Assisting Requirements Recovery from Legacy Documents

Paul Rayson, Roger Garside and Pete Sawyer
Computing Department
Lancaster University
Lancaster
UK. LA1 4YR
Tel: +44 1524 593780
Fax: +44 1524 593608
{paul, rgg, sawyer}@comp.lancs.ac.uk

Abstract

Business change is often accompanied by loss of continuity of experience. This has serious implications for the adaptation of an organisation's software since people with detailed knowledge of either the software or business processes may be unavailable to inform its adaptation. In many cases organisational memory will persist principally in the form of documents such as requirements specifications, operating procedures, regulatory standards, etc. These offer an important resource for informing what features of the software are redundant, need to be retained or can be reused. Exploiting this resource poses formidable problems, however, since it is often incomplete, poorly structured, poorly maintained and voluminous. This paper proposes that tools exploiting probabilistic natural language processing techniques offer the potential to ease these problems. Such tools are available, mature and have been proven in other domains.

1. Introduction

Many organisations react to changes to their business environment by changing their strategic business goals and reengineering their organisational structures. This often dramatically changes the requirements of the socio-technical systems used at the operational level to implement the business processes. A precondition for understanding the implications of the changed requirements is an understanding of the systems' original requirements. Business reorganisation often means that this understanding is hard to acquire because continuity of experience has been lost. This paper reports the preliminary results of the REVERE[†] project where we are concerned with helping cope with this. We are investigating the recovery of requirements from documentation which frequently comprises an important element of the remaining organisational memory.

There are many types of legacy and classifications of change (e.g. [1]). We use that derived by Alderson and Shah [2]:

- *Strategic*: the boardroom-level view of how events in the business environment affect the business. Consequent change to the business might be imposed (e.g. by changed legislation) or it might be opportunistic (e.g. new business opportunities).
- *Organisational*: how the business structure supports the strategy. Changes to the strategy may necessitate reengineering the business. This may be horizontal, where roles and responsibilities are redefined, or vertical where support for new business sectors are introduced and redundant business areas are stripped away.
- *Operational*: the socio-technical systems that put the business processes into operation. If business change degrades their support for the business processes in terms of function, throughput, reliability, compliance with regulation, etc. they must be adapted accordingly.
- *Developmental*: the development and maintenance of the hardware and software that implement the automated parts of the socio-technical systems. Legacy may be a consequence of changes at the levels above but, it may also be independent of these; for example because of a critical shortage of COBOL programmers needed to maintain the software.

Our work is motivated by our industrial partner's experience of tackling organisational change that has *already occurred* (figure 1). In a typical scenario, change to the business strategy has led to both vertical and horizontal reengineering of the organisation and its business processes. The pace and/or

[†] REVerse Engineering of REquirements. EPSRC Systems Engineering for Business Process Change (SEBPC) programme project number GR/MO4846. Further details can be found at:
http://www.comp.lancs.ac.uk/computing/research/cseg/projects/revere/

scale of change has inhibited adaptation of the operational systems. Consequently, the organisation finds itself in a position where its new business processes are inadequately supported by the legacy software. This is illustrated by the experience of UK clearing banks. After many decades of relative stability, recent changes to the global financial services market have caused their core business to change from administering accounts to selling financial products [3]. As a result, they have a legacy of systems they cannot do without, but which inadequately support their new business. They still need to manage customers' accounts but they also need to support marketing requirements by, for example, building customer profiles from account data.

In a situation such as this, the legacy software needs to be adapted by, (e.g.) evolution or replacement. However, this change must be informed not only by the requirements of the changed business but also by the requirements that originally motivated the legacy systems. In figure 1, these are depicted by the grey block arrows bearing onto the existing operational software. These requirements may be derived from many levels; from the end-users who enact the business processes to strategic business goals. It is often tacitly assumed that while user requirements are relatively volatile, requirements that are consequent on the business strategy are relatively stable. However, changes to the business environment in the form of new legislation, globalisation and introduction of the Euro (among many others) show this assumption to be unsafe.

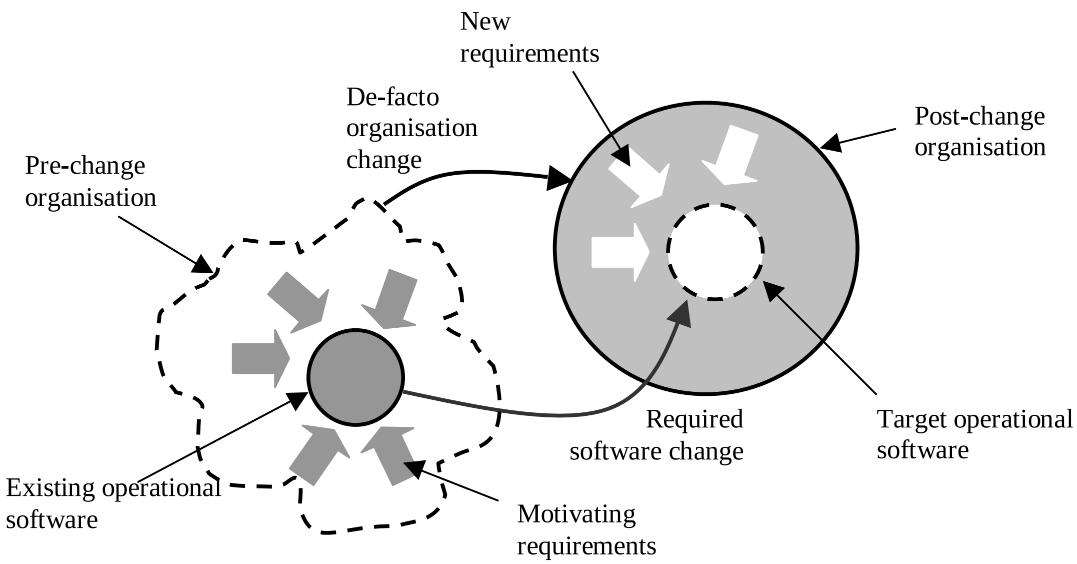

Figure 1. Legacy software and organisational change

Failure to understand the legacy software requirements and their motivation means that rational choices about how to adapt the software cannot be made. It will be uncertain what features of the software are redundant, need to be retained or can be reused. This uncertainty can lead to throwing the baby out with the bath water; solutions that support new business processes but fail to support key requirements that persist from the old business. More often, organisations dare not risk this happening and adopt costly solutions that retain functionality or data that is redundant. The uncertainty that leads to this is often a consequence of the loss of experienced people able to answer questions like "why does the system keep this data?".

This places a premium on the experience that is available and on other sources of the information. These are typically the legacy software itself and documents such as requirements specifications, operating procedures, regulatory standards, etc. In most cases these sources will be incomplete but complementary and the analyst will need to use each to construct partial models that can be verified against each other. The REVERE project is specifically concerned with legacy requirements recovery from documents. In the following section we give our motivation for this focus, describe the practical difficulties it poses and introduce our proposed solution.

2. Requirements recovery from documents

The process of recovering the requirements for existing software systems from fragmentary sources of information is analogous to following multiple audit trails that lead from requirements inferred from the domain or business into technical documentation and through to the software. The analyst must use whatever information resources are available to construct conceptual models of the pre-change organisation and its business processes and from these derive the requirements of the legacy software [4]. This typically entails an iterative process of inferring stakeholders, roles, tasks and business objects and verifying these against the structure and behaviour of the in-service software.

This information has to be gathered from many different sources, both human and documentary. The elicitation of information from human stakeholders has received a great deal of attention elsewhere and there has also been some work on the reverse engineering of source code. The retrieval of requirements information from free text documents has, by comparison, been neglected.

Good requirements engineering practice [5] recommends that requirements are carefully documented and managed. Hence, it should be straightforward to list the stakeholders' requirements, understand their motivation and consequent trade-offs, and trace them forwards (ultimately) into the operational software. Unfortunately, good requirements practice is rare and systems are still routinely constructed with minimal requirements documentation [6]. Some domains (e.g. defence) place a premium on documentation and here it is reasonable to expect requirements specifications, operating procedures, safety cases, etc. to be available. In most cases the available documentation will be less comprehensive but if any documentation does exist, it will represent a potentially important resource; particularly where human expertise is patchy. Given the existence of *some* documentation, the real problem is then how to process it. This will be hard if the volume is large; in extreme cases, there may be filing cabinets full. Similarly, variable quality and the structure of the documents will pose problems if, for example, they are heavily cross-referenced and version control has been poor. This is compounded by the linear structure of paper documents. Even if the documents have good tables of contents, have comprehensive indexes and are in, or can be transformed (via scanning and OCR) into, electronic form, the documents' as-written structure inevitably constrains the way in which people can read and interact with them.

Identification and assimilation of the subset of useful information contained in the documents is therefore difficult, costly and error-prone.

Our aim is to develop tools to ease these problems by exploiting mature techniques for natural language processing (NLP). Although technical documents often employ special notations, such as object models, workflow diagrams, etc., the bulk of nearly all such documents is comprised of natural language. Ryan [7] has noted the promise of NLP for information retrieval [8] from textual requirements. Several researchers have used rule-based NLP techniques for synthesising database conceptual schemas [9], generating graphical representations of VLSI system requirements [10] and automatically abstracting requirements from requirements specifications [11]. However, these approaches are all hamstrung by the limitations of the rule-based NLP techniques they employ. Natural language (NL) is invariably so complex that a very large base of grammar rules is required even for small NL subsets. All the techniques above depend for their efficacy on a tightly constrained subset of English.

Analysts of legacy software have no control over the documents that they must analyse so purely rule-based techniques are impractical. Our approach is therefore to exploit *probabilistic* NLP techniques that were pioneered at Lancaster and a number of other sites in the 1980s. Instead of attempting to model the grammar of a NL as a set of grammar rules, the class of probabilistic tools that we are interested in classify words on the statistical likelihood of them being a member of a particular lexical, syntactic or semantic category in a particular context. The probabilities are derived from very large corpora of free text which have already been analysed and 'tagged' (often manually) with each word's lexical, syntactic or semantic category. Extremely large corpora have been compiled (the British National Corpus consists of approximately 100 million words [12]). For some levels of analysis, notably part-of-speech tagging, probabilistic NLP tools have been able to achieve levels of accuracy and robustness that rule-based techniques cannot approach.

Probabilistic tools do not attempt to automate understanding of the text. Rather, they abstract interesting properties of the text that a human user can combine and use to infer meaning. Evidence from other domains suggests that such tools can be effectively used to provide a 'quick way in' to large documents. For example, in [13] probabilistic NLP tools were used to quickly confirm the results of a painstaking manual discourse analysis of doctor-patient interaction. In this application, they were also able to reveal information that had not been discovered manually.

A further, crucial, characteristic of probabilistic NLP techniques is that they scale. The execution time of the tagging process varies approximately linearly with the document size. Once the text has been tagged, retrieval and display tools allow the user to interact with the document. These use the tags to provide views on the document that reveal interesting properties and suppress the bulk of text. They do this in a way that is largely independent of the size of the document. Hence, the user is protected from information overload by being selective about the information they want to extract.

We do not claim that NLP will solve the problem of requirements recovery from documents. The information recovered will never be complete or a perfectly accurate snapshot of the legacy software's requirements. What is recoverable is bounded by the quality of the documents themselves. However, at present, analysts have very few useful tools to help recover this information. We believe that probabilistic NLP are sufficiently mature to make a substantial improvement if they can be integrated within a framework of guidance for what properties to look for and how to interpret them.

3. Preliminary results

During the preliminary stage of REVERE, we have adapted, and experimented with, a set of existing NLP tools developed at Lancaster for the processing of English language text. The most important of these is CLAWS [14]. CLAWS uses a statistical hidden Markov model technique and a rule-based component to identify the parts-of-speech (POS) of words in a document to an accuracy of 97-98%. CLAWS' output is the document text annotated with POS tags and this provides the foundation for further levels of analysis. A semantic analyser [15] uses the POS-tagged text to assign semantic tags that represent the general sense field of words from a lexicon of single words and an idiom list of multi-word combinations (e.g. 'as a rule'). This lexicon contains approximately 52000 words or idioms and classifies words according to a hierarchy of semantic classes. For example, the tag *A1.5.1* represents words or idioms meaning *Using* which is a subclass of *general and abstract terms*. Words that would be assigned this tag (in the appropriate POS context) include *user*, *end-user* and *operator*. Similarly, the tag X2.4 is a subclass of *Psychological actions, states and processes* and would be assigned to terms meaning *Investigate*, such as *search*, *browse* and *look for*.

A further tool, XMATRIX [16] provides a means for retrieving the analysis results. XMATRIX is used to perform *frequency profiling.* At the most basic, this produces a simple concordance of individual words in context. However, POS and semantic tagging allows more useful concordances to be formed. For example, XMATRIX can show the frequency of occurrence of different parts of speech. An obvious example of how this can benefit the identification of requirements is the extraction of all the occurrences of modal verbs ('shall', 'must', 'will', 'should', etc.). Expressions of need, desire, etc., consistent with user or system requirements can therefore be located in a document very easily and without the need to construct complex regular expressions or search templates. Even this basic level of analysis goes beyond what the current generation of commercial requirements and document management tools allow.

Frequency profiling becomes even more useful when a document can be compared against a *normative corpus*: a large body of pre-tagged text from a representative domain. For this we are tagging a number of public-domain software and systems engineering standards, operating manuals, a large IBM technical documents corpus, a corpus of text from the applied sciences (a subset of the British National Corpus) and a number of technical reports and papers. Comparison with the normative corpus allows information to be extracted from a document by searching for statistically significant deviations from the frequency norm suggested by the corpus.

To provide a snapshot of the results so far, and illustrate some of the key issues, we now briefly describe three examples.

3.1 Example #1: A library information system

Our first experiment was with a 34-page user requirements definition of a library information system. As requirements documents for legacy systems are not always available for analysis, the main point of using this document was to help calibrate the tools rather than provide a realistic simulation. Nevertheless, the results demonstrate some interesting points.

The tools were applied to the document and the semantically tagged text was sorted by deviation from a normative corpus – a semantically tagged subset of the British National Corpus (BNC). It revealed library-specific domain terms quite clearly. However, as described below, it also revealed some anomalies with some of the technical terms (that could perhaps be characterised as jargon).

The most over-represented semantic categories (under-represented categories can also be interesting) included:

717.5	Q1.2	*paper documents and writing* ('documents', 'records', 'prints');
483.8	T1.1.3	*time, future* ('will', 'shall');
260.1	A1.5.1	*using* (phrases containing: 'user', 'end-user');
208.8	I2.1	*business* ('agents', 'commercial');
159.5	S7.1+	*power, organising* ('administrator', 'management', 'order');

144.6	Q4.1	*the media* ('author', 'catalogues', 'librarian');
129.9	X9.1+	*ability, intelligence* ('be able to').
102.5	X2.4	*investigate* ('search');

The figures in the left-hand column represent the semantic categories' log-likelihood (LL) figure - a measure of statistical significance. The middle column is the mnemonic for the semantic category and the right-hand column is the semantic category's label and typical examples from the analysed text.

The LL shows the degree of deviation from the norm. A LL greater than 6.63 indicates that there is a 99% confidence in the result's reliability. Hence, the above 8 semantic categories stand out as highly unusual deviations from normal English usage, which suggest that it would be rewarding to investigate the usage of the corresponding words or idioms within the document.

In this case, for example, 'author', 'catalogue' and 'search' emerge as candidate roles, objects and tasks respectively. To further investigate these, and validate whether these are really candidates for components of a conceptual model, XMATRIX can be used to display each semantic category (figure 2). This shows words and idioms of the semantic category *paper documents and writing* (tag Q1.2) in the context in which the words appear in the document.

From this, the analyst can infer that significant objects within the library application include 'document', 'web page' and 'photocopy'. Similarly, 'print' and 'receive' are candidates for operations on the objects.

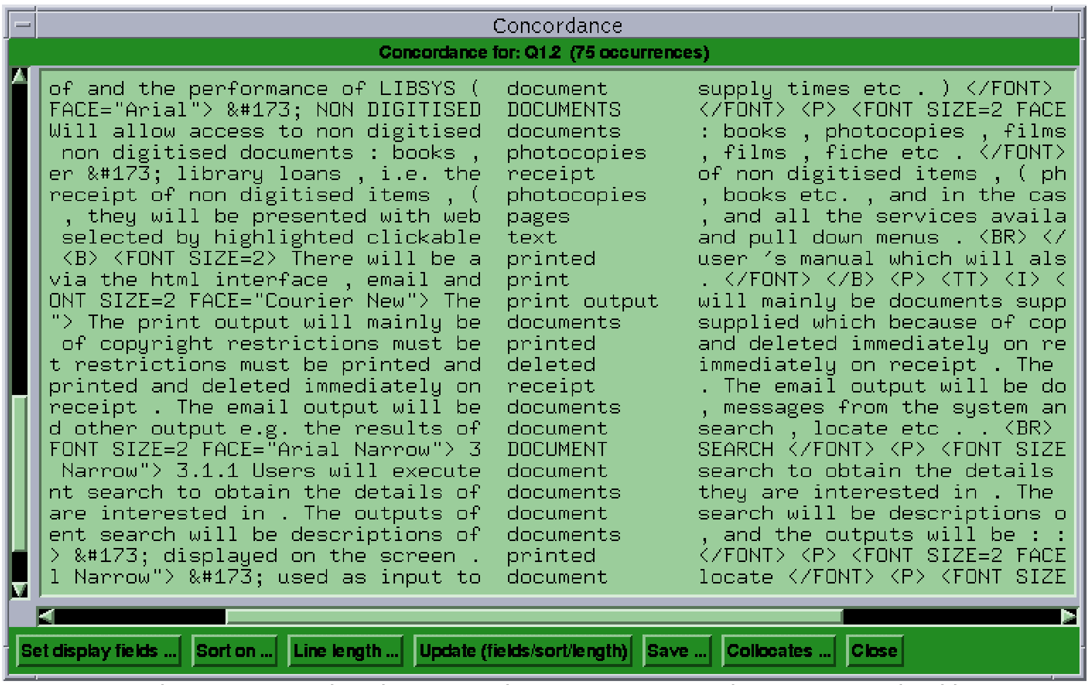

Figure 2. Browsing the semantic category *paper documents and writing*

3.2 Example #2: Air-traffic control

The second example illustrated uses a very different class of document. The target documents are field reports of a series of ethnographic studies at an air traffic control centre. These consist of both the verbatim transcripts of the ethnographer's observations and interviews with controllers, and of reports compiled by the ethnographer for later analysis by a multi-disciplinary team of social scientists and software engineers. Quite apart from representing a technique (ethnography) that would benefit from tools to help analyse the output, the field reports form an interesting study because they exhibit many characteristics typical of legacy documents. The volume of the text is a little higher than in example #1 (103 pages) but, more significantly, the density of the information is high and the documents are not structured in a way (say around business processes or system architecture) designed to help the extraction of requirements.

The documents were processed using the same procedure that we applied in example #1. This time, however, two different normative corpora were used in the analysis. These represented two different approaches to corpus selection. The first was an attempt to introduce an element of domain filtering. In the absence of a corpus of the ATC domain, we selected (perhaps naively), one oriented to technology.

This is a subset of the BNC from the pure and applied science section of the BNC related to information technology. Collectively, these form a corpus of 1.7 million words, of which about 60% are news stories relating to IT. Using this corpus, the most over-represented semantic categories in the ATC field reports were (sorted by significance of deviation from the norm):

1927.7	M5	*flying* (‘plane’, ‘flight’, ‘airport’)
1500.8	Z8	*pronouns* (‘you’, ‘they’)
627.2	Z4	*interjections* (‘um’, ‘yeah’)
502.9	S7.1+	*power, organising* (‘controller’, ‘chief’)
412.8	A3+	*being* (‘is’)
330.1	L1+	*life and living things* (‘live’)
239.9	O3	*electrical equipment* (‘radar’, ‘blip’)
234.1	W3	*geographical terms* (‘Pole Hill’, ‘Dish Sea’)
184.5	Z6	*negatives* (‘not’)
158.6	N3.7	*measurement* (‘length’, ‘height’, ‘distance’, ‘levels’, ‘1000ft’)
146.8	M1	*moving* (‘arrival’, ‘incoming’, ‘climb’, ‘descent’)
126.3	L2	*living creatures* (‘wing’, ‘wings’)

The following summarises the useful data from this list:

- M5 is a major class of domain objects and emerges clearly as the most over-represented category.
- S7.1+ includes the principal user roles.
- L1+ (the word ‘live’) is an attribute of a major object type (flight strip) who’s LL is too small for it to appear on this list.
- O3 includes electronic objects and properties of those objects.
- W3 are instances of geographical objects (the names of sectors of airspace and aircraft reporting ‘beacons’).
- N3.7 includes attributes and values taken by some of the objects
- M1 shows actions performed by some of the objects
- L2 refers to an area of the ATC suite and the people who work in that area (‘wingmen’)

Hence, by using XMATRIX to view the semantic categories in context, the analyst would be able to identify some of the major domain objects, roles, functions, etc. for ATC (see Figure 3). However, there is also much noise in the result. For example, Z8, Z4 and A3+ are largely a consequence of the fact that much of the documents’ contents are text quoted verbatim from users. Given this characteristic of the documents, it is perhaps unsurprising that an IT-oriented normative corpus should throw up significant deviations from the norm that are of no significance to the domain under investigation.

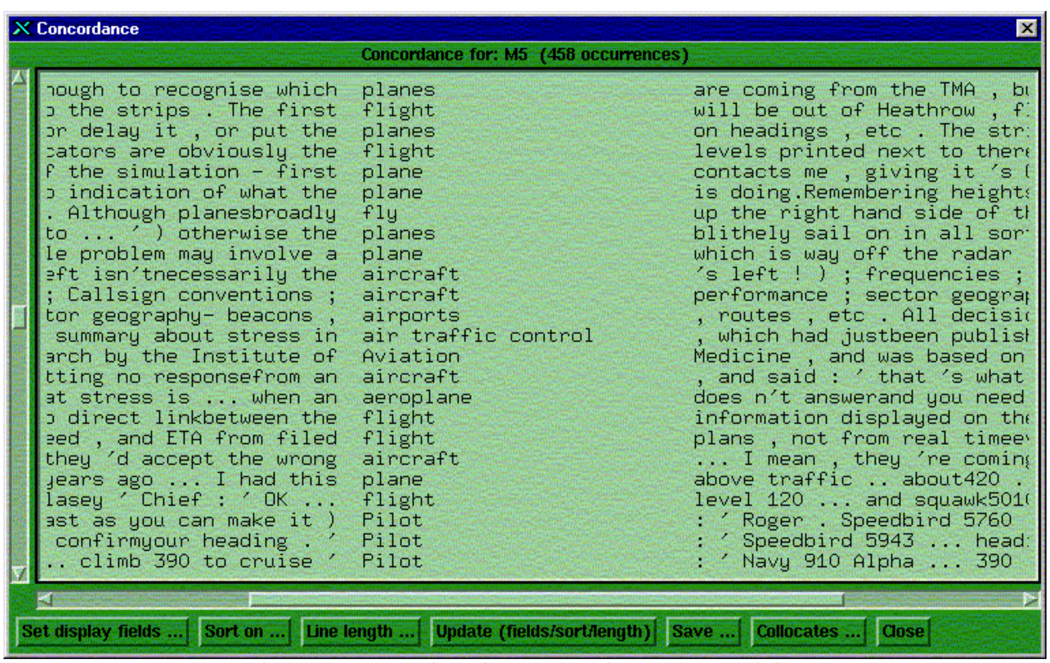

Figure 3. Browsing the semantic category *flying*

In view of this, the second normative corpus that we applied to the same documents was a 2.3 million-word subset of the BNC derived from the transcripts of spoken English. The results from this were quite different:

3366.66	S7.1+	as above
2578.80	M5	as above
988.09	O2	*general objects* (‘strip’, ‘holder’, ‘rack’)
643.54	O3	as above
535.97	Y1	*science and technology* (‘PH’)
449.34	W3	as above
432.33	Q1.2	*paper documents and writing* (‘writing’, ‘written’, ‘notes’)
372.80	N3.7	as above
318.82	L1+	as above
310.32	A10+	*indicating actions* (‘pointing’, ‘indicating’, ‘display’)
306.90	X4.2	*mental objects* (‘systems’, ‘approach’, ‘mode’, ‘tactical’, ‘procedure’)
290.06	A4.1	*kinds, groups* (‘sector’, ‘sectors’)

With the exception of Y1 (an anomaly caused by an interviewee’s initials being mistaken for the PH unit of acidity), all of these semantic categories include important objects, roles, functions, etc. in the ATC domain.

It is interesting that in example #2, the normative corpus that was not domain-specific performed better than the normative corpus that we (wrongly, as it turned out) guessed would provide a partial match to the domain. In retrospect, it seems likely that this was due to the fact that the domain of ATC has very little to do with IT. A more relevant comparison would be to derive an ATC normative corpus and apply that to the ethnographic field reports.

However, we believe that the results using the tools in their current untuned form are potentially very useful, provided that a normative corpus is carefully chosen. Initial results indicate that where no domain-specific corpus exists, it is better to default to a non-specific corpus, and then manually separate domain-specific from application-specific variations.

3.3 Example #3: Banking

As a further illustration of the success of a comparison to a non-specific corpus, we present a third example. The corpus under consideration contains reports and notes from an ethnographic study of financial services in a major ‘high street’ bank and fieldwork that examined features of working in a major Lending Centre and, in particular, the work of the 'Technology Co-ordinator' [17]. This is the largest of the three examples, consisting of 244 pages of text, to which we applied the same process as above. On comparison with the 2.3 million-word spoken subset, we observed the following significant results:

3461.86	S7.1+	*power, organising*: ('co-ordinate', 'control' as in lending control, 'administrative')
3369.71	I3.1	*work and employment generally* ('work', 'working', 'employed')
3318.69	Y2	*IT and Computing* ('computer', 'software', 'screen')
2343.24	Q1.2	*paper documents and writing* ('paperwork', 'file', 'notes', 'brief' as in Customer brief)
1857.55	S5+	*groups and affiliation* ('Branch', 'Team', 'Unit' as in Lending Unit)
1772.26	I2.2	*business selling*: ('lease', 'sales', 'flogging', 'markets')
1765.56	I2.1	*business generally* ('business' as in business centre, 'audit')
1361.74	S8+	*obligation & necessity* ('teamwork', 'support', 'service', 'Assistant Manager')
1340.13	A6.2+	*comparing usual* ('everyday', 'routine', 'standardised')
1127.62	I1	*money* ('buck', 'account', 'cheques', 'interest rates', 'ledger', 'balance')
1114.05	A9-	*giving* ('lending', 'borrowing', 'loan')
866.82	I1.1	*affluence* ('credit', 'income', 'profits', 'capital', 'funds', 'pay in', 'savings', 'investments')

Once again, this third example shows that the software is able to extract important features from the content of the text under examination. In particular, I2.2 (business selling category) highlights the new nature of the bank's business of selling financial products.

4. Future development

Our next task is to devise a large-scale experiment with our project partners and refine the semantic categories and lexicon of words and idioms (originally defined by professional linguists for general English). At present, the lexicon contains a number of categorisations that, in an IT context, appear anomalous. For example, while 'browse' would be tagged with the semantic category *investigate* along with 'search' and 'look for', 'query' would be tagged as a *speech act*. We expect that it will be necessary to refine the lexicon to derive an IT-oriented lexicon and semantic classification.

Our expectation is this would underpin a core toolkit that can be tailored for particular application domains. Hence, for example, an analyst would be able to add new semantic categories for banking or railway signalling. A capability already exists to refine the lexicon by reclassifying words or idioms. This will have to be extended with the additional abilities to import and classify new words or idioms, and to define new semantic categories.

To further extend the utility of the NLP tools, we plan to develop a framework of guidance for constructing scenarios and conceptual models (possibly with some automated support [18]) from the information. The NLP tools will be integrated with a tool (JPREview) that implements the PREview method [19, 20] for modelling systems and processes using viewpoints. This will support an iterative investigative process where the analyst posits a set of stakeholder types and iteratively refines this set, confirming or discounting the posited viewpoints and extending the set as new ones are inferred by the text analysis (figure 4).

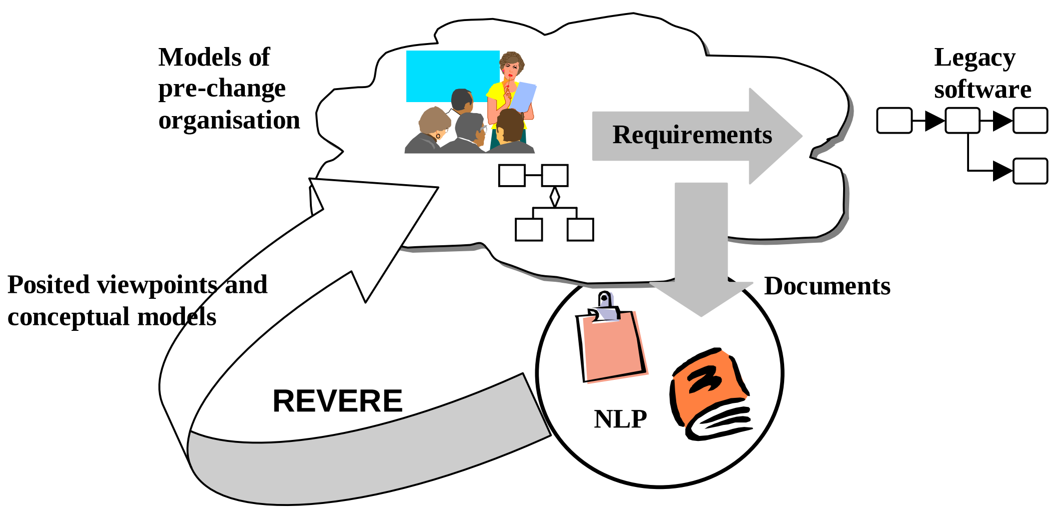

Figure 4. Iterative NLP - informed modelling

5. Conclusions

This paper has described the preliminary results of the REVERE project investigating support for the recovery of legacy software requirements from documents. The motivation for this is that:

- Decisions about how to adapt legacy software must be properly informed by an understanding of what is redundant, what must be retained and what can be reused. A precondition for this is that the requirements that motivated the legacy software before the organisational change occurred must be understood.
- Documents often form an important, and sometimes the primary, source of information about the legacy software and the pre-change organisation. These documents are often large and of variable quality so extracting information from them can be costly. They are usually written in natural language.

We plan to integrate a number of techniques to provide a set of tools to help analysts explore the documentation, and reconstruct models of the business that motivated the software. At the core of this toolset are probabilistic NLP tools to provide a 'quick way in' to large, complex and imperfectly structured documents. At present, little effective support is available for the analysis of such documents. Probabilistic NLP offers the potential to save much painstaking and error-prone manual effort. A crucial requirement of our work is that it must scale in a way that the manual analysis of documents does not. The probabilistic NLP tools that we have chosen have been proven in other domains to do so. They are also mature and can tolerate variation in the use of language contained in documents. Hence, in contrast to other attempts at applying NLP techniques to the analysis of technical documents, they are not restricted to a well-defined subset of (e.g.) English.

The paper describes initial experiments on an English language requirements definition document for a library system, and ethnographic field notes and reports from both the air traffic control and banking domains. These have been used to illustrate the use of frequency profiling, part-of-speech tagging and semantic tagging to reveal interesting properties of the text. For example, certain semantic categories of words appear in the documents with a frequency that significantly differs than a norm suggested by a corpus of English text. Examining occurrences of words of these categories reveals words that the analyst can infer to represent objects, roles and tasks.

6. Acknowledgements

We are grateful for the support of our industrial partner, Adelard, who has provided us with motivation, data and advice.

7. References

1. Lam, W., Loomes, M.: "Requirements Evolution in the Midst of Environmental Change*", Proc. Second Euromicro Conference on Software Maintenance and Reengineering, Florence*, 1998.
2. Alderson, A., Shah, H.: "Viewpoints on Legacy systems", *Communications of the ACM*. (in press)
3. Blythin, S., Rouncefield, M., Hughes, J.: “Never Mind The Ethno Stuff - What Does All This Mean and What Do We Do Now?”, *ACM Interactions*, 4 (3), 1997.
4. Butler, K., Esposito, C., Hebron, R.: "Connecting the Design of Software to the Design of Work", *Communications of the ACM*. 42 (1), 1999.
5. Sommerville, I., Sawyer, P.: Requirements Engineering - A Good Practice Guide, John Wiley, 1997.
6. Melchisedech, R.: "Investigation of Requirements Documents Written in Natural Language", *Requirements Engineering*, 3 (2), 1998.
7. Ryan, K. :"The Role of Natural Language in Requirements Engineering", *Proc. IEEE International Symposium on Requirements Engineering*, San Diego, January 1993.
8. Blair, D., Maron, M. :"Full-Text Information Retrieval: Further Analysis and Clarification", *Information Processing and Management*, 26 (3), 1990.
9. Rolland, C., Proix, C. :"A Natural Language Approach for Requirements Engineering", *Lecture Notes in Computer Science*, Vol. 593, 1992.
10. Cyre, W., Thakar, A. :"Generating Validation Feedback for Automatic Interpretation of Informal Requirements", in *Formal Methods in System Design*, Kluwer, 1997.
11. Goldin, L., Berry, D.: "AbstFinder, A Prototype Natural Language Text Abstraction Finder for Use in Requirements Elicitation", *Automated Software Engineering*, 4, 1997.
12. Aston, G. and Burnard, L.: The BNC Handbook: Exploring the British National Corpus with SARA, Edinburgh University Press, 1998.
13. Thomas, J., Wilson, A.: "Methodologies for Studying a Corpus of Doctor-Patient Interaction", in Thomas, J. and Short, M. (eds.) Using Corpora for Language Research, Longman, 1996.
14. Garside, R., Smith, N.: "A Hybrid Grammatical Tagger: CLAWS4", in Garside, R., Leech, G., and McEnery, A. (eds.) Corpus Annotation: Linguistic Information from Computer Text, Longman, 1997.
15. Rayson, P., and Wilson, A.: "The ACAMRIT semantic tagging system: progress report", *Proc. Language Engineering for Document Analysis and Recognition (LEDAR)*, Brighton, England. 1996.
16. Rayson, P., Leech, G., and Hodges, M.: "Social differentiation in the use of English vocabulary: some analyses of the conversational component of the British National Corpus", *International Journal of Corpus Linguistics*. 2 (1), 1997.
17. Hughes, J., O’Brien, J., Rouncefield, M., and Tolmie, P. "Some 'real' problems of 'virtual' teams." Presented at the Labour Process Conference, Manchester, 1998.
18. Burg, J., van de Riet, R.: "COLOR-X: Object Modeling profits from Linguistics*", Proc. Second International Conference on Building and Sharing of Very Large-Scale Knowledge Bases (KB&KS'95)*, Enschede, The Netherlands, 1995.
19. Sommerville, I., Sawyer, P., Viller, S.: "Viewpoints for Requirements Elicitation: a Practical Approach", *Proc. Third IEEE International Conference on Requirements Engineering (ICRE 98)*, April 1998.
20. Sommerville, I., Sawyer, P., Viller, S.: "Managing Process Inconsistency using Viewpoints", IEEE Transactions on Software Engineering (to appear).