NDAS: The Nondeterministic Agent System for Engineering Drawing Analysis

Tom Henderson and Lavanya Swaminathan,School of Computing, University of Utah
Salt lake City, Utah 84112, USA, tch@cs.utah.edu

Abstract – We describe the use of nondeterministic agent systems (NDAS) to achieve a more flexible system for technical drawing analysis. They are called nondeterministic because the agents explore alternative parts of the solution space simultaneously, and every agent works to produce some result which may or may not contribute to the final result. (Note that this is a form of *speculative parallelism* [1]; this can also be viewed as a distributed blackboard system [15].) The final result derives from only a subset of the work put in by all the agents. We explore nondeterminism in this problem domain since deterministic systems usually make irrevocable decisions (e.g., threshold selection) that eliminate possible solutions. The technical drawing problem domain contains many factors that vary with the drawing: thresholds, text fonts and size, noise levels, etc., and this variation makes it interesting to explore the possible solution space dynamically and in a breadth-first way.

1. INTRODUCTION

The semantic interpretation of industrial drawings is a very difficult problem. Tombre summarizes the state of the field for CAD drawings [11] and as a document image analysis problem [12, 13]. An earlier overview by Haralick [8] gives insight into the scale of the technical drawing problem (e.g., about 250 million drawings are generated annually), as well as a discussion of performance measurement. A major problem pointed out by these reviewers is that there is a dearth of work relating high-level models of documents, drawings or graphics to the various levels of analysis. The goal of technical drawing analysis is to interpret the contents of an image, such as that shown in in Figure 1.

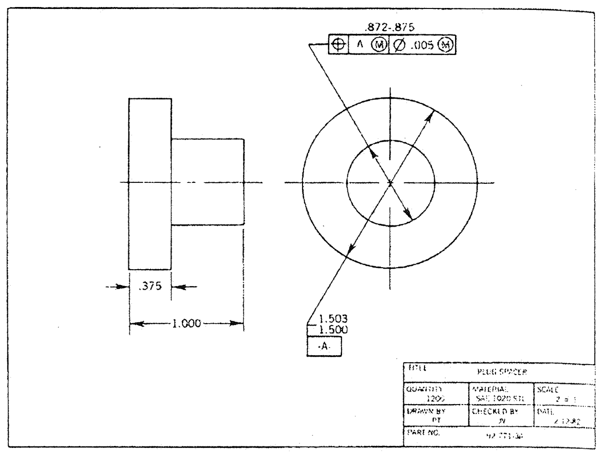

Figure 1. Part of a Scanned Technical Drawing

In the standard form of analysis, the scanned drawing is digitized, noise is removed, text and graphics are recognized, graphics is vectorized, dimensions are extracted and the whole drawing is analyzed by applying knowledge rules. Each of these steps normally uses just a single set of thresholds. Our goal is to allow a wide range of thresholds to generate a potentially large search space (hopefully including the correct components), and then to apply domain constraints to prune the space.

A High Level Model for Technical Drawings
We define the layout of the technical drawings in terms of a structural grammar (see [10] for more details on this approach). The components of a technical drawing, such as *text*, *pointer arrows*, *dimensions*, etc., are defined as either terminal symbols in the grammar (e.g., *line segments*), or as nonterminal symbols (e.g., *dimension description*) defined in terms of rewrite rules that describe their sub-components and the relations that must be satisfied between the sub-components.
The combination of the structural approach and nondeterministic agent analysis is a natural fit and provides

KIMAS 2003, October 1-3, 2003, Boston, MA, USA.

s to control the exploration of the exponentially
arch space.

2. AGENT ARCHITECTURE

are independent software processes with the following properties: autonomous (react to environment), ate (beliefs, commitments, etc.), persistent (never terminates), can communicate (send and re- essages related to effort), and perform some ac- ave abilities to analyze and create data). For mplete accounts, see [9, 14].

it architecture is a software architecture for de- naking with intelligent (flexible) processes em- within it. The agents may be proactive or re- and should cooperate (including communicate) ve a goal.

lore the use of nondeterministic agent systems to achieve a more flexible system for techni- ing analysis. They are called nondeterministic the agents explore alternative parts of the so- pace simultaneously, and every agent works to some result which may or may not contribute nal result. The final result derives from only of the work put in by all the agents. We ex- ndeterminism in this problem domain since de- tic systems usually make irrevocable decisions reshold selection) that eliminate possible solu- he technical drawing problem domain contains ctors that vary with the drawing: thresholds, s and size, noise levels, etc., and this variation interesting to explore the possible solution namically and in a breadth-first way.

onstrate this through the design and analysis DAS system and provide experimental results rt the claims. (See [10] for more detail.)

to organize the activity of the analysis agents, developed an engineering drawing model com- structures typically found in such drawings, ions between those structures. This approach on structural and syntactic shape methods [7]); however, our method is novel in that it r the natural application of the NDAS agent recover the desired structures from an image.

structures correspond to the terminal sym- shape grammar, and include: **text**, **box**, **ray**, **pointer_line**, **pointer_arc**, **circle**, **ment**, and **graphic**.

vel structures correspond to the nonterminal n a shape grammar, and can be described by ules which define sub-structures which com- new structure, and the relations that must veen the sub-structures.

The issues that we have explored using the NDSA approach include:

- *The analysis of technical drawing annotations*: This includes the ability to recognize dimensioning, features and their annotations, tolerances, and references to nomenclature.
- *threshold sensitivity analysis*: The clear determination of the relationship between a change in threshold and its impact on the analysis process and result.
- *precision, robustness and performance analysis*: An engineering account of these aspects of a system is extremely useful in making cost performance tradeoff decisions. Most results in the literature do not define very carefully what it means for a system to work nor what error is.

Annotation models have been presented in the literature (e.g., see [3, 5, 4, 6]), and we have extended these ideas and developed a novel approach to high-level modeling. We have also implemented the basic image analysis tools to extract text, graphics, graphical primitives. These form the basis actions for the agent architecture approach. **The thesis of this work is that a structural analysis model may be realized through a set of software agents acting independently and in parallel to ultimately produce a coherent analysis.**

We demonstrate this through the design and analysis of the NDAS system and provide experimental results to support the claims. We have explored the:

1. organization
2. communication, and
3. higher-level modeling capabilities

of nondeterministic agent systems (NDAS) using the analysis of technical drawings as the application domain. Figure 2 shows the organizational and operational view of the technical drawing analysis agent system.

The a_i, a_j and a_k's are agents which look for certain kinds of objects in their field of action, and when the appropriate conditions hold, the agent will act and produce another set of objects. These objects may be desired data objects (e.g., a segmented image) or a message to enlist new agents (e.g., to produce a better analysis). The idea is that the agents work independently and in parallel and continue to work until conditions cause them to become quiescent.

Such an approach allows for feedback loops as previously processed data may reappear after modification

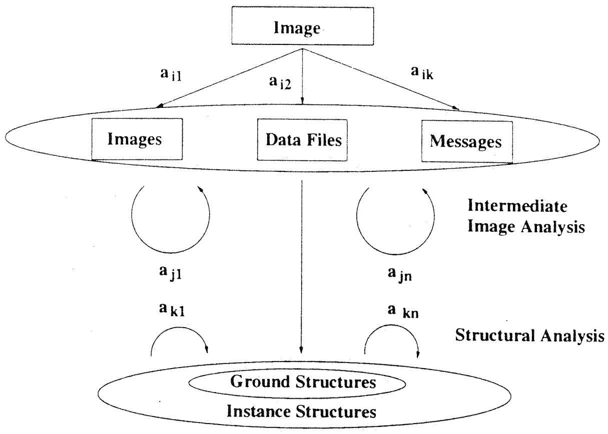

Figure 2. Nondeterministic Agent System for Technical Drawing Analysis.

by a higher-level process. This also allows a coarse to fine improvement as more work of broader context is completed. Agents may also monitor the activity of other agents and measure performance or provide data for consumption by the (human) user. Of course, the user may participate as an agent as well! (Note that it is straightforward to define the agents so as to capture the standard processing paradigm.) This approach is based on distributed agents and files, and avoids centralized control or bottlenecks through centralized agents which record and relay information.

A similar structure was proposed by Carriero and Gelernter [2] and called a process trellis. It was a hierarchical graph of decision processes, from low-level to high-level and with others in between. All processes executed continuously and concurrently. This was an example of their specialist - parallel problem approach. The big question was efficiency. The process trellis clearly allows massive parallelism and scales well. Some differences with what we propose are that the trellis:

- has dataflow-like streams of input and output; pass data up and query down,
- there is a master and n worker processes, and
- the goal was a real-time algorithm for medical equipment control.

3. NDAS ORGANIZATION

The nondeterministic agents are a collection of independent Unix processes. There are two aspects of agent organization: internal and external.

Internal Agent Organization

Each agent is written as a C++ program, which watches its current executing environment (directory) for the existence of certain files or processes and then takes actions as specified by its internal code. This can be described by the following set of actions:

1. *Monitor*: watch for the existence of files, processes, etc.
2. *Action Program*: Executable actions to take.
3. *Wrapup*: Cleanup files, check termination conditions, etc.

External Organization

The organization between agents includes how they announce their existence, if necessary, file protocols, etc. In addition, this includes the variety of agents that the user may wish to define; e.g., facilitation agents, performance analysis agents, process control agents, image analysis agents, feature analysis agents, higher-level model agents, etc.

Communication between agents is also an issue of their organization. We propose to have them communicate through files; for example, each agent will create a file which describes itself and its internal organization at a high level. These files are created as ASCII text files. A message protocol has been defined. This includes the medium (files), the syntax (NDAS Query Language – NQL), and semantics (program actions).

At a minimum, each communication includes:

1. sender ID,
2. receiver ID (may be broadcast or single recipient),
3. language (ASCII or binary),
4. ontology (defines the meaning of the syntactic expressions), and
5. file name (where message can be found).
6. history (names of all the agents which have played some role in giving rise to this message).

In general, an ontology is similar to a database schema and gives a specification of the objects, concepts and relationships that exist in the domain of interest. For NDAS, this has been defined by means of a semantic network. Concepts are represented by objects and relations between objects; these objects for the technical drawing analysis are called structures, and can be either structures (terminals) derived from the image, or model structures (nonterminals) which arise in the semantic model definition.

4. SEMANTIC NETWORKS AND AGENTS

Higher-level models involve descriptions of the semantics of the drawings, and as such involve determining the relations between the primitives of the drawing and their meanings. We use graph models (semantic nets), which are explored through a grammatical paradigm. After the document is digitized, vectorized and the connected components extracted, the result is a set of image primitives such as segments, arcs, arrows and text blocks. Based upon the relationships that exist between these primitives, a semantic net model is used to recognize important features like dimensions, annotations, and legends in the document. (Note: a semantic network may represent knowledge to be used to perform the low-level image analysis as well.) The relational structure of technical drawings and the analysis of digital images of such drawings are driven by a semantic network. This constitutes the high-level model. The image is analyzed for text, geometry (2D and perhaps 3D), regions, boundaries, and relations between the extracted objects. As an example higher-level structure, consider a *box_description* which is comprised of a *text* structure and a *box* structure, where the *text* is inside the *box*.
To determine how well the structural analysis is performed, we applied NDAS to the several engineering drawing images; for details on the analysis, see [10].

5. CONCLUSIONS/ FUTURE WORK

The results of these experiments are encouraging. The structural analysis proceeds correctly, and the agent system seems to be robust and explores much of the interesting part of the search space. The percentage of dimensions found ranges from 16.6 % for noisy images to 100 % for clean images (e.g., see Figure 3), and the pruning methods lead to orders of magnitude reductions in the number of symbols considered during the analysis.
We have demonstrated that the nondeterministic agent system (NDAS) combined with a complementary structural modeling approach can achieve a coherent analysis of annotations in technical drawings. This process can be made more efficient by means of symbolic pruning as introduced here. The experimental data indicates that this is a feasible approach and gives a firm basis upon which to define error, precision and performance measures.
In developing and studying our approach, we have raised several issues which require further study:

1. Agent System: A prototype agent system exists and it may be useful to re-implement the system in a commercial (or industrial strength) agent system. Other types of agents need to be studied, including performance monitoring agents, resource utilization agents, top-down analysis agents, user interface agents, etc.
2. Image Analysis
   - Image Codes: Currently, we have devised most of the algorithms for image analysis. If we have better or more than one algorithm (for the same purpose), the performance could be improved.
   - Time: We need to device better output reduction techniques to reduce the overall time taken by the image analysis NDAS system.
3. Structural Analysis
   - Symbolic Pruning: Currently, symbolic pruning is done only for one of the rules. This would be more effective if we had run the pruning on the complete 0-form grammar.
   - Empirical Pruning: This has been done by hand, and a stronger approach requires automation. In particular, this might be achieved by using learning techniques based on supervised learning (user feedback on final results).

References

[1] N. Carriero and D. Gelernter. *How to Write Parallel Programs.* MIT Press, Cambridge, MA, 1990.

[2] N. Carriero and D. Gelernter. *How to Write Parallel Programs.* MIT Press, Cambridge, 1990.

[3] S. Collin and D. Colnet. Syntactic analysis of technical drawing dimensions. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(5):1131–1148, 1994.

[4] D. Dori. A syntactic/geometric approach to recognition of dimensions in engineering drawings. *Computer Vision, Graphics and Image Processing*, 47:271–291, 1989.

[5] D. Dori and A. Pnueli. The grammar of dimensions in machine drawings. *Computer Vision, Graphics and Image Processing*, 42:1–18, 1988.

[6] A. Habed and B. Boufama. Dimension sets in technical drawings. In *Proceedings of Vision Interface*, pages 217–223, Trios-Rivieres, CA, 1999.

[7] T. C. Henderson and A. Samal. Shape grammar compilers. *Pattern Recognition*, 19(4):279–288, 1985.

[8] T. Kanungo, R. M. Haralick, and D. Dori. Understanding engineering drawings: A survey. In *Proceedings of First IARP Workshop on Graphics Recognition*, pages 217–228, University Park, P.A, 1995.

[9] V. Subrahmanian. *Heterogeneous Agent Systems*. MIT Press, Cambridge, MA, 2000.

[10] L. Swaminathan. Agent-based engineering drawing analysis. Master's thesis, University of Utah, Salt Lake City, Utah, December 2002.

[11] K. Tombre. Analysis of engineering drawings: State of art and challenges. In *Graphics Recognition - Algorithms and Systems*, volume 1389 of *Lecture Notes in Computer Science*, pages 257–264, Springer Verlag, April 1998.

[12] K. Tombre. Graphics documents: Achievements and open problems. In *Proceedings of the 10th Portuguese Conference on Pattern Recognition*, Lisbon (Portugal), 1998.

[13] K. Tombre. Ten years of research in the analysis of graphics documents: Achievements and open problems. In *Proceedings of the 10th Portuguese Conference on Pattern Recognition*, Lisbon (Portugal), 1998.

[14] G. Weiss. *Multi-Agent Systems*. MIT Press, Cambridge, MA, 1999.

[15] P. Winston. *Artificial Intelligence*. Addison-Wesley, Reading, MA, 1984.

Acknowledgment: This work was supported in part by ARO grant number DAAD19-01-1-0013.

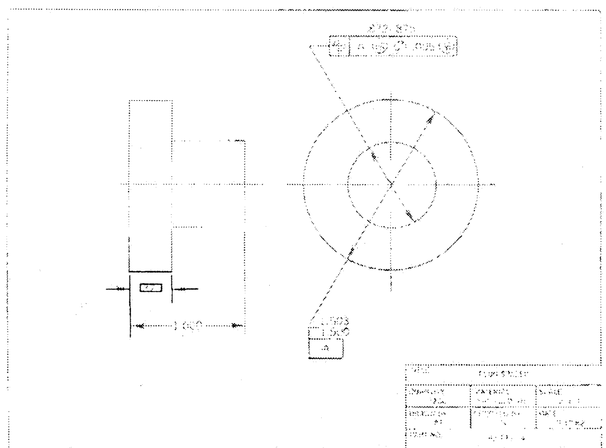

Figure 3. Dimension from Figure 1 Drawing