Recovering Legacy Requirements

Paul Rayson, Roger Garside and Pete Sawyer

Computing Department, Lancaster University,
Lancaster, UK. LA1 4YR
{paul, rgg, sawyer}@comp.lancs.ac.uk

Abstract. It is common for organisations to introduce substantial changes to their structure and operations in order to adapt to new business environments. This often confers legacy status on their software systems because they can't adequately support the new business processes. In this paper, we argue that it is necessary to recover the requirements of in-service legacy software to ensure that its evolution or replacement is properly informed by an understanding of what is redundant, what must be retained and what can be reused. Much of this information is often contained in documents. However, retrieval of the information is often difficult due to problems of completeness, quality and sheer volume. In the REVERE project we are integrating a number of techniques to provide a set of tools to help requirements engineers explore the documentation and reconstruct conceptual models of the software and business processes. At the core of this work is the exploitation of probabilistic NLP tools to provide a 'quick way in' to large, complex and imperfectly structured documents, saving much painstaking and error-prone manual effort.

1. Introduction

This paper reports the preliminary results of the REVERE[1] project, which is concerned with informing decisions about legacy software in changing organisations. Many organisations are buffeted by change to their business environment and react to this 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. Informing the process of adapting to such change poses a major challenge for requirements engineering (RE).

There are many types of legacy [1] and classifications of change [2] but we address only a subset of these. Our work is motivated by our industrial partner's experience of tackling organisational change that has *already occurred*. The pace and/or scale of change has not permitted the smooth adaptation of the operational

[1] 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/. We are grateful for help from our industrial partner, Adelard, who provided us with motivation, data and advice.

systems. Consequently, planning for change has not been possible and the business processes that are required to support the new business strategy are not supported by adequate operational systems.

Change to the systems and their software must be informed not only by the requirements of the changed business but also by the requirements that originally motivated the legacy systems. Failure to understand this risks key requirements implicit in the software going unsupported by the new or evolved software. Even if the risk is acknowledged, the degree of risk sometimes remains unknown leading to costly solutions that dare not discard redundant functionality or data. The risk exists because business change often takes place against a background of poor organisational memory. Experienced people are not available to articulate new requirements or answer questions like "why does the system keep this data?".

This is further complicated by the orthodox view that requirements derived from the business domain are stable. However, the effects of new legislation, globalisation and introduction of the Euro (among many others) show this assumption to be unsafe. For example, recent changes to the global financial services market have caused the core business of UK clearing banks 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.

We are investigating how the integration of a number of techniques can help reverse engineer the requirements for legacy software so that the risks of change can be properly evaluated. The information resources available to do this are likely to be incomplete and a mixture of the legacy software itself, the knowledge of people within the organisation, and documents relating to the software and the pre-change organisation. Our focus is on the use of these documents to help reconstruct models of the legacy software and the business processes it was developed to support.

2. Requirements Recovery Sources

To inform the adaptation of legacy, the following questions need to be answered:

- What were the requirements of the existing system?
- What motivated these requirements?

Only once these have been answered can we determine what needs to be retained from the existing system to meet the requirements of the changed organisation. The process of answering the questions 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 requirements engineer 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. 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. In a typical legacy application, none of these will be complete so it is necessary to make best use of what is available. 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. Retrieval of requirements information from free text documents has, by comparison, been neglected.

In practical terms, recognition of the importance of requirements management is a recent phenomenon and the documentation of legacy systems will almost certainly not reflect current best practice [4]. Some domains (e.g. defence) place a premium on documenting requirements, operating procedures, safety cases, etc., but in most cases the available documentation will be less comprehensive. Assuming the existence of documentation in some form, the real problem is then how to process the documents. There may be many documents (filing cabinets full in some cases) which may be heavily cross-referenced and of variable quality. Even if a requirements specification exists, recovery of the key customer requirements will be difficult if, as is usually the case, it is poorly maintained, poorly structured, and untraced [5].

Identification and assimilation of the subset of useful information contained in the documents is therefore very difficult and labour-intensive. 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.

Our aim is to develop tools to ease these problems by exploiting mature techniques for natural language processing (NLP). Ryan [6] has noted the promise of NLP for information retrieval from textual requirements. A number of researchers have applied rule-based NLP techniques to requirements understanding by building semantic network models of requirements (e.g. OICSI [7], ASPIN [8]). These have typically been used to synthesise lower-level models such as database conceptual schemas. AbstFinder [9] abstracts key requirements from free text documents. However, these approaches are characterised by the limitations of the rule-based NLP techniques they employ: because of the complexity of natural language (NL) they can only cope if a well-defined NL subset is used to express the requirements.

This is an unrealistic restriction. Our approach is therefore to exploit *probabilistic* NLP techniques that were pioneered at Lancaster in the 1980s. Rather than modelling NL as a set of grammar rules, the tools classify words on the statistical likelihood of them having a particular syntactic or semantic function in a given context. The probabilities are derived from very large corpora consisting of free text (the British National Corpus is approx. 100 million words [10]) which have already been analysed and 'tagged' (often manually) with each word's syntactic or semantic category.

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 [11] probabilistic NLP tools were used to quickly confirm the results of a painstaking manual discourse analysis of doctor-patient interaction. 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, good performance can be achieved by interactive tools for retrieval and display. These use the tags to provide views that

slice through the text to reveal interesting properties. Hence, the user is given a means to control and interact with the documents that is largely independent of their size. Our hypothesis is that similar benefits may be accrued by applying statistical NLP techniques to requirements recovery for legacy systems.

Of course, we cannot hope to provide a full-proof answer to requirements recovery from documents. The information recovered will never be complete or a perfectly accurate snapshot of the legacy software's motivating requirements. What is recoverable is bounded by the quality of the documents themselves. However, we believe that the potential exists to substantially improve on what is currently an entirely manual and error-prone task.

3. Preliminary Results and Next Steps

We have been adapting, and experimenting with, a set of existing NLP tools developed at Lancaster for the processing of English language text. CLAWS [12] 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%. This provides the foundation for further levels of analysis. We use a semantic analyser [13] to assign tags that represent the general sense field of words and multi-word combinations. The tags classify words according to a hierarchy of semantic classes.

A further tool, XMATRIX [14] provides a means for retrieving results via *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. XMATRIX allows extraction of all phrases containing modal verbs ('shall', 'must', 'will', 'should', etc.). Hence, the occurrence of expressions of need, desire, etc., consistent with user or system requirements can 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.

However, 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 have tagged 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. One of our first experiments was with a user requirements definition of a library information system. When we sorted the semantically tagged text by deviation from the norm, among the most over-represented semantic categories (underused categories can also be interesting) were:

- *using* (e.g.: 'user', 'end-user');
- *business* ('agents', 'commercial');
- *paper and document writing* ('documents', 'records', 'prints');
- *the media* ('author', 'catalogues', 'librarian');
- *power, organizing* ('administrator', 'management', 'order');

- *time, future* ('will', 'shall');
- *investigate* ('search');
- *ability, intelligence* ('be able to', 'will', 'must not').

This initial experiment illustrates a number of interesting things. Candidates for roles ('author'), objects ('catalogue'), tasks ('search') emerge as unusually frequently used terms. In the case of this document the principal roles, objects and tasks correspond very closely to the most statistically significant deviations from the norm. The ability to view the words that corresponded to these in their proper context allows the requirements engineer to quickly confirm or reject the candidate objects etc.

We plan to exploit this by integrating the NLP tools with a viewpoint-oriented RE tool (JPREview) that implements the PREview method [15, 16]. This will support an iterative investigative process where the requirements engineer 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. To support this, we plan to develop methodological guidance for constructing scenarios and conceptual models (possibly with some automated support [17]) from the information to scope the stakeholder viewpoints and infer their requirements.

We should note that the experiment described above tends to flatter our approach because the document it applied to conforms to good requirements definition practice, is relatively small (24 pages of text) and we were familiar with it. Information is unlikely to emerge so easily from poorly maintained documents. Nor will relevant information be so easily extracted from documents of (e.g.) business procedures because such documents are unlikely to embody information about the role, scope or context of the legacy software. However, we believe that the tools nevertheless have the potential to help manage large amounts of textual data and the rapid, iterative refinement of theories and models of the system under investigation.

To verify this, our next task is to devise a large-scale experiment 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 a software engineering 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*.

4. Conclusions

This paper has argued for the need to recover the requirements of in-service legacy software to ensure that evolution of that software is properly informed by an understanding of what is redundant, what must be retained and what can be reused. We also argue that in many applications, much useful information about the requirements is implicit, but locked up in documents. Indeed, in many organisations, the nature of the business change that has conferred legacy status on their software has also resulted in the near-elimination of alternative sources of information by causing people with expert knowledge to leave, retire or be re-deployed.

We plan to integrate a number of techniques to provide a set of tools to help requirements engineers explore the documentation, and reconstruct models of the business that motivated the software. At the core of this work is the exploitation of

probabilistic NLP tools to provide a 'quick way in' to large, complex and imperfectly structured documents, saving 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 tools that we have chosen have been proven to do so.

References

1. Alderson, A., Shah, H.: "Viewpoints on Legacy systems", *Communications of the ACM*. (in press)
2. Lam, W., Loomes, M.: "Requirements Evolution in the Midst of Environmental Change", *Proc. Second Euromicro Conference on Software Maintenance and Reengineering, Florence*, 1998.
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. Sommerville, I., Sawyer, P.: Requirements Engineering - A Good Practice Guide, John Wiley, 1997.
5. Melchisedech, R.: "Investigation of Requirements Documents Written in Natural Language", *Requirements Engineering*, 3 (2), 1998.
6. Ryan, K. :"The Role of Natural Language in Requirements Engineering", *Proc. IEEE International Symposium on Requirements Engineering*, San Diego, January 1993.
7. Rolland, C., Proix, C. :"A Natural Language Approach for Requirements Engineering", *Lecture Notes in Computer Science*, Vol. 593, 1992.
8. Cyre, W., Thakar, A. :"Generating Validation Feedback for Automatic Interpretation of Informal Requirements", in *Formal Methods in System Design*, Kluwer, 1997.
9. Goldin, L., Berry, D.: "AbstFinder, A Prototype Natural Language Text Abstraction Finder for Use in Requirements Elicitation", *Automated Software Engineering*, 4, 1997.
10. Aston, G. and Burnard, L.: The BNC Handbook: Exploring the British National Corpus with SARA, Edinburgh University Press, 1998.
11. 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.
12. Garside, R., Smith, N.: "A Hybrid Grammatical Tagger: CLAWS4", in Garside, R., Leech, G., and McEnery, A. (eds.) *Corpus Annotation*, Longman, 1997.
13. Rayson, P., and Wilson, A.: "The ACAMRIT semantic tagging system: progress report", *Proc. Language Engineering for Document Analysis and Recognition (LEDAR)*, Brighton, England. 1996.
14. 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.
15. 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.
16. Sommerville, I., Sawyer, P., Viller, S.: "Managing Process Inconsistency using Viewpoints", IEEE Transactions on Software Engineering (to appear).
17. 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.