SAND2000-0868C

RECEIVED APR 27 2000 OSTI

A Case Study in Modeling Company Policy Documents as a Source of Requirments

Kathleen Marie Crumpton
Department of Computer Science
New Mexico State University
Las Cruces, New Mexico 88003-8001
USA
kcrumpto@cs.nmsu.edu

Regina M. Gonzales
Klipsh School of Electrical and Computer Engineering
New Mexico State University
Las Cruces, New Mexico 88003-8001
USA
regonzal@nmsu.edu

Sharon Trauth
Sandia National Laboratories
Albuquerque, NM 87185-0535 USA
sltraut@sandia.gov

Abstract. This paper describes an approach that was developed to produce structured models that graphically reflect the requirements contained within a text document. The document used in this research is a draft policy document governing business in a research and development environment. In this paper, we present a basic understanding of why this approach is needed, the techniques developed, lessons learned during modeling and analysis, and recommendations for future investigation.

The modeling method applied on the policy document was developed as an extension to entity relationship (ER) diagrams, which built in some structural information typically associated with object oriented techniques. This approach afforded some structure as an analysis tool, while remaining flexible enough to be used with the text document. It provided a visual representation that allowed further analysis and layering of the model to be done.

INTRODUCTION

Determining requirements is a daunting task demanding in-depth knowledge of the environment for the requirements to effectively represent an issue or problem. Loucopoulos and Karakostas (Berry, 1999) describes Requirements Engineering as a "systematic process of developing requirements through an iterative cooperative process of analyzing the problem, documenting the resulting observations in a variety of representation formats, and checking the accuracy of the understanding gained." Dr. Berry points out that the problem in establishing requirements is that requirements always change. As peoples' perceptions change and more information is gathered, requirements need to be adjusted. Generally, people are reluctant to deal with requirements because they simply don't want to deal with the chaos (Berry, 1999).

There are always many sources of requirements, irrespective of the kind of product under development. These can include regulatory requirements, quality requirements, interface requirements for existing systems, technical requirements for functionality, and user requirements -- to name just a few. This information can come in different forms, many of which are textual. Clearly, these various sources of requirements are not written at the same level of abstraction. In fact, a single document can, itself, contain requirements at several abstraction levels. Individuals attempting to enter text into a requirements management tool may discover that several sections of a source document will cover facets of the same topic, necessitating some way of grouping or merging the thought behind the words. It is, therefore, imperative that a consistent method for modeling requirements from such sources be used, especially in critical systems, so that requirements can be analyzed, synthesized, and addressed effectively.

This paper focuses on a particular type of text-based requirements documents. Specifically, we present the approach developed and the results obtained in analyzing requirements in a policy document that was under development and evolving. The methodology in this paper has also been successfully applied to a standards document and to a subset of a collection of user-specified needs. Our experience shows that simply representing requirements' information graphically is not sufficient for adequate human use and interpretation. Although not intentionally complex or ambiguous, human thought

DISCLAIMER

DISCLAIMER

transferred to text on paper can lead to documents that are inherently ambiguous and difficult to read. This can be true even when an outline is used prior to writing the full text. This makes it difficult to know for sure that one has met the intended requirements contained in the documents. Representing such information graphically in a layered model can provide a mechanism for semi-formal verification of written requirements. Customers and developers can more readily understand the requirements and the relationships between them by studying the graphical model.

BACKGROUND

This paper covers a portion of the work done using this methodology. Originally, the project task was to develop a modernized engineering process that could be readily used by anyone in the technical community in their daily work. It was desirable to understand the programmatic and regulatory requirements with which the process had to comply.

As the project evolved, it became apparent that, to be used, the process had to be easily implementable in a desktop engineering environment. Further, it was recognized that managerial support and advocacy was essential if we were to realize widespread use of the process. To address the need for management support, an initiative was launched to create the policy behind the process.

By their nature, policy documents are typically at the highest level of abstraction in a comprehensive system of doing business. Further, policy documents typically specify only the "what's" that must be done, leaving the "how's" to the process. Thus, as the policy specification emerged, the process group needed to be sure their process specification would be consistent with the emerging policies. In the spirit of concurrent engineering, both the policy and process development proceeded in parallel, with many of the same individuals working on both projects.

The modeling approach discussed in this paper was intended to provide some specific benefits. First, it was desired to identify specific policy statements related to the process so that the process being developed could be verified for its compliance with the policy statements. Second, it was hoped that by modeling the policy, we could identify any gaps or inconsistencies within the policy structure that would have to be addressed. As discussed below, both these expectations, as well as others, were met.

CASE STUDY

The policy document that was the focus of our attention was under development for Product Realization in the Nuclear Weapons Strategic Business Unit (NW SBU) at Sandia National Laboratories. A small committee developed this document, such that the policy is part of an overall structure. Policy development is an on-going activity as of the writing of this paper.

As noted earlier, preliminary work had already been done in developing an approach to modeling a more static regulatory standards document. We began by applying this ER-based approach to our draft policy and investigated other viable approaches for potential comparison. Many of the other possible approaches were either based on ER diagrams (as ours was) or were more suited to be used during brainstorming sessions prior to creating the document (Robertson, 1999).

Since the majority of the initial effort in our modeling methodology was simply parsing the sentences contained in the document before we could determine abstraction layers, we continued using the modified ER diagrams. We wanted to work with the detailed text in the policy and did not want to draw simply from the headings in the document and fundamentally reproduce the outline structure. We felt it was important to understand the connectivity of the information contained within the document, since we hypothesized that documents have inherent overlaps and inconsistencies when written without modeling first. We expected this to be true for this case, especially since this was an evolving document.

We created an initial raw ER model and analyzed the network of nodes and connections so that structures could be abstracted, synthesized, and suppressed as necessary. In this way, a layered model emerged that was easier to understand and that clarified important entities and interfaces within the document.

FACTORED ER MODELS

In order to create our initial models, we analyzed the policy document sentence by sentence. Nouns, verbs, and adjectives were identified and used to parse each sentence into a graphical entity relationship (ER) model. Within the model, nouns are represented as nodes. Nodes are linked together by verbs. A node may have adjectives listed, attached as attributes, if deemed appropriate. In this particular document, however, there were not many adjectives included in the text to describe a noun.

Two characteristics of this document made modeling activities unique and inherently complex. First, the policy was still under development. Second, we found it necessary to rely on a domain expert for interpretation of the text to fully specify the model. This was necessary given the level of abstraction of the policies and their intended strategies. We had access to a domain expert who could interpret for us the intended meaning whenever the meaning was ambiguous. To illustrate why interpretation became important, consider the following statement from the configuration management portion of a June 1999 version:

Management of issues, lessons learned, and recommendations resulting from support and evaluation activities is required to assure that issues are addressed.

Figure 1 is a representation of the purist approach to modeling from the document. Represented in figure 2 is a model of the same sentence after the domain expert evaluated the purist model. Notice how the concepts of evaluation activities and support activities inter-relate with issues, lessons learned and recommendations. The graph in figure 1 is a directed tree structure, whereas that of figure 2 becomes more of a network structure. Besides the simple relationships that were clarified by the domain expert, some questions were posed like, *who manages* and *what are the support and evaluation activities*. This information is intentionally left obscure to be defined further in the processes that implement the policies.

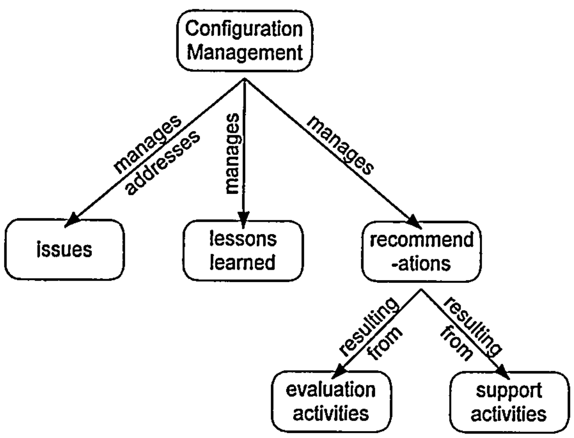

Figure 1. Modeling straight from the text

This is a simple example, but in general the domain expert is able to review the graphs generated by parsing the text and provide feedback regarding the strategic intent, assumptions, and ambiguous phraseology. As part of the interaction with the domain expert during the modeling we added brackets around nouns and verbs to represent the idea that something was inserted in the model that the corresponding document did not specifically include. The domain expert always approved these insertions.

ANALYSIS AND LAYERING PROCESS

The goal of the layering process is to gather clusters of entities that can be suppressed as part of other super-entities, while at the same time, to highlight the significant entities, giving an accurate representation in a synthesized global model. It is here where we borrowed from object oriented approaches (Coad and Yourdon, 1990). We analyzed the connectivity of entities and evaluated graphically the coupling and cohesion of the entities. Part of the process included understanding the significance of certain entities. We replaced clusters of nodes, where possible, with a single 'super' node. Sometimes we absorbed nodes into another entity such that the resulting super-entity was described by a structure of its own.

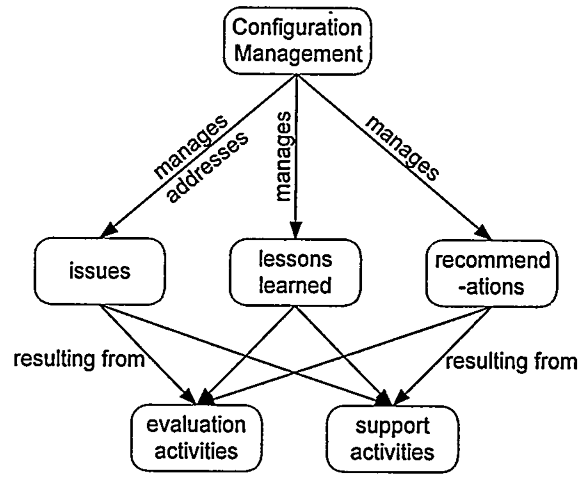

Figure 2. Model with Domain Expert Intervention

During this phase, we looked for structures that represented assemblies and inheritance, focusing on 'part of' and 'is a' relationships. In this way, we iteratively developed a simplified diagram, and a high-level entity relationship diagram, evolved where the relationships were between structures (figure 3). At this point our models were complete, and each structure, as well as the general model, was individually transferred to a single page.

STRATEGIES APPLIED

Orthogonal Entities. We observed that there were two orthogonal sets of entities that had complex relationships to one another. These orthogonal entities were contained in the document as major heading and subheadings. Figure 4 conceptually depicts the complexity between these relationships. We found that in order to accurately model the complex interfaces represented by these entities, we needed to work with the whole of the document. This means that we could not layer the model based on one or the other set of entities shown in figure 4 and then combine, since interrelationships vital to whole would be buried.

Tagging Entities. We also tagged significant nodes

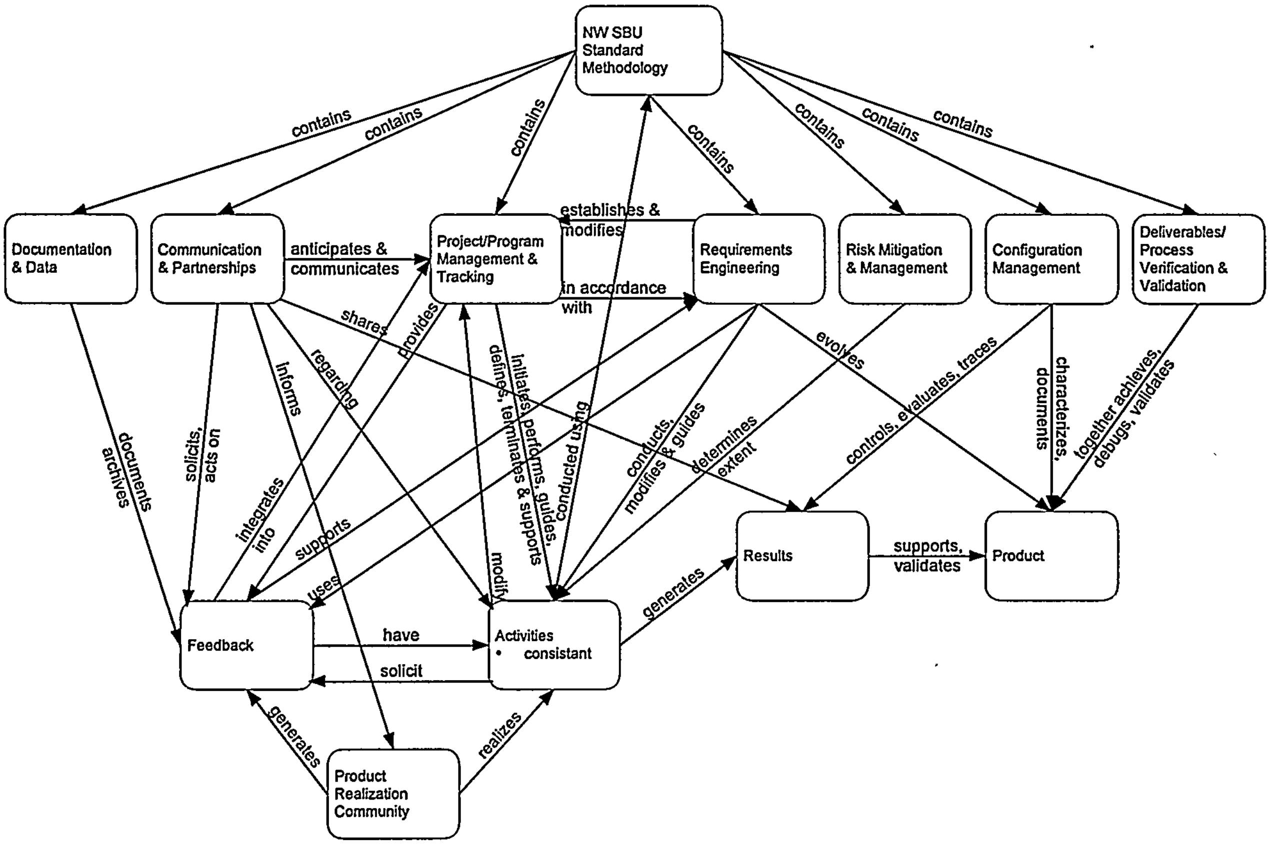

Figure 3. Highest Abstraction Layer of the Final Model

using a hexagonal icon to indicate significance (figure 5). This included the entities along the x-axis of figure 4, referred to by the domain expert as the crosscutting processes. These entities included, Communication & Partnerships, Project/Program Management & Training, Requirements Engineering, Risk Mitigation & Management, Deliverables/Process Verification & Validation, Configuration Management, Documentation & Data. The nodes representing these entities were ones we did not want to become subsumed while we were conducting analysis and layering. Our strategy was to provide a way of moving structures around more freely while minimizing confusion in the graphical representation. Another form of tagging included using color to tag the significant entities that were now being represented as structures or 'super' nodes allowing the attention to be focused on the entities that were singleton entities, those that were on the highest abstraction layer but were not super-entities. These entities were candidates for new structures to be

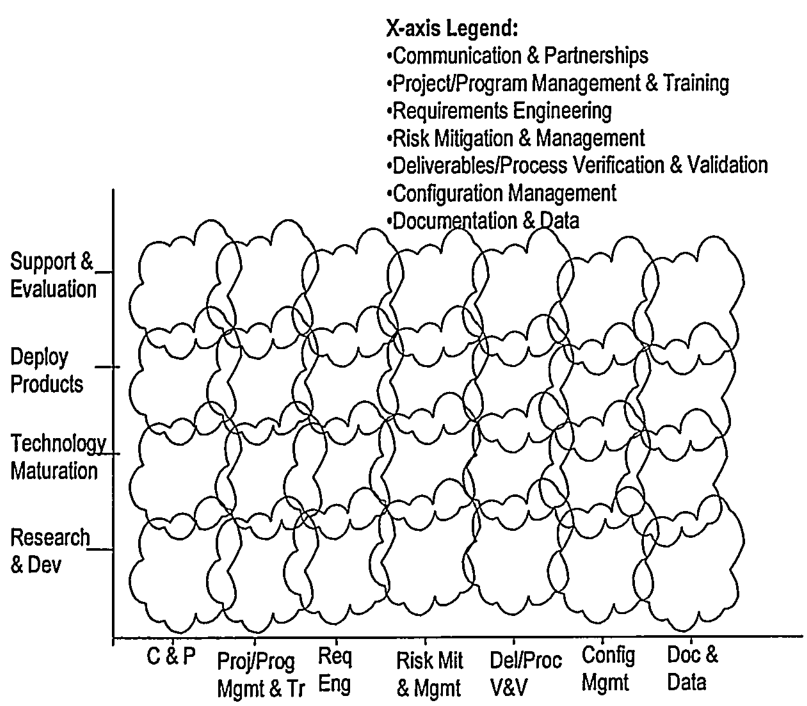

Figure 4. Conceptual Framework for Policy Information

created.

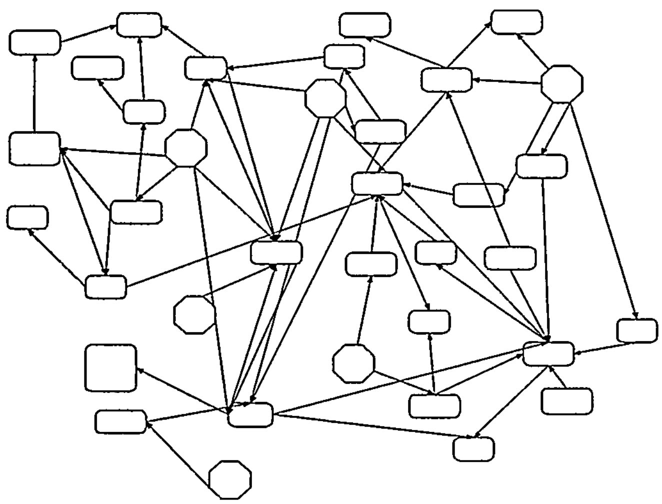

Figure 5. Graph in Development

Incidence Frequency. Another strategy applied to evolve interfaces between the two orthogonal entities was to generate an incidence frequency table for entity nouns (table 1). Those nodes having a relatively high frequency were identified as being possible candidates for consolidation in the graphs.

Noun	Freq	Noun	Freq
requirements	7	Defects	3
cost	4	Review	7
activities	9	Develop.	3
results	5	Research	4
support	3	Form	3
programs	3	Process	5
evaluation	3	Technology	10
system	4	Status	3
product	13	Partner	2
inform.	6	Project	3
risk	6	Plan	4

Table 1. Incidence Frequency

Using the incidence frequency, we repeatedly tagged the entities containing certain nouns. One such entity was 'activity.' Each activity node was colored for initial identification. Subsequently, all such nodes were consolidated to form a new 'activity' structure. Sometimes we had to identify connections and make appropriate substitutions without changing the original meaning of the model. For instance, our analysis revealed a connection between 'ability' and 'results,' so we changed 'ability' to 'usability.' Since 'usability' appeared on several occasions, and making this adjustment preserved the original meaning, this slight adjustment allowed us to make a consolidation in the graph.

Iteration. On occasion it was necessary to back out of the suppression of clusters when it obscured meaning at a higher abstraction level. Examining the words and talking out loud were part of the intense effort to understand and consolidate without taking away meaning.

Nouns were chosen and discarded for several reasons. The primary basis for making a structure was the strength, significance, or ambiguity of the noun. Sometimes it became apparent that the node was weak. Weak nouns are those that can not stand on there own. An example of a weak noun was technology. Even though it appeared ten times in the document, it existed in the outermost parts of relationships and was, therefore, ineffective in simplifying the structure of the model. In some cases, nouns existing only in the extremities of the model were also small quantitatively, so that they were insignificant in the analysis, as well. Other nouns did not form strong relationships and were ambiguous, leading to modifications based on discussions with domain experts.

In making judgements regarding what noun to build a meaningful structure around, we were looking for nodes that would stand out as germane, or terms that would offer some insight into the interface between the orthogonal entities. For example, when we pulled out the structure for activities, we found that there were *Research and Development activities, Technology Maturation activities, Deploy Products activities, and Support and Evaluation activities.* The structure created for 'activities' formed a key interface between the two sets of orthogonal entities.

Initially, the term 'requirements' looked like a candidate because there was a reasonable amount of redundancy, and it suggested a tighter coupling between elements. But further analysis revealed that it was too embedded within the model and, therefore, counterproductive for our goals. In other words, it convoluted the graph while providing little insight about the interaction between the orthogonal entities represented in figure 4.

Once super-entities were created, the links between the super-entities had to be examined for continuity and clarity. Many links represented by verbs could be discarded based on the strength of the other verbs. For example, in a link between 'activities' and 'product life cycle,' we were able to remove the verb 'has' because another existing verb 'integrates early' already had 'has' implied. Many times verbs like 'has' and 'creates' stood out as superfluous verbs and could be absorbed into stronger verbs that connected the same nodes –verbs like 'documents' and 'manages' could provide a more meaningful relationship that also embodied the more simple or implied relationships.

ANALYSIS OF THE FINAL MODEL

Once the synthesized, global model was generated, we examined the model for insights about the policy being generated and the associated programmatic structure. In the final highest layer of the model (figure 3), we observed some clear interconnections between the crosscuts indicated along the x-axis of figure 4. For example, the link between the nodes, Requirements Engineering and Program/Project Management and Tracking shows up heavily and understandably, while the in-depth technical linkage between Requirements Engineering and Product/Process Verification and Validation does not. The same is true regarding the need for stronger interconnections between Configuration Management, Requirements Engineering, and Verification & Validation. For the work in the NW SBU, the activities just mentioned form the heart of a technical approach to assuring that the product is really what is wanted and will meet the customers' needs. The review of the synthesized graph revealed that since Requirements Engineering, Configuration Management, and Product/Process Verification and Validation must work tightly together if they are to be effective, perhaps more explicit links should be addressed within the policy document.

LESSONS LEARNED

Orthogonal Entities. It was clear that there were two orthogonal sets of entities as shown in figure 4 that had complex relationships with one another. We first tried to project one set of entities onto the other set and then combine along one axis of entities. This approach actually resulted in suppression of the critical interfaces between these entities. After exploration and discussion, we abandoned our initial attempts with this approach and used the entire raw model as the basis for analysis and layering.

Changing Documents. At the time analysis was being conducted, the policy document was under frequent review and revision. As a result, progress with the modeling activities was difficult. Our initial efforts were largely directed at keeping up with modifications as opposed to analyzing a static document. This was positive in the sense that we provided feedback into the revision process, but also frustrating, and at times, confused our attempts at modeling and synthesis.

Strategy Considerations. Some surprising and important issues came out in the analysis process. One example is the relationships, similarities, and differences between Technology Maturation and Research and Development. Technology Maturation has a smaller scope than Research and Development, but the two are strongly related. The focus of Technology Maturation is to assure that a given technology should be matured by review against a collection of requirements to determine whether the risk associated with its use is low enough to warrant its incorporation into a product. If the review indicates otherwise, the product, itself, has to mature the technology during the associated development effort. This latter approach can make a difference in an estimated lead-time to turn out the product, and the associated risks to the product can increase dramatically. Initial attempts to synthesize the graphical model of the policy documents included recommendations to merge the Technology Maturation and the Research and Development functions into a single entity. Discussions with the domain experts revealed that such a simplification would loose a significant strategy of the policy structure itself – namely to focus on technology maturation as a separate and distinct process. Further development of the policy is required to delineate the crucial role that Technology Maturation plays so that it is not treated casually.

Roles. The roles that people played in modeling the policy document were another important outcome in this study. It was found that a modeling expert could successfully drive the analysis process. Although our process contained rules and elements conducive to repeatability, modeling as described in this paper is still something of an art that is perfected with experience. The massaging of the data and getting the intuitive feel of how things should go are skills that are not yet taught, but passed on to others through experience over time. In this case study, the modeling expert also had some domain experience and was able to draw from that. But, in general, the modeling expert is one that draws out pertinent data from domain experts and then converts this information into an accurate representation of that data.

As mentioned earlier, of particular note was the role of the domain expert in both analyzing and approving models of the policy document. In contrast is the situation we experienced in modeling a draft of ISO 9001 (a standards document). Although also in draft form, the ISO 9001 document was more structured and stable, and there also was no domain expert available to assist in the analysis.

In a third situation, that of a document listing user interview information, strategy and intent were not present as they were in the policy domain. While this user data was clearly the most flexible, there exists no single domain expert in this situation to represent the underlying intention behind the users' statements. It was desired to obtain a true essence of the users' statements, so that assuring that user requirements were more faithfully represented was an essential objective of the analysis. Thus, based on the type of document being analyzed, the role of the domain expert can provide essential input to the analysis.

Also important to this analysis process was the role of the average engineer – to observe the process that was unfolding, to recommend alternative ideas for the analysis, and to ask questions. In this case study, this person had no prior experience in a project of this nature. This lack of experience was not particularly troublesome; in fact this person was more able to probe and offer an unbiased point of view that sometimes people with experience might not have.

CONCLUSIONS

The work described by this paper not only assisted in identifying requirements specifically associated with the process work under development, it provided policy developers a framework to determine the completeness and sensibility of the policy structure and associated programmatic strategies. Basically in real-time, the development team was able to review text that was particularly problematic or ambiguous, and to determine a more effective way to re-word the text. This resulted in text that could be more easily understood and that would reflect the desired intent.

Additionally, the inclusions of review and discussion with the domain expert enabled the determination of policy strategies that could likely be mistakenly compressed or even edited out of the text, inadvertently.

An additional benefit was also obtained from this process. When the final synthesized graphical model was reviewed, it was clear that interrelationships between policy elements had not been explicitly articulated in the draft document. This insight was available to reviewers at a glance, rather than through many hours of intense document review and discussion, or rather than being overlooked completely. Policy developers were then able to determine whether the interrelationships were better elucidated in the "what's" called out in the policy, or left for the processes to define the specific "how's."

In this particular case study, we observed that the inclusion of the domain expert produced more of a model of intent, as opposed to one that would be 100% true to the written text. While this enabled synthesis of the model, it was somewhat disconcerting to more experienced modelers. This intent versus literal 'war' of sorts may actually be characteristic of this particular type of case study. As was mentioned, the policy effort evolved as an outgrowth of the process development effort that preceded it. The policy development team contained personnel who also participated in the process development effort. The policy development team intentionally included provisions for many of the "how's" that were created in the process effort – namely the crosscutting process functions.

As such, it is certain that the policy effort did not begin at ground zero as if no analysis had ever been conducted. When the policy statements were written, only essential details were included. The details specified were those directly related to a strategy of specifying environmental changes for which a movement along a continuum of outcomes was desired. Thus, the policy was the result of both some pre-analysis and consciously chosen omissions of detail. This type of approach to policy specification may naturally result in the need for a domain expert.

FUTURE RESEARCH

The approach discussed in this paper is an attempt at structuring and establishing a process for the difficult tasks of analyzing, synthesizing, and simplifying requirements statements. The positive results obtained in this and related efforts indicate some natural areas for follow-on research.

Other modeling approaches need to be examined further and compared. Other graphical and natural language approaches to examining text documents should be compared with regard to advantages and disadvantages.

Automation is also a goal. Applying a natural language tagger, used in much of the research in Natural Language Processing (NLP), to the textual document seems very feasible. Once a document has been tagged, a first cut graphical rendering of the document could theoretically be produced automatically. However, synthesis and simplification would probably require interactive intervention by the analyst, perhaps using intelligent agent technology. Once an initial ER rendering of the document is created, another tool could be developed based on graph theory that assists in identifying clusters and entities that could replace several nodes, i.e., part of this automated process could include suggestions for possible structures.

Another area for research is that of alternative approaches to text policy documents or formats for standards documents. It is possible that the graphical representation could be more easily integrated into the human mind through web-based systems, resulting in more readily assimilated documentation. Such graphical representations could assist not only human learning, but also implementation, traceability determination, and audit interpretation.

Realistically, however, the analysis and graphical abstraction process presented in this paper will support such automation and application only as it becomes more formalized. Progress is being made in finding methods that represent data accurately and in a usable, meaningful way. But, as yet, there exist no easy solutions (Berry, 1999). Only further application and study of this promising technique will progress it beyond its current state of being largely an art form.

ACKNOWLEDGEMENT

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

BIBLIOGRAPHY

Berry, Daniel M., "Requirements Engineering Requirements Iceberg," Presentation given at New Mexico State University, Computer Science Dept, August 11, 1999.

Coad, P., and E. Yourdon, Object-Oriented Analysis, Prentice-Hall, 1990.

Robertson, Suzanne, The Atlantic Systems Guild, "Requirements Patterns via Events/Use Cases", http://www.atlsysguild.com/GuildSite/SQR/Requirements_Patterns.html, June 1999.

Schvaneveldt, Roger W., "Pathfinder Associative Networks: Studies in Knowledge Organization", Ablex Publishing Corp, New Jersey, 1990.

BIOGRAPHY

Kathleen "Kit" Crumpton is a graduate student at NMSU where she is working on a Masters Degree in Computer Science. She currently holds a Bachelor of Science in Computer Science with a minor in Mathematics from NMSU and a Bachelor of Science in Administration of Justice from American University in Washington, D.C. She has 12 years experience in business software applications and has worked in the manufacturing community, telecommuniations and the banking industry.

Regina M. Gonzales is on the faculty at New Mexico State University. She has over ten years experience working full-time in industry as a Computer Systems Engineer and Project Manager. Regina also consults for several companies including Sandia National Laboratories and Insight Technologies Group. She has a B.S. in Electrical and Computer Engineering form New Mexico State University, an MS in Electrical and Computer Engineering from University of Arizona, an MS in Computer Science from University of Colorado, and is finishing up a PhD in Requirements Engineering from New Mexico State University.

Sharon Trauth has worked at Sandia National Laboratories for over 15 years and is a manager in the Engineering Business Practices and Information Center. She has extensive experience in many aspects of the systems development lifecycle including quality engineering, component design engineering, manufacturing processes, and failure analysis. She has applied this knowledge to her recent assignments of developing a standardized engineering and lifecycle process as well as the associated policy infrastructure. She has a Bachelors degree in Physics and a Masters in Electrical Engineering from Louisiana State University.