Heravizadeh, and Arthur H.M. Hofstede: Ontology Markup for Web Forms Generation. Proceedongs of the WWW'02 Workshop on Real-World Applications of RDF and the Semantic Web, Honolulu, May 2002.
10. J Manuel Gomez-Perez, Mercedes Blazquez, Jesus Contreras, M^a Jesus Fernandez, Diego Paton and Luis Rodrigo: An Intelligent Search Engine for Online Access to Municipal Services. in Proceedings of 1st ESTC, Vienna May 31, June 1 2007
11. Yangui Lei, Victoria Uren and Enrico Motta: SemSearch - A Search Engine for the Semantic Web. In Proceedings of 3rd European Semantic Web Conference, Budva, Montenegro from the 11th - 14th June, 2006
12. Martin Dzbor, John B. Domingue and Enrico Motta: Magpie - towards a semantic web browser. In Proceedings of the 2nd International Semantic Web Conference, Sanibel Island, Florida, USA, 2003.

13 Vita Lanfranchi, Fabio Ciravegna and Daniela Petrelli,: Semantic Web-based Document: Editing and Browsing in AktiveDoc, Proceedings of the 2nd ESWC, Heraklion, Greece, 2005.

14 Pia Borlund: Experimental Components for the Evaluation of Interactive Information Retrieval Systems. In: Journal of Documentation, 56 (1) (2000) 71-90.

15. Tim Glover, Alistair Duke and John Davies: Squirrel: An Advanced Semantic Search and Browse Facility. In Proceedings of The 4th European Semantic Web Conference, Innsbruck, May 2007.
16. Thanh Tran, Philipp Cimiano, Sebastian Rudolph, and Rudi Studer: Ontology-based Interpretation of Keywords for Semantic Search. In Proceedings of the 6th International Semantic Web Conference and the 2nd ASWC, Busan, Korea, November 2007
17. Oscar Corcho, Asun Gómez-Pérez, Angel López-Cima, V. López-García and MC Suárez-Figueroa: ODESeW. Automatic Generation of Knowledge Portals for Intranets and Extranets. In Proc. of the 2nd International Semantic Web Conference, Sanibel island, October 2003