

Information Extraction for Reorganizing Specifications*

Krishnaprasad Thirunarayan
(Email: tkprasad@cs.wright.edu)
Dept. of Computer Science and Engr.,
Wright State University, Dayton, OH-45435, USA.

Aaron Berkovich, Steve Grace, Dan Sokol
(Email: {aberkovich,sgrace,dzsokol}@cohesia.com)
Cohesia Corporation, Dayton, OH-45402, USA.

Abstract: Materials and Process Specifications are complex semi-structured documents containing numeric data, text, and images, which are critical to materials, aerospace, and automotive industries. This paper describes the architecture, the detailed design and implementation of a coarse-grain extraction tool to automatically reorganize content to enable extraction of primitive specs using suitable criteria. The working prototypes were built in the context of Cohesia's existing software infrastructure, and use techniques from Information Extraction, XML technology, and Parsing.

Keyword: Data/Knowledge Management Techniques, Information Extraction Tools.

1 Introduction and Background

Materials and Process Specifications are basically presentations of numeric data with accompanying text and graphics that explain the quantitative information, which are critical to materials, aerospace, and automotive industries. A spec describes requirements on the processing of a material (alloy) in the mill, and the capabilities that the material should possess eventually. Specs are semi-structured documents with discernible organization and constrained vocabulary. Figure 1 shows a sample specification.

Cohesia Corporation created the Specification Definition Representation (SDR) as an ontology to articulate the semantic view of the components that comprise a spec, and capture user's interpretation of it. SDR is a tree-based declarative language for description (not a programming language for instruction). SDR introduced constructs such as **Procedures** to indicate boundaries for standards requirements

*This work was supported in part by NSF SBIR Phases I, II, and IIb Grants DMI-0078525 (1999-2002). The opinions expressed here do not necessarily reflect those of NSF.

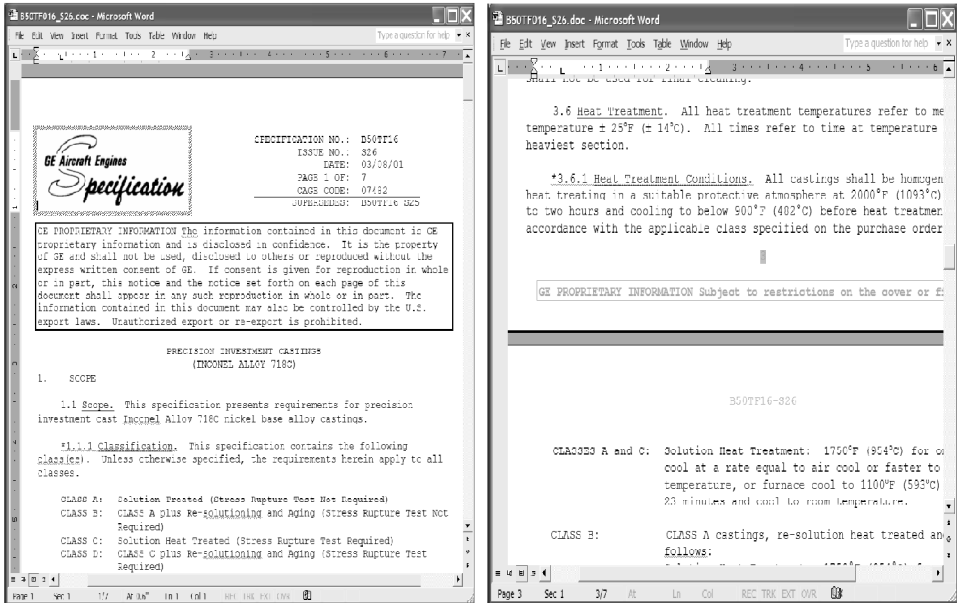


Figure 1: A Sample Specification Fragment

such as chemical composition, tensile test, melt method, etc. Procedures are composed of elemental Characteristics that describe the requirements that are essential for performing the associated process (e.g., carbon content, yield strength, minimum temperature range, etc). SDR also permits defining a common (standard) vocabulary of industry terms called the domain library, which includes lists of names for procedures, characteristics, controlled terms, units of measure, and their inter-relationships. SDR provides a consistent representation structure that allows retrieval, comparison, and combination of specs to drive manufacturing process. The SDR technology has been incorporated into a commercial software system called MASS (Management and Application of Specifications and Standards), and is in use at Fortune 500 companies.

Content extraction involves conversion of a spec into an “equivalent” formalized description in SDR. Manual content extraction of a spec can require one to several person days, depending on the complexity of the spec and the granularity of the extraction desired. To be able to combine and manipulate specs in practice, thousands of specs should already exist in SDR format, to serve as a foundation to build on. This is because: (a) Each product may have to meet requirements spelt out in several specs, which may in turn reference additional specs. (b) It may be that a vendor may choose to create a product spec that can satisfy a number of different customer orders (each having some leeway). (c) Or a spec may have been modified based on existing Practices, Agreements, Customer needs, or Vendor constraints. Thus, extracting 1000 specs can require approximately 3 person years! So, the cost

of data preparation can far outweigh the cost of the manipulation software, and the software is “useless” without the necessary legacy data. Thus, from business point of view, it is imperative that the workload of a human extractor be reduced to any extent possible, to minimize costs and improve revenue.

This paper describes the design details of a tool for semi-automatic recognition and reorganization of phrases in spec that are associated with requirements, which is viable for integration with the commercial product.

2 Related Work

The DARPA sponsored TIPSTER program and the Message Understanding Conferences (MUC) provided a major impetus to the progress in Information Extraction technology. Information extraction consists of defining the form of extraction rules, acquiring them (by hand-crafting or by machine learning), and then applying them [7]. In terms of the design space for extraction and transformation from semi-structured documents studied in [2], materials and process specs have the following “coordinates”: they have implicit structure, and are single documents with references; they are partly irregular and relatively stable, but can contain format and content errors. The work reported here resembles that described in [8, 9] in so far as they attempt to exploit domain specific background information to elicit the semantics, and use language processing tools to transform the spec as desired.

The spirit of our work is similar to what is described in [1, 3, 4, 5, 6]: In order to apply AI in realistic, large scale document processing applications, it is necessary to make explicit machine-processable semantics of sources. The research done in the context of *OntoWeb* is particularly relevant.

3 Detailed Design and Implementation of Extraction Wizard

The design attempts to solve the extraction problems by providing a set of tools for flexible handling of specs in different formats and content, and for refining extraction parameters. It deals with issues such as: (1) multiple layouts of spec content; (2) use of an English Domain Library to enhance domain library terms identification; (3) exporting to different formats for manipulation of the results in various packages; (4) table formatting; (5) image and extended character handling; and (6) tracing extraction results back to the original spec. The user interface provides the extractor with a workbench in which they can perform multiple extractions and tweak extraction parameters. For details, see [10, 11].

3.1 Separating Tables and Images from Text

To enable automatic processing of text while deferring tables and images for manual handling, the following preprocessing steps were used to obtain a spec in plain text form:

1. Identify paragraphs properly and replace non-ASCII characters in MS-Word file with their ASCII encodings by running a MS-Word macro.
2. Save the MS-Word file in RTF format.
3. Convert the RTF file into an XML file using tools such as IBM developer-Work's Majix (a Java application) that delimits text, tables, and images. In particular, it replaces an image with image tags and a reference to an image file, and preserves the structure of the table using table tags.
4. Finally generate a plain text file by removing image tags, and replacing table tags by appropriate indentation.

Specifically, MS-Word to ASCII converters were created: (1) to properly interpret paragraph breaking points and include additional line separators where needed, (2) to encode non-ASCII characters such as $^{\circ}$, \pm , etc into ASCII, (3) to preserve context for the embedded images, and (4) to properly format tables by aligning columns with column breaks in ASCII representation, for readability, and (5) to generate spec in plain text form that could then be used by the extraction utility.

3.2 Document Import Formats

The input formats of specs issued by societies such as SAE, ASTM, AMS, etc or by companies such as GE, Pratt and Whitney, Rolls Royce, Alcoa, etc can be gleaned from concrete examples and encoded in an XML-based **Format** file. The XML is used here for ease of parsing. The **Format** file is consulted to guide the search for spec header information and requirements, and to skip peripheral text.

3.3 Levels of Extraction

The text of a spec can be annotated to different levels of detail in order to make explicit mechanically processable information. To balance the commercial viability of the extraction task and its tractability, the following granularity levels were explored:

Basic Extraction: Basic Extraction involves the identification of header information such as spec name, spec title, organization, revision information includ-

ing revision date, etc, and filtering the spec text, to be subjected to further scrutiny later.

Level 1 Extraction: A note is a contiguous block of spec text. In Level 1 extraction, a spec is transformed into a (possibly, nested) sequence of implicated notes. The implication can contain either a single characteristic or multiple characteristics. The detailed requirements are still in text. (If copyright restrictions prohibit quoting, use references to section numbers.) These are cheap to produce and can be used to reorganize/filter a spec. Such extractions are used with MASS FAI (First Article Inspection).

Level 2 Extraction: Procedures group related requirements. A procedure can appear in a spec explicitly or implicitly through related characteristics. In Level 2 extraction, a spec is transformed into a sequence of procedures, with the spec text relevant to each procedure, structured as a sequence of notes, serving as the procedure body. This is finer grain than Level 1 but still has insufficient structure to allow numeric result reporting or machine manipulation and detection of conflicting requirements. Such extractions are further refined by a human extractor before being used with MASS Order Product.

...

Level * Extraction: The requirements expressed in numeric and symbolic terms are captured formally in a machine processable form in SDR to enable automatic analysis, combination and conflict resolution. At this stage of maturity, this level is the exclusive province of manual extraction by domain experts.

3.3.1 Extraction Utility

Extraction utility consists of three main components: the SDR objects, the extraction engine, and the formatter. The SDR objects represent the extracted SDR tree. The extraction engine contains the bulk of the functionality and is responsible for parsing the spec text into SDR tree. The formatter outputs the SDR tree in a form that can be imported into the Spec Editor.

The core of the extraction engine is implemented using four classes: **Extractor**, **Locator**, **DomainLibrary**, and **Tokenizer**. The **Extractor** object is responsible for orchestrating the entire extraction process. The **Config** file is read to determine the extraction type, file path settings, domain library search settings, etc. The detailed extraction is carried out with the help of **Locator** object, which has methods to recognize procedures, characteristics, and controlled terms (domain library terms) appearing in a piece of text. These methods are implemented on top of the domain library search engine. The **DomainLibrary** object implements advanced domain library search capabilities. It is intended to exploit information about synonyms, prefixes, suffixes, stop words, broader and narrower terms, etc in mapping a spec phrase into semantically equivalent domain library term [12]. In fact, the entire domain library can be viewed as consisting of the object domain

library (ODL) containing the core set of terms, enclosed by the English domain library (EDL) wrapper that incorporates flexible and robust matching algorithm to improve search.

3.3.2 Assorted Examples

We sketch specific situations that arise in the context of GE B-family Specs for illustration.

Structure using Section Numbers: It is important to recognize nested sections, and their section numbers in order to avoid mistaking numeric data with section numbers. For example, if the current section number is ‘3.1.2’, the next section numbers can be ‘3.1.2.1’ if it is a deeper section, ‘3.1.3’ if it is at the same level, and ‘3.2’ or ‘4’ if it is a shallower section [11].

Level 1 Extraction with Spec Classes: `Spec class` designators manifest themselves as upper case letters (A, B, C, ...) as follows: “B50T75A”, “B50T75A, E, F and G”, “Class G, H, and I”, “Classes A and C”, “P21TF7 Cl-A”, etc. However, upper case letters can also appear in other contexts, such as a spec reference “ASTM E 46” or as a (hardness) value such as “Rockwell B 40-75”, or for unit of temperature (F or C), etc.

To associate a Spec Class with a section or a paragraph, we use the following heuristic: Every (sub-)section is qualified on all Spec Classes named in section “Scope”. Explicit Spec Class references in a paragraph override the default qualifier. Otherwise, a paragraph inherits the qualifier from its left sibling (earlier paragraph), or transitively from its parent (enclosing (sub-)section). The rationale behind this approach is that, when the conditionals in a Level 1 extraction are evaluated against the given qualifier values, it should generate all applicable fragments of the spec.

4 Conclusion

Technical specifications of materials and processes are semi-structured documents with constrained vocabulary, which are crucial to companies involved in complex manufacturing and B2B E-commerce. In this work, we discussed a Visual C++ implementation of computer-assisted coarse-level content extraction tool. It was a challenge to scope the problem to make it tractable to the software developers while simultaneously ensuring that the results are useful in unburdening the domain experts. This work brought together tools and techniques from Language Processing, Knowledge Representation, and Web Technologies, and produced results usable by Content Extractors. Of course, what we have accomplished is just the necessary first step towards the long-term goal of automatic reorganization and summarization of semi-structured documents.

Acknowledgment: The authors would like to thank Steve Crowley and Prasanna Soundarapandian for numerous discussions throughout the project.

References

- [1] T. Berners-Lee, J. Hendler, and O. Lassila: The Semantic Web, Scientific American, May 2001 issue. (<http://www.sciam.com/>)
- [2] A. Crespo, J. Jannink, E. Neuhold, M. Rys, and R. Studer: A Survey of Semi-Automatic Extraction and Transformation, Information Systems, 2000.
- [3] F. van Harmelen and D. Fensel: Practical Knowledge Representation for the Web. In Proceedings of the Workshop on Intelligent Information Integration (III99) , 1999.
- [4] D. Fensel, J. Angele, S. Decker, M. Erdmann, H.-P. Schnurr, S. Staab, R. Studer, and A. Witt: On2broker: Semantic-based access to information sources at the WWW, In Proceedings of World Conference on the WWW and Internet (Web-Net99). 1999.
- [5] D. Fensel: Semantic Web enabled Web Services, Invited Talk at: In 7th International Workshop on Applications of Natural Language to Information Systems, June 27-28, 2002, Stockholm - Sweden.
- [6] J. Hammer, H. Garcia-Molina, J. Cho, A. Crespo, and R. Aranha: Extracting Semistructured Infomration from the Web, In Proceedings of the Workshop on Management of Semistructured Data, 1997.
- [7] I. Muslea: Extraction Patterns for Information Extraction Tasks: A Survey, In Proceedings of AAAI-99 Workshop on Machine Learning for Information Extraction, 1999.
- [8] L. Obrst, K.N. Jha, and G. Coen. Mass Change of On-line Textual Databases Using Natural Language Processing, In Proceedings of The 9th Symposium and Exhibition on Industrial Applications of Prolog, INAP '96, Tokyo, Japan.
- [9] L. Obrst and K.N. Jha, NLP and Industry: Transfer and Reuse of Technologies, In From Research to Commercial Applications: Making NLP Work in Practice, Eds: Jill Burstein and Claudia Leacock, Association for Computational Linguistics, pp. 57-63, 1997.
- [10] D. Z. Sokol et al: Computer Assisted Document Interpretation Tools, NSF Phase II Final Report (Executive Summary and Technical Details), 2002.
- [11] P. Soundarapandian: Information Extraction for Reorganizing Materials and Process Specifications, M.S. Thesis, Dept. of Computer Science and Engineering, Wright State University, 2002.
- [12] K. Thirunarayan, A. Berkovich, and D. Z. Sokol, Semi-automatic Content Extraction from Specifications, In Proceedings of 7th International Workshop on Applications of Natural Language to Information Systems, LNCS 2553, Springer Verlag, pp. 40-51, 2002.