Semantic Interpretation of Architectural Drawings

Olubi Babalola
Georgia Institute of Technology, USA

Charles Eastman
Georgia Institute of Technology, USA

Abstract
The paper reviews the needs and issues of automatically interpreting architectural drawings into building model representations. It distinguishes between recognition and semantic interpretation and reviews the steps involved in developing such a conversion capability, referring to the relevant literature and concepts. It identifies two potentially useful components, neither of which has received attention. One is the development of a syntactically defined drafting language. The other is a strategy for interpreting the semantic content of architectural drawings, based on the analogy of natural language interpretation

Keywords
Semantic Interpretation, Drawing Understanding

Part I: An Overview and Survey

1 Introduction

Semantic interpretation of architectural drawings has the potential for simplifying the process of conversion of archived drawing information in paper or old CADD formats into the newer and intelligent model-based CADD representations which posses vastly greater capabilities. The following section addresses some of the pertinent questions, such as the difference between architectural and other recognition.

1.1 Justification for Research

Interest in extending the functionality of CAD beyond simple drafting and editing through the addition of model-based design support and analysis intelligence is almost as old as CAD research itself. Progress in these extensions hinge on the evolution of CAD systems from drawing-oriented representations to model-oriented representations, which identify each element in the design, associating its geometry and other properties and relations. A growing number of architectural CAD products of this sort exist. While each uses its own internal representation, most of these intelligent CAD systems interface with a standard building model, for exchange and interfacing with special applications. A standard building model is a public-domain open standard for representing the useful information about a building, defined as a composition of objects at different levels of aggregation.

Considerable progress has been made in the development of a standard building model. The Industry Alliance for Interoperability group (IAI) is about to release the fifth version of its Industry Foundation Classes (IFC 2.X), an industry-led effort to develop a standard building model. Similarly, The Standard for the Exchange of Product model data under the International Standards Organization (ISO-STEP) has developed exchange technologies that have been used to develop a range of standard exchange models in building, steel structures, precast concrete, and other industry domains [**CE99**]. Eventually, researchers in the building industry expect that such a standard building model will become the basis for most work in design, contracting, fabrication and building operation. Yet, given the amount of data archived in paper as well as non-intelligent CAD formats, it is clear that some means of automatically migrating drawings from these formats into intelligent model representations would save millions of man hours.

1.2 Current view of the drawing recognition problem

Semantic recognition of architectural drawings from paper-based input is a potentially vast research area. Small but relevant segments of the process have been separately tackled from the perspective of different drawing recognition domains. The most relevant effort, tackled from a directly architectural viewpoint, is the Knowledge Based Interpretation of Architectural Drawing system, KBIAD [**CJ90**] and has a focus somewhat similar to that of this paper. The KBIAD research sought to develop a generalized method for drawing interpretation, and demonstrated this using an architectural floor-plan example. A number of ongoing efforts have also been directed towards semantic reconstruction of drawing information using layer-separated CAD input.

Outside of architectural drawing recognition, there is a large body of drawing recognition research from other domains, such as office documents, schematic drawings, and mechanical drawings. Many issues are common across these domains, especially in the early stages of their respective processes, providing a useful body of relevant research sources, techniques and occasionally ready solutions to borrow. Non-architectural domains also overlap similarly, and it is conceivable that discoveries in architectural recognition may in turn influence approaches in other domains.

The objectives of drawing recognition vary widely, ranging from the conversion of raster-scanned drawings into layered CADD documents, to generating the input for a specific analysis application, to the identification of the represented objects, along with their various properties. Our interest in this paper focuses upon interpreting non-layered CAD drawings for input to building product models, which offer a common base for all other uses.

1.3 An overview of issues addressed in the paper

This paper has two parts. The first defines the various aspects of the drawing understanding problem and the multiple issues that must be addressed for product model interpretation. A rough process model is provided both as a means of demonstrating relationships and a possible sequence of the various activities, as well as providing some structure for discussion of our survey of related research. The purpose of the survey is to identify work and concepts specifically relevant to the field of architectural drawing recognition, including techniques, principles and/or algorithms.

The second part of the paper explores how rich semantics may eventually be extracted from architectural drawings, using natural language processing as an analogy. We identify the difference between drawing recognition and drawing understanding, pointing out the importance of semantics in understanding. We discuss some of the major problems involved, drawing parallels with the resolution of similar problems in natural language understanding, and in the process identifying several useful areas of research. The objective is to arrive at the beginnings of a theory of architectural drawing understanding, in the belief that such formalization will help in bridging some of the existing gaps in the process.

2 Drawing Recognition Input and Process Model Outline

2.1 Detail Description of Architectural Drawings

A variety of drawing-types are employed in architecture at different stages in the design process, ranging from unstructured sketches, to schematic designs, through to a richer and more formalized representation used in communicating construction information to a contractor [**GN97**]. The latter, called construction drawings, is the focus of this paper.

Architectural construction drawings are graphically complex, depicting an assemblage of subassemblages and parts. A typical architectural floor plan for a 3 bedroom residence may run in excess of 3500 graphical entities, while a drawing for a machine part may consist of less than 300. We define complexity here in terms of the number of geometric entities involved in depicting the ob-

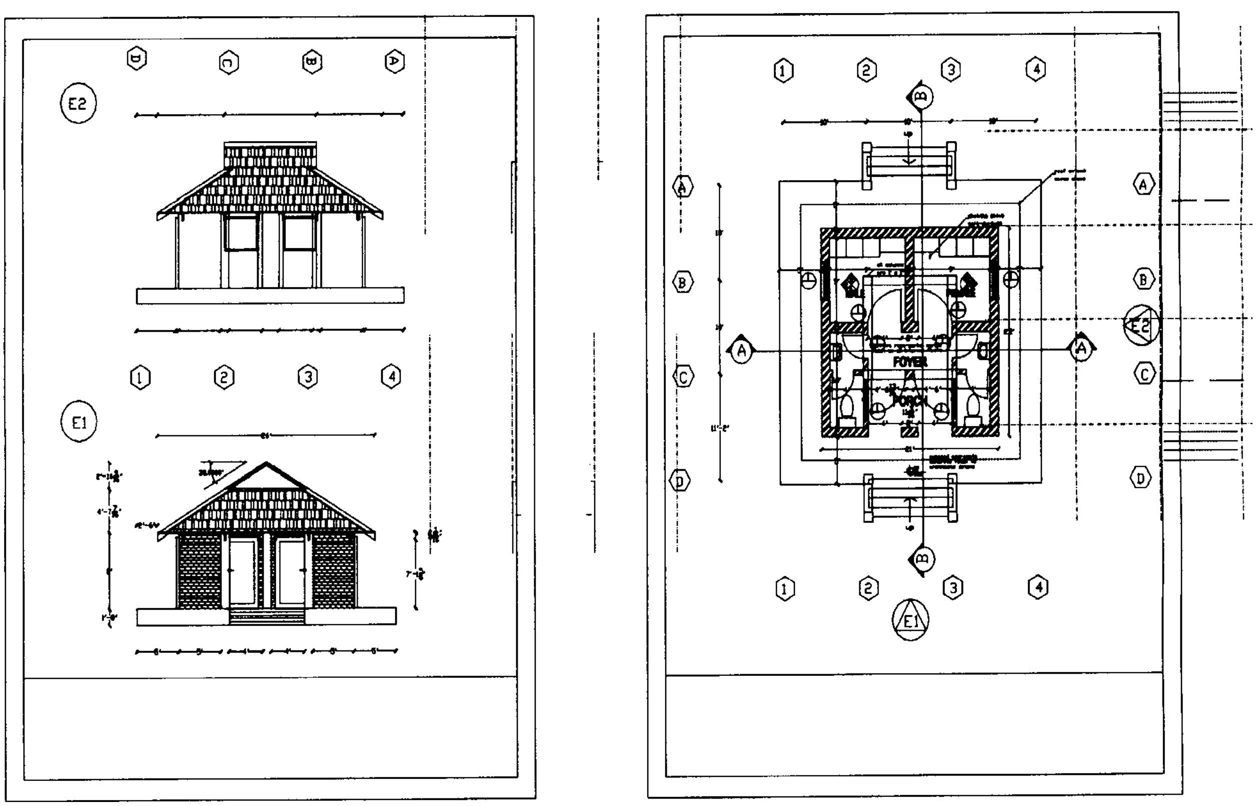

Figure 1. Architectural Drawing Samples

ject, as well as the nature of the spatial relationship between these entities, such as connections, overlap, containment and nesting. Figure 1 illustrates this complexity with sample architectural floor plans from a construction drawing.

We summarize some general properties of drawing relevant for the task of interpretation below (some of these properties are graphically presented in Figure 2.)

- Architectural drawings are typically a mixture of presentation format and semantic content, with a bordered sheet, with a panel detailing general drawing content, version and ownership, individually labeled and scaled views in each of the sheets.

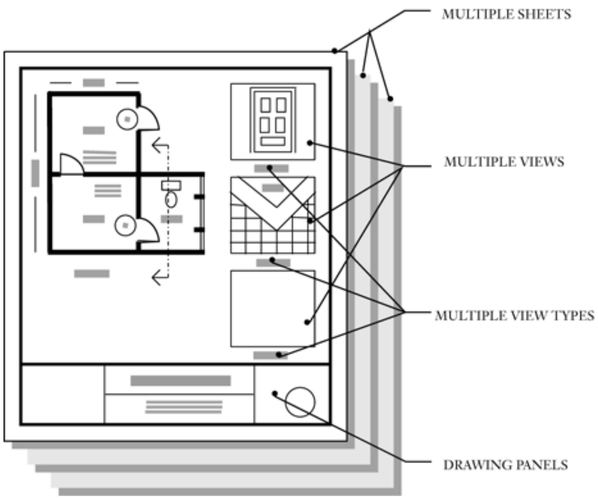

Figure 2. Component-level Schematic of Typical Architectural Drawing

- Views are orthographic 2D, 3D parallel projections (isometric or oblique), and occasionally 3D perspective views.
- Multiple views must be interpreted to derive all 3D shape and detail aspects of the design; Views required for complete object-description typically reside on multiple drawing sheets.
- Views represent building assemblages. Distinct isolated building parts are seldom represented separately.
- A view within a drawing often depicts multiple systems (architectural, electrical, structural, MEP) to indicate their placement or relationships. As a result, a view often is a complex overlay of different types of systems.
- Floor plans are the central organizing view in contract drawings, with sections and other views keyed on the plan.
- Drawings consist of a variety of symbol types, ranging from text through more restricted text notation types. Abstract symbols consisting of text, graphic symbols or a combination are used in conjunction with scaled geometry representing physical object parts.

These properties of architectural construction drawings are unique and not exactly matched by drawings in other domains.

2.2 The Drawing Recognition Process

A block diagram of the recognition process is illustrated in fig 3. It will be used as a backdrop for discussion of the architectural drawing recognition process, with raster or CADD produced drawings as input, and ending in an interpreted building model representation.

The stages in the drawing recognition process are broadly separated into low-level and high-level

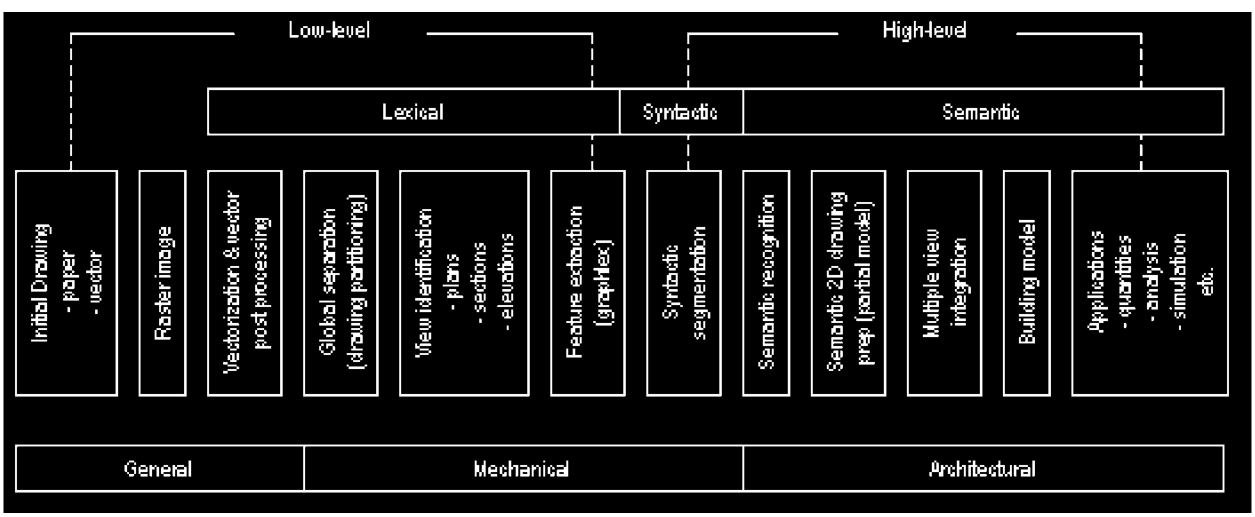

Figure 3. Drawing Recognition Process

recognition respectively. Low-level refers to any process whose input and output do not carry any understanding of what real-world object the graphic depicts [**DT95**]. An alternative taxonomy describes recognition in terms of a three level hierarchy (each level in turn consisting of multiple stages). These are *lexical*, *syntactic*, and *semantic* recognition [**KH94**, **DT95**, and **PP99**]. Lexical recognition is generally considered as the process through which the basic drawing primitives are derived. In drawing recognition, these typically take the form of lines, arcs, circles and text. Dashed lines and hatch lines are additional primitives worth identifying as a single object rather than as multiple lines, and could thus serve as an extension to a basic graphical lexicon or *graphlex*. Syntactic recognition processes the graphlex to extract complete symbols, through the application of grammar-based rules that check for localized spatial relationships between graphical entities. Dimension lines and certain symbols can be partially assembled in this manner. Semantics mainly address process control and the introduction of external contextual knowledge about the recognition process. The objective is to incorporate drawing understanding knowledge residing in the interpreter's expertise into the recognition process. The term "high-level recognition" is used in reference to approaches utilizing an object model to guide the understanding process. In fig. 3, the stages from "*initial drawing*" through to "***feature extraction (graphlex)***" will be classified as low-level processes, while all processes beyond this will be termed high-level recognition.

2.2.1 Review of the Interpretation Stages

Traditionally, architectural drawings are produced and archived in a variety of paper- based media. The production medium of drawings has a direct impact on the ability to automatically recognize them. The implication is that some production media may always prove unsuitable for computer interpretation [**KH94**].

Rasterization

Scanning is used to convert paper-based architectural representations into a digital raster format. It is usually undertaken in 8-bit grayscale, at 200 dpi or higher. The scanning process can introduce skewing, poor contrast and other problems, that must be controlled and eliminated before later stages. Raster pre-processing is applied to clean up the scanned raster image and applies *thresholding* to convert the image to a binary-valued (black-white) bitmap [**KT97**]. Other routines attempt to eliminate small isolated pixels in the drawing, smooth lines and edges, and sharpen line intersections [**TK94**]. Raster pre-processing remains an area of active research.

Vectorization & post processing

The position of the vectorization stage in fig.3 is variable, dependent upon the approach being employed. The output from this process is a subset of the lexical (graphical) entities referred to in the similarly named recognition phase. They are basic domain-independent geometric primitives such as lines, arcs, circles and text and are valid primitives across most drawing domains. The mechanical drawing recognition domain has also dealt with hatch lines and arrowheads, and we propose a few extras for the architectural domain. The task is to convert graphical entities from their raster representation into vectors. A variety of low-level algorithms have been developed in response to this task, with qualified success. In one approach, lines and arcs are detected using a 2-stage process, first involving algorithms that attempt to create a centerline through skeletonization, or by contour tracing algorithms that generate a centerline by tracing the outlines of black-pixel areas, and computing the midpoint of the outlines [**AF97**, **AB97**]. The vector representations returned vary by implementation, and range from collections of short vectors (x, y) to actual full-length vector line representations [**AF97**]. Post-processing is then applied combining segments based upon an analysis of relationships. Algorithmic design decisions arise in post-processing, such

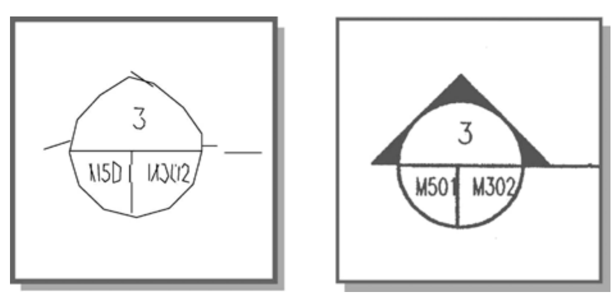

Figure 4. Post & Pre section symbol

as whether to return maximal lines or break lines at each intersection. These have potential implications in the later recognition process.

More complex basic entity shapes, including polygons, arrowheads and hatch patterns, are recognized with further post-processing. They are typically recognized using grammar-based techniques [**KV95**]. The problem with using such approaches is that it is sometimes difficult to fully predict the range of malformations that may arise from vectorization. This makes it difficult to establish a grammar-based description that is guaranteed to capture all instances of the sought pattern. A variation is a grammar-based genetic algorithm approach, which is somewhat akin to a vector-based template matching technique, with invariance to scale, rotation and translation [**EC96**].

A third alternative approach maps raster patterns to be recognized into an alternative metric space. The most common method is through the use of generalized Hough Transforms. Hough transforms are generalized template matching systems with invariance to scale, rotation and translation. Some allow the recognition of simple geometric shapes incorporating multiple graphical elements e.g. drawing symbols as primitives [**BR82**, **BT88**].

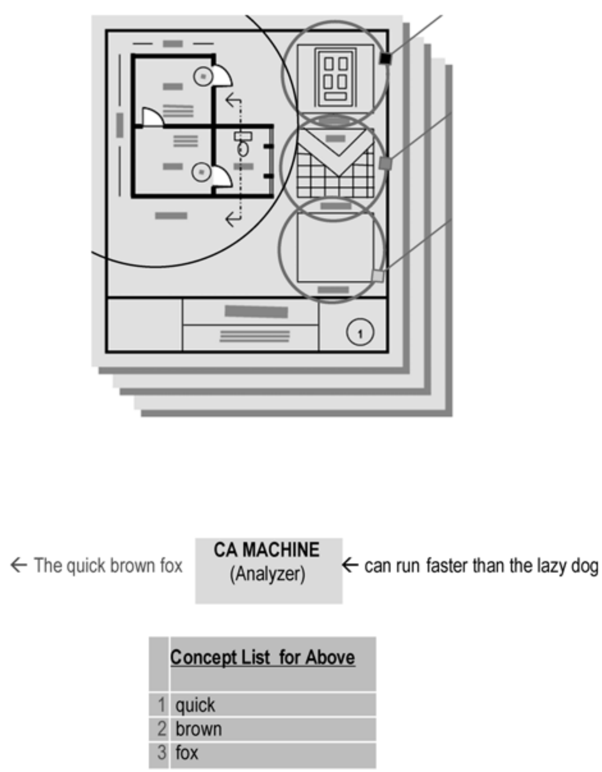

Figure 5. Segmented Views

Global separation

This process, like the next, could precede the vectorization process. Its purpose is to separate views from each other and from non-view information such as drawing panels and tables. Approaches to separation have been developed in the engineering drawing domain [**IW83**] and in office document recognition. There has been no architectural application of global separation. The challenges in separation of architectural drawings are that multiple views are sometimes so closely placed in a drawing that the process may join multiple views together as one, creating confusion in the subsequent stage.

View Type identification and relations among views

Each segmented view needs to be identified as to its projection or view type and what part of the whole project it represents. Plan, elevation or section views depict different information and use different symbols and drawing conventions. While in the mechanical drawing domain, efforts have been made to automatically associate different mechanical drawing views [**KH94**], their approach, based mainly on the comparison of dimensional relationships, will not work for architectural drawings. The problems include:

- Often, on a given page, different scales may be employed, such as for details and sections
- Architectural drawings are not arranged according to standard orthographic convention
- Architectural drawings are typically a lot more complicated than individual machine parts.

Automatic determination of correspondence between different views thus becomes a different and substantial problem in the architectural domain. There has been little or no work in view identification or segmentation for architectural or related domains.

It appears that the plan view plays a narrative role in a recognition process involving multiple views, similar to the role it plays in human drawing interpretation and it seems a logical starting point for machine recognition as well. Plans usually include symbols designating the relation of sections and some details.

Feature (graphlex) extraction:

Feature selection is a major challenge in any recognition problem, regardless of approach. The domain independent features defined in the raster preprocessing stage are combined in vector processing into higher-level domain independent entities, including hatch patterns, dotted lines, dashed-lines, polygons and arrowheads. The challenge is to identify lower-level graphical properties that characterize a domain without semantic knowledge of what the depicting geometry represents. In architectural drawing recognition, parallel lines, collinear lines, door and toilets symbols may be useful lexical entities. While the basic graphlex entities are typically derived from low-level routines, many of the more complex ones are derived from vector entities using context-free grammars or knowledge-based routines in post-processing operations [**AF97, KH94**]. An issue is the high degree of variability of many symbols, such as toilet, often requiring higher-level processes to recognize them.

Syntactic segmentation

We earlier referred to the graphical complexity that characterizes the architectural drawing recognition domain (Figure 1). In the view separation and identification stage, the focus is on limited regions of a drawing sheet. However, information relevant to the recognition of an individual view also must be partitioned, to distinguish the type of information it is. This is referred to as *segmentation*.

In an architectural floor plan, even after the more complex basic entities are recognized, a large number of simple line and arc geometry remain, such as dimension lines, grid geometry, symbols, etc. These could just as easily represent a part of wall geometry or dimension lines. The object here is to separate the possible dimension lines from the possible wall lines etc. in a segmentation process. In many ways this segmentation stage involves the same effect as the layering process used in CADD. Without segmentation, the number of lexical entities is very large and the likelihood of errors is overwhelming. We are aware of no work in this area.

Semantic recognition

Semantics in a formal linguistic sense is defined as language meaning independent of the specific wording. In architectural drawings we shall define semantics as real-object associated meaning, hence we consider a building in terms of its functional properties and structural components, and these alone or in combination will be referred to as building semantics. In architectural construction drawings, the semantics of drawings should capture associations between elements of the depicted entity and the graphical entities depicting them. For example, window-wall relationships should be captured, along with the window representing geometry in different views. Also, the relationship of these elements with other associated non-depictive symbols, such as reference symbols and annotation such as window symbols must be associated the wall geometry.

The use of context-free grammars in segmentation (typically categorized as bottom-up recognition) is useful in restricting the search space, and may actually be used to find simple symbols either partially or in their entirety (ref.). For example, a dimension line consists of numerical text coexisting in certain proximal relationships with line geometry. As far as verifying actual drawing elements in a context-sensitive grammar however, some meta-knowledge that is usually difficult to express in grammars is required. In the dimension line example stated above, drawing complexity makes it possible to associate the wrong set of text and lines using simple bottom-up or localized context-free rules. It is possible however to eliminate some of the wrong selections after subjecting the set to some higher level check.

At an implementation level, the top-down high-level knowledge controlling recognition and verification tends to be distributed. Some implementations embody this knowledge in a Knowledge Based System (KBS), others in the algorithmic flow of the program. Still other object-oriented implementations have incorporated some of the knowledge in object methods (member functions)[**CJ90**].

Our notion of semantics in architectural drawing understanding requires that we read the drawing into a representation that seeks to duplicate a human reader's understanding. As in many AI models of human understanding, this may take on the form of a semantic network [**AL95**]. The representation of drawing understanding includes not only knowledge of buildings and the component parts and their respective relationships, but also knowledge of what the different graphical entities in the drawing language represent relative to each other as well as to a real world building. These two aspects must be distinguished, even though the differences are subtle. Our understanding of buildings allows us to design our high-level checks or top-down processes according to what we know about them in the real world. For example, toilets always exist inside rooms, doors exist within walls, occupiable spaces must be accessible through openings or doors.

These examples of context sensitivity present a considerable challenge in parsing. Architectural drawings are representations of assemblages, just as sentences are collections of smaller meaningful parts such as words and clauses. It is the existence of certain context-free relationships between elements at these lower levels of meaning that provides some useful insight into tackling the interpretation problem.

Multi-view integration

The process described so far has focused largely on a single view. Integration of several views is required to define much of the represented geometry. A complete understanding of a floor plan will not result in a full understanding of the building, even for a human interpreter. While multi-view integration is an integral aspect of 3D-reconstruction in mechanical drawing understanding, it proves to be considerably more daunting for architectural drawing understanding. Independent understanding of the individual views may have to precede multi-view integration. There has been no work in architectural multi-view integration to our knowledge.

Summary

Architectural drawing understanding consist of multiple stages and activities. The preceding survey attempted to frame the architectural drawing understanding problem, collating some of the wide and disparate body of related research work in this and other related drawing understanding domains. While considerable work has been carried out in some of the stages, others remain untouched, perhaps as a consequence of the nature of the domains and focus of the related research. Amongst the drawing domains, Mechanical drawing recognition offers the greatest amount of relevant precedence, since both represent 3D objects with multiple 2D views, using somewhat similar drawing conventions.

Similarities notwithstanding, there are considerable differences between architectural drawing understanding and most of the exiting research in mechanical drawing recognition, where most efforts (save for 1 or 2) have focused on geometric descriptions of individual parts, rather than the interpretation of assemblages or semantic interpretation of the depicted objects. Given the input representation (2D drawings) we need answers to questions such as:

- How do we deal with the graphical complexity that characterizes architectural drawings?
- Is it possible to achieve (partial) semantic understanding of a view, or does the entire object need to be simultaneously understood?
- If partial semantic understanding is possible, what are the intermediate representations that will carry the semantic view information?
- Can these partial semantic views be defined such that they can be integrated in a subsequent process?

These questions reveal the fact that in spite of widespread familiarity in the use of architectural drafting conventions, there is a lack of formal understanding of the elements in terms of functional and structural relationships composing and connecting these. The semantics of the drafting language is not the same as the semantics of the represented object, and we need an understanding of the former in order to be able to answer the questions listed earlier. Simply put, in order to read drawings into a semantic model, we need some formal understanding of the architectural drawing language.

This leads us to an examination of issues relating to the definition of an architectural drafting language, and searches for analogy from the betterresearched field of natural language understanding.

Notes

1) A graphical entity in this case is a line, arc, box , text or any other basic visible constituent of the drawing
2) The stage could alternatively be located after the view identification and segmentation stage.
3) In several implementations, both the raster to vector and post processing operations appear integrated
4) A distinction must be made here between graphical primitives and components of a represented object
5) Annotation symbols, dimension objects, reference symbols and cross-reference symbols must be associated with real world-object denotations in the drawing, such as the walls to which they refer

Part II: Natural Language and Drawings

3 Theoretical background

3.1 The language/drawing analogy

Interpretation of architectural drawings implies at the very least an ability to recognize building elements, properties and relationships. Technical drawings are the primary means for communicating product information between members of the construction team. In this regard technical drawings can be considered as a language of construction communication. Nelson Goodman in 'Languages of Art' [**GN97**] attempts to identify the properties of notational and non-notational symbol systems. He lists the properties of notational symbol systems as consisting of a combination of specific syntactic and semantic properties: syntactic disjointness, syntactic infinite differentiability, semantic disjointness, semantic dissambiguity and semantic infinite differentiability. Satisfaction of these properties by architectural construction drawings supports the notion that architectural construction drawings are indeed a notational symbol system [**VG96**].

3.1.1 Architectural Drawings Understanding and Natural Language Understanding

Architectural Drawing Understanding (*ADU*) is faced with many problems similar to those in Natural Language Understanding (*NLU*). The goal in both domains is the parsing of input tokens into understanding structures, specifically words into semantic structures representing meaning and construction drawings into building models. At a more specific level, both domains share several common structural and functional properties. In both cases, understanding exists at multiple levels. In NLU, an understanding system can be queried at a somewhat low-level about actors, actor interactions, events and event chronologies. At a higher level, the same system can be queried about the moral of a story, which requires inferences at a more abstract level. Similarly with drawings, at a low-level a drawing understanding system may be queried about the spatial relationships of physical building elements, or at a higher level about spatial "parti" or "style".

Both Natural Language as well as Architectural Drafting are context-sensitive. This implies that a small subset of the information can have different possible meanings depending upon the nature and meaning of the other information that surrounds it, or in other words its context. Both systems involve a variety of kinds of knowledge for successful implementation. Parallels can be drawn between the different types of knowledge needed [**AL95**].

While the domains are by no means identical, an analysis of some approaches to text understanding, including the design of knowledge structures and process control mechanisms should help in establishing a starting point for an architectural drawing theory. The implementations of most knowledge representation structures in cognitive psychology and Artificial Intelligence take on the form of inter-linked data structures or semantic networks. At a functional level, the final representation should be capable of creating new associative links. Understanding in such systems could be described as different patterns of association links [**QM68**].

Schank and Abelson, in their work on text understanding [**SA97**], propose a hierarchy of knowledge structures that attempt to simulate a computer representation of the complexities of human understanding, including its dynamic and incremental nature. The cognitive emphasis of this work had direct implications in the knowledge structures designed, as well as process design implications.

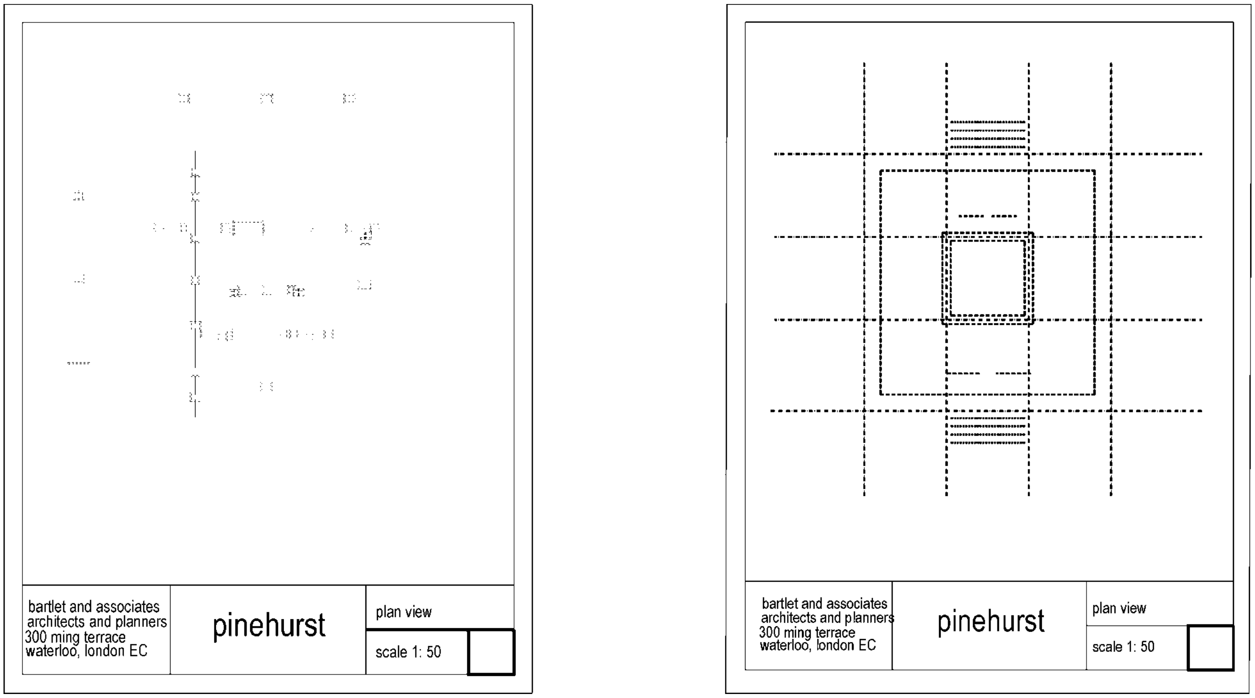

Figure 6. Dimension & Dotted Segmentation (Subsets of figure. 1)

3.1.2 Elements in Understanding Systems

The input tokens we are interested in here are at word-level. Preprocessing operations may be required in order to arrive at such input. In the case of speech, this presents a problem similar to that encountered in drawing understanding, where the raw input requires considerable preprocessing. A dictionary or *lexicon* of all words that the recognizer "understands" is an integral aspect of the recognition process. It contains instances of word types (e.g. nouns, verbs, adjectives, etc.), defined as data-structures, with fields for the instance name and interrelationships with other word types. The recognition process consists of placing input words matched in the lexicon on a *concept list*, instantiating the object methods which attempt to fill the empty fields/slots with some other word on the concept list, guided by semantic syntax rules. Since the word definitions, like the design of a data structure, largely determine what is possible in the program, particular attention and their definition is required.

The concept list for drawings will likely take on a different form in order to deal with the differences in the nature of the input. The concept list will carry the various semantic drawing elements, and sorted according to some practical spatial criteria associated with each element. This resolves the problem of trying to sort circles and squares. This also provides crucial spatial information to guide searches of the concept list, just as linear order is essential for NLU concept lists.

3.1.3 Process control mechanisms in an architectural drafting interpreter:

The semantic interrelationship of building elements suggests a possible paradox in the process of semantic interpretation. Which object forms the starting process for drawing recognition, given that its definition is almost certainly tied to that of other objects? Text interpretation involves the parsing of semantic input tokens, and the grammatical rules determine when and how these tokens should be inter-linked. The differences between single-stage semantic interpretation and hierarchical interpretation are best illustrated in Figures. 7a and 7b.

3.2 NLU and ADU: Differences and Similarities:

Drawing knowledge representation and recognition process control have not enjoyed the level of formalization and development that natural language text understanding systems have. Beyond classification of architectural drawings as a lan-

Table 1: Architectural Notational Systems

	Architectural Drawing Notational Systems	Characteristics	Sample Members
1	Textual (text and symbols have building object association. either by leader line or by containment)		
	• natural language text	Straight text in natural language. There are actually some restricted subsets of this, including room areas i.e. '2000mm x 2000mm'	• room labels • annotation
2	Symbolic Graphics		
a)	• combined text & graphics	Objects that do not exist in the physical world. Graphically composed of text and geometry in some simple relationship.	• annotation with leader lines • View reference symbols • Dimensions • grid
b)	• non object graphics	Real world building elements represented abstractly	• door symbol • toilet symbol
3	Literal graphics		
	• scaled building geometry	Directly scaled parts of objet geometry.	• walls • columns • slab edges
4	Inferred symbols		
	• spaces	Not explicitly represented in the drawing, but	• rooms • zones

Table 2: NLU / ADU Equivalent Elements

NLU / ADU Equivalent elements	
Natural language	Architectural drawings
Phonemes	Raw primitives • lines • arcs • circles • text
Morphemes	Graphlex • Polygons • Hatch lines • Isolated symbols • Collinear lines • Loops • etc.
Words	drawing semantic elements • drawing symbols • drawing building elements
Syntactic constructs (NP, VP, PP)	Segmented views • possible walls • possible dimension lines • opening object geometry
Predicates (active) ▪ atrans ▪ ptrans ▪ move ▪ ingest	Predicates (stative) • containment • connectivity • overlap • proximity
Conceptualizations/Clause ▪ implemented as data structure	Conceptualizations/Clause • implemented as data structure

guage, little effort has gone into defining the specific elements of the language and the syntactic or other rules guiding the composition of the component parts into meaningful architectural drawings. What are the structural and functional relationships between the two languages, and what are the equivalent elements in the two domains? Some initial responses to the questions are offered below.

3.2.1 Formalizing architectural drawing descriptions:

Architectural construction drawings are used to communicate descriptive information about an artifact, existing or proposed, and are composed of a variety of sub-symbol systems, both *depictive* (the "object" graphics and convention symbols) and *descriptive* (the various natural language symbols). Table 1 identifies some of the types of notational systems that comprise the architectural drawing language.

3.3 Syntactic and semantic constructs

In seeking appropriate decompositions of architectural drawing parts, we must also consider them according to the syntactic and semantic elements that compose them.

3.3.1 Syntactic constructs

At some level, there is a limit to the kind of syntactic spatial relationships that any two symbols share. There are *connectivity*, *proximity*, *containment* and possibly other predicates that can define syntactic relationships, again irrespective of domain. Different drawing domains are distinguishable by the reoccurrence of a small group of such patterns, which make it easy for a human being to trivially assign a given drawing to its appropriate domain. These constitute potential candidate features.

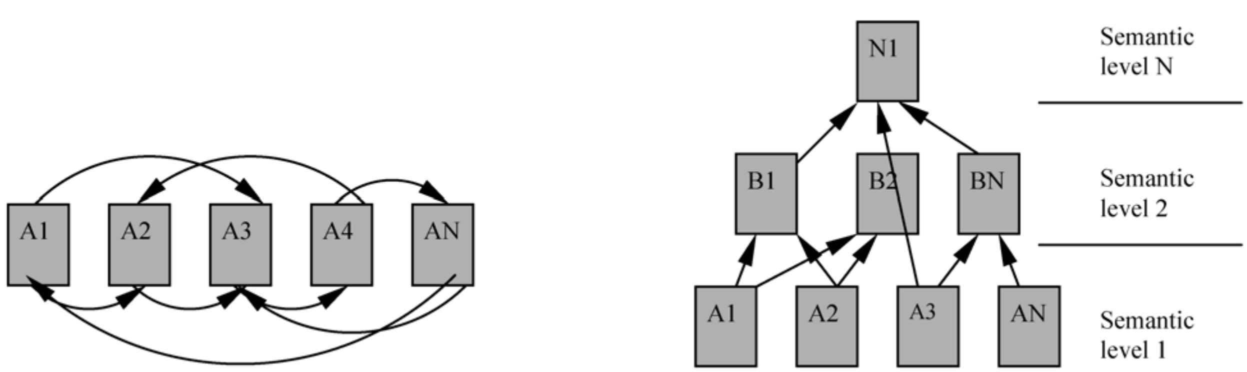

Figure 7. Single-Stage and hierarchical semantic interpretations

3.3.2 Semantic constructs:

Semantic building constructs share both a physical part-of relationship, and a hierarchical sequential relationship. Cherneff describes this as an inheritance hierarchy, and represents it in a tree-like structure [**CJ90**]. Graphical building elements in the architectural drawing language such as doors and toilets, and drawings symbols such as section and datum symbols can be considered as the equivalent of words in natural language, since they carry semantic information about the depicted/denoted object. The dependencies of the semantic parts upon each other need to be determined, and may be embedded in the building model representation. From this, appropriate grammars can be determined and a verifiable understanding of recognition dependencies can be derived.

Some structural parallels relations between NLU and ADU entities are proposed in *Table 2*. The semantic building elements, which are compared to words, are the ones defined in the drawing recognition lexicon.

3.3.3 Challenges in semantic structure design

The problem of the various levels of understanding and the possible open-ended nature of defining these levels becomes apparent as different notions of semantics come into play. This raises the question as to whether there is a generalized notion of understanding that can be defined a-priori, either in drawing or language understanding. It can be argued that for drawings, such a level exists, and is the physical-object description level of understanding. This is the level at which

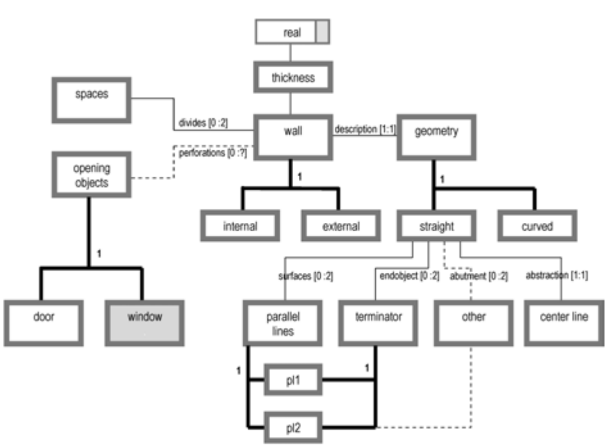

Figure 8. Partial Wall Model

the system can recognize the physical elements of the building, and their spatial relationships. Most routine queries of a drawing understanding system occur at this level of reasoning.

Table 2 proposes some of comparable structures across both domains based upon the preceding criteria.

We summarize some of the main differences in textual and drawing input as follows:

- Text is a set of 0 dimensional objects in a 1 dimensional string. There is a notion of precedence in the input sequence. Drawings on the other hand are laid out in 2 dimensions, and there is no meaningful notion of input precedence. Control structures will differ considerably as a result.
- Some drawing elements in their initial form are non-semantic (lines, circles) and others are (door and window symbols), while textual input strings are words, which carry semantic meaning.
- Textual conceptualizations generally focus on (human) actions, while drawings require stative conceptualizations, focusing on objects and descriptions of states.
- Some of the problems of ambiguity in word sense definitions (may be) less of a problem in the drawing domain.

3.4 Using the drafting representation model

Having identified possible elements of a drafting language, we need to integrate the semantics of the language with the object semantics. One way of representing this would be through the use of a drafting representation model. This is best developed as partial models with well-defined boundaries, such as a door model, a window model and a wall model. The partial models can then be integrated into a comprehensive mode, with all the related semantics. This model is useful in several ways. First, grammars for the different semantic elements or parts can be derived from these models, and this can be expressed in formal logic or other means. The modularity of the model makes it easier to focus on the subset of the description that is necessary in the formalization. Secondly, the model if properly developed provides a ready interface to the building model, which is the final representation into which we seek to read drawings. Figure 9 represents a possible module, a part representation of a wall in plan view. Finally, the model can help clarify process issues such as recognition dependencies and semantic orders of precedence.

4 Conclusion

Drawing recognition is a broad area, with a lot of cross-domain interrelationships. Architectural drawing recognition is one of the sub domains, and is perhaps most closely related to mechanical drawing recognition.

Architectural drawing understanding should be seen not only as a geometric recognition problem, which it is at the lower levels, but also as a language understanding problem, hence we wish to take a look at research in natural language text understanding. This is an area where a lot of work has been done, a similar level of which needs to be done for the drawing language.

There are obvious differences in the form of the data, hence it is necessary to establishing a greater level of formalization in our understanding of the drafting language. This may proceed along the following lines:

1. Clear functional identification of all the component symbol systems in the language, partially outlined in table 1:
2. Semantic modeling of the relationships of these to each other: determination of the grammars for the different annotation types:
3. Syntactic and semantic modeling of the relationship between views, including association between graphical representations across views.
4. Enumeration of semantic building elements such as doors and walls represented in construction drawings.
5. Partial modeling of each building element. Complete structural (syntactic) descriptions are not possible for many symbols because of the variability in their representation. Functionally (semantically), there is largely consistence. Also, in spite of variability, there are often some consistent structural characteristics. Syntactic and semantic descriptions of the parts at multiple aggregation levels are needed.
6. Integration of the partial models into a drafting language model, which itself integrates readily into the IFC or step building model.
7. Development of a practical process model for the recognition process that takes into account the graphical complexity of the domain, and simplifies this in some sequence.

A clearer understanding of the limit of the drawing language analogy is needed, as a guide to both process control and knowledge representation design.

Finally, it is important to establish a strong framework for drawing interpretation research, so that problems are understood and can be addressed with a clear focus as to where a particular problem falls within the larger recognition framework. Some of the important challenges of this have been discussed in the preceding sections, though further study is necessary before a suitably robust framework is arrived at.

Bibliography

[AF97] Ablameyko, S. Bereishik V. Frantskevitch O. (1997*) Algorithms For Recognition of The Main Engineering Drawing Entities* IEEE PP 776-779??

[AB97] Ablameyko, S. Bereishik V. (1997*) Recognizing Engineering Drawing Entities: Technology And Results*. IEEE Conference Publication No. 443 PP 15-17

[AL95] Allen J. (1995) *Natural Language Understanding.* Benjamin/Cummins Book Publishers Inc. Redwood California.

[BB87] Barton, G. E. Berrywick, R.C. Ristad, E.S. (1987*) Computational Complexity and Natural Language*. MIT Press, Cambridge MA

[BR82] Ballard .D and Brown. C (1982) *Computer Vision*, Prentice-Hall,

[BG98] Bechtel, W. and **Graham, G**. (1998) *A Companion to Cognitive Science*

[BS74] Bijl A. Shawcross G (1974) *Housing Site Layout System :* Report on System Developed For Scottish Special Housing Association & Department of the Environment

[BT88] R. Boyle and R. Thomas (1988) *Computer Vision: A First Course*, Blackwell Scientific Publications,

[EC99] Eastman, C (1999) *Building Product Models (Computer Environments Supporting Design)* CRC Press

[CJ90] Cherneff, J. (1990) *Knowledge Based Interpretation of Architectural Drawings :* Ph.D. Thesis, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, MA

[CL75] Collins, A.M. and **Loftus, E.F**. (1975*). A spreading-activation theory of semantic processing*. Psychological Review, 82, 407-429.

[GK86] Gessima E. Kamal L. (1986) *Computer Aided Analysis of Schematic Drawings* Pattern Recognition In Practice II Elsevier Science Publications.

[VG96] Goel V. (1996) *Sketches of Thought*, MIT Press, Cambridge, MA.

[GR78] Gonzalez R. Thomason M. (1978*) Syntactic Pattern Recognition : An Introduction*. Addison Wesley Publishing

[GN97] Goodman Nelson (1997*) Languages of Art: An Approach to a Theory of Symbols* Hackett Publishing Co, Inc

[GV86] Groen, F.C.A.., Van Munster, R.J (1986.) *Computer Aided Analysis of Schematic Diagrams* : in Pattern Recognition in Practice ll, E.S Gelseman

and L.N. Kanals eds. Elsevier Science Publishers B.V. , North Holland,

[HQ82] Haralick, R.M. Queeny, **D.** (1982) *Understanding Engineering Drawings* Computer graphics and image Processing Vol 20, pp. 244-258

[HG86] Hirschtick, J. Gossard, D. (July 1986*) Geometric Reasoning for Design Advisory Systems*: Proceedings of the1996 American Society of Mechanical Engineers Computers in Engineering Conference and Exhibition, Chicago, IL, pp. 263-270

[HO87] Hofer-Alfeis, J., (1987) *Automated Conversion of Existing Mechanical Drawings to CAD Data Structures : State of the Art*, in K. BØ, L. Estensen, P.Falster, and E.A. Warman (eds) Computer Applications in Production and Engineering, CAPE'86, North-Holland Amsterdam, pp. 259-267

[HU79] Hopcroft, J.E. ,Ullman, J.D. , (1979) *Introduction to Automata Theory*, *Languages and Computation*, Addisson Wessly

[IW86] Iwaki O (1986) *Document Recognition System for Office Automation* Proceedings 8th ICP

[JO89] Joseph, S.H., (1989*) Processing of Engineering Drawings for Automatic input to CAD*. Pattern Recognition, Vol. 22 pp. 1-11

[KH94] Kanugo T. Haralick R. Dori D. (1994) *Understanding Engineering Drawings: A Survey*. Intelligent Systems Laboratory Department of Electrical Engineering University of Washington.

[KV95] Kiyko V. (1995) *Recognition of Objects In Images of Paper Based Line Drawings*. IEEE xxxx PP 970-9736

[LR82] Lawson, B.R., Riley J.P.,(1982) *A Technique for the Automatic Interpretation of Spaces from Draw Building Floor Plans*. In CAD 82: 5th International Conference and Exhibition on Computers in Design Engineering, Butterworths, Guildford pp. 663-666

[ML86] Miclet, L., (1986*) Structural Methods in Pattern Recognition* North Oxford Academic, London

[MW87] Mitchell, W., (1987) *The Logic of Architecture* MIT Press, Cambridge, MA

[MS76] Mitchell W. Steadman J. Ligett R. (1976*) Synthesis And Optimization of Small Rectangular Floor Plans*: Environment And Planning B, Vol. 3 PP 37-70

[PP99] Prabhu B. Pande S. (1999) *Intelligent Interpretation of CADD Drawings* Computer and Graphics Vol 23 PP 25-44

[QM68] Quillian, M. R. (1968). *Semantic Memory*. In Minsky, M. (ed.). Semantic Information Processing. Cambridge, Mass.: MIT Press.

[SG83] Sakurai, H., Gossard, D. (1983) *Solid Model Input Through Orthographic Views*. Computer Graphics, Vol 17, No3, July

[SA77]Schank, R. Abelson, R. (1997) *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures* Lawrence Erlbaum Associates

[**TK94**] **Terzides Kostadinos** (1994*) Computer Aided Extraction of Morphological Information from Architectural Drawings* Ph.D. Dissertation, Michigan

[TS93] Tsai, D.H. Shaw, D. (1993*) Syntactic Pattern Recognition and Aided System for Mechanical Drawing Diagnosis*. American Society of Mechanical Engineers Winter Annual Meeting, New Orleans, LA Pub. ASME, New York, NY, Vol. 64, pp. 273-280

[WD75] Waltz, D. (1975*) Understanding Line drawings of Scenes with Shadows*. In The Psychology of Computer Vision, P. Winston, ed., McGraw-Hill Book Company, New York.