Intelligent Critic System for Architectural Design

Hon Wai Chun, *Member, IEEE*, and Edmund Ming-Kit Lai, *Member, IEEE*

Abstract—This paper describes an intelligent computer-aided architectural design system (ICAAD) called ICADS. ICADS encapsulates different types of design knowledge into independent “critic” modules. Each “critic” module possesses expertise in evaluating an architect’s work in different areas of architectural design and can offer expert advice when needed. This research focuses on the representation of spatial information encoded in architectural floor plans and the representation of expert design knowledge. Described in this paper is our research in designing and developing two particular “critic” modules. The first module, FPDX, checks a residential apartment floor plan, verifies that the plan meets a set of government regulations, and offers suggestions for floor plan changes if regulations are not met. The second module, IDX, analyzes room and furniture layout according to a set of interior design guidelines and offers ideas on how furniture should be moved if the placement does not follow good design principles.

Index Terms—Computer-aided architectural design, design knowledge representation, spatial knowledge representation, hybrid knowledge representation, diagrammatic reasoning, representation of government regulations, interior design principles, expert systems, decision support systems.

◆

1 INTRODUCTION

COMPUTER-aided architectural design systems (CAAD) have the potential of being a much more intelligent assistant to the user. Besides providing the basic tools to draw floor plans and create 3D models, CAAD systems can be extended using artificial intelligence (AI) techniques to offer advice and suggestions to the user as he performs his work [25], [28], [31], [35]. This next step in the evolution of CAAD systems is in many ways analogous to the evolution in modern word-processing software. Before, word processors were simple replacements for the typewriter, albeit with many enhanced features. As these software packages improved, they began to provide more intelligent tools such as spell and grammar checkers. Some word processors can even correct mistakes as you type. The current level of intelligence has improved to the point that some packages even provide a set of “experts” or “wizards” that stand by and offer expert assistance when requested. These wizards offer suggestions and guide the user step-by-step in performing complicated word-processing work.

Our research is focused on designing similar “wizards” for architectural CAD systems. We call the software wizards in our research “critics” since the modules we have developed acts like human critics that review architectural drawings and offer criticism and advice. This paper describes the overall system architecture of our Intelligent CAD System (ICADS) which include two main components—a component that analyzes a drawing to produce an internal knowledge representation and another component, consisting of the critic modules, that reasons with this internal representation. The first critic module is called “Floor Plan Design Expert” (FPDX) which assists architects in evaluating whether an apartment building floor plan adheres to certain government requirements and standards. The second critic module is called “Interior Design Expert” (IDX) which analyzes an interior design based upon a set of design principles and offers suggestions for improvements. Our research approach is to gradually improve the intelligence of ICADS by incrementally adding additional critic modules that offer expertise in different areas of architectural work [41]. The critic modules are very similar in concept to the knowledge worlds implemented in EKSPRO [33].

Similar to EKSPRO, the critic modules described in this paper are rule-based in implementation using Prolog. Although much of the architectural design work is an art that varies with individual style and taste, there are still some basic guidelines or rules to follow [4], [23], [43], [47]. For example, a floor plan for an apartment must follow government regulations that ensure the building design is safe and provides a healthy and comfortable environment for the residents. For interior design, there is also a set of established design principles or rules-of-thumb, which should be followed to provide a pleasant environment. The designers’ creativity must work within these constraints. One of the main objectives of ICADS is to provide an environment where the architect or designer can focus his attention on creative work leaving the regulations and rules to ICADS.

2 RESEARCH BACKGROUND

The main research focus is to develop a spatial representation that is rich enough to capture qualities of spatial relationships

---

- *H.W. Chun is with the Department of Electronic Engineering, City University of Hong Kong, Tat Chee Ave., Kowloon, Hong Kong. E-mail: eehwchun@cityu.edu.hk.*
- *E.M.-K. Lai is with the Department of Electronics and Communication Engineering, Edith-Cowan University, Perth, Western Australia.*

Manuscript received 14 Mar. 1995; revised 18 Jan. 1996.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 104401.

that are important in reasoning with government regulations and design principles. In the past, spatial reasoning has been concerned mainly with image processing, robot manipulation, natural language understanding, route finding and exploration, imagery, or qualitative reasoning. Few research projects have dealt with reasoning with objects in the architectural domain. The main goal of our research is to investigate the limits of and to propose extensions to current spatial representation for architectural design work.

ICADS is closely related to research performed by Pau and Nielsen [33] in a system called EKSPRO, a knowledge-based system for architectural design considering energy saving and thermal comfort. This system selects layouts, materials and equipment to improve heating and lighting while minimizing total energy costs. Similarly, EKSPRO also has several knowledge bases; KB-1 to store knowledge on occupational safety regulations and KB-2 to store knowledge on architectural design and engineering knowledge. The FPDX critic module of ICADS is similar to KB-1 except it contains regulations on fire safety and natural lighting. ICADS's IDX critic module is similar to KB-2 except the architectural design knowledge is mainly focused on interior design. In addition to KB-1 and KB-2, EKSPRO also has a set of calculation modules—CAD-1 to CAD-17, which performs computation on a CAD database. Although not as extensive as EKSPRO, ICADS also has calculation capabilities to perform basic geometric computation ranging from length, areas, and diameters to directions and angles. Both systems are implemented using Prolog logic programming.

The motivation and concept behind ICADS are also related to the Fischer's work in critic system for kitchen design called Janus [16]. Janus's knowledge-based critics "watch over the shoulders" of designers and display their critique when design principles are violated. Janus is linked with a hypertext system called CatalogExplorer [15] to retrieve and display design examples to assist the designer. CatalogExplorer is a system that assists architectural design of kitchen floor plans through an innovative combination of case-based reasoning [44] and hypermedia. The task is to retrieve floor plan designs that match given design requirements to improve human designers' productivity. ICADS has expanded upon the idea of Janus to provide a general representational framework with which different types of critics can be built.

ICADS is also related to Waltz and Boggess' work that is documented in [48]. In that system, natural language input sentences were parsed and processed to produce an internal visual analog representation of the scene described by the input. The program can then answer questions about object relationships that may or may not have been explicit in the original scene description. It uses a knowledge base of default object size and location to fill-in missing information. It was found that direct testing of visual analog representations can bypass long chains of reasoning and avoid combinatorial problems. ICADS also uses an internal analog model for its reasoning, but adopts concepts of reference intervals [2] from temporal reasoning to control combinatorics.

ICADS is also related to the NALIG system described in [18]. In NALIG, natural language scene descriptions are parsed into conceptualizations. Missing information is filled-in using knowledge of typical situations. A positioning module locates possible positions for each object, while a graphics module displays possible interpretations of the natural language input. In ICADS, the focus is not in providing an educated guess for the scene description, but more on encoding higher level relationships among objects for design reasoning. In particular, the interface to ICADS is just reversed; the user provides the graphic interpretation to ICADS. Another related research is Garijo and Garrido's work [17] which is a knowledge-based system that designs the orientation and layout of a house based upon user requirements, such as cost and family size. Good design principles for room layouts were also incorporated. This system creates a design from given requirements, while FPDX and IDX critic modules analyze a given design.

3 ICADS SYSTEM DESIGN CRITERIA

In designing the ICADS system architecture, several important assumptions and criteria were considered. These factors determined the scope of ICADS knowledge representation. In spatial reasoning systems that take natural language as inputs [18], [48], one of the main problems is that the input to these systems only provides partial and incomplete information. Default knowledge of each object must be available, such as the typical object location and size. From the user input and default reasoning, an internal representation is then created. In ICADS, it is assumed the architectural or interior design drawing given to the system is quite complete and precise. From the drawing, ICADS extracts basic geometric data. These spatial concepts are then used to generate higher level relational primitives used by the rules stored in the critic modules. However, default knowledge such as typical dimensions and orientation of objects [13], [48], [18] are still maintained by ICADS. In addition, the scene analyzed by ICADS is assumed to be static. On the surface, this domain may seem to be easier than those tackled by other AI systems. However, this is not the case.

Although the preciseness of the domain gives ICADS a computational advantage, the derivation and assimilation of spatial relationships are still difficult tasks especially when the number of objects involved is large. By isomorphism between time and space [30], the concept of reference intervals [2] is adopted to solve the combinatorics by creating a hierarchy of partitions to group the objects [10], [11]. This concept will be further discussed in following sections. In addition, design rules are usually imprecise and fuzzy. Therefore, the choice of knowledge representation used within ICADS must handle this type of fuzziness.

Another criterion is that a uniform reference frame is not sufficient for spatial reasoning. Instead, different reference frames or coordinate systems have to be maintained according to the context and objects involved [1], [13], [48]. Transformation is required to switch between different reference frames.

4 ICADS SYSTEM ARCHITECTURE

The ICADS architecture (see Fig. 1) consisting of two major parts—the Spatial Analysis Component and Critic

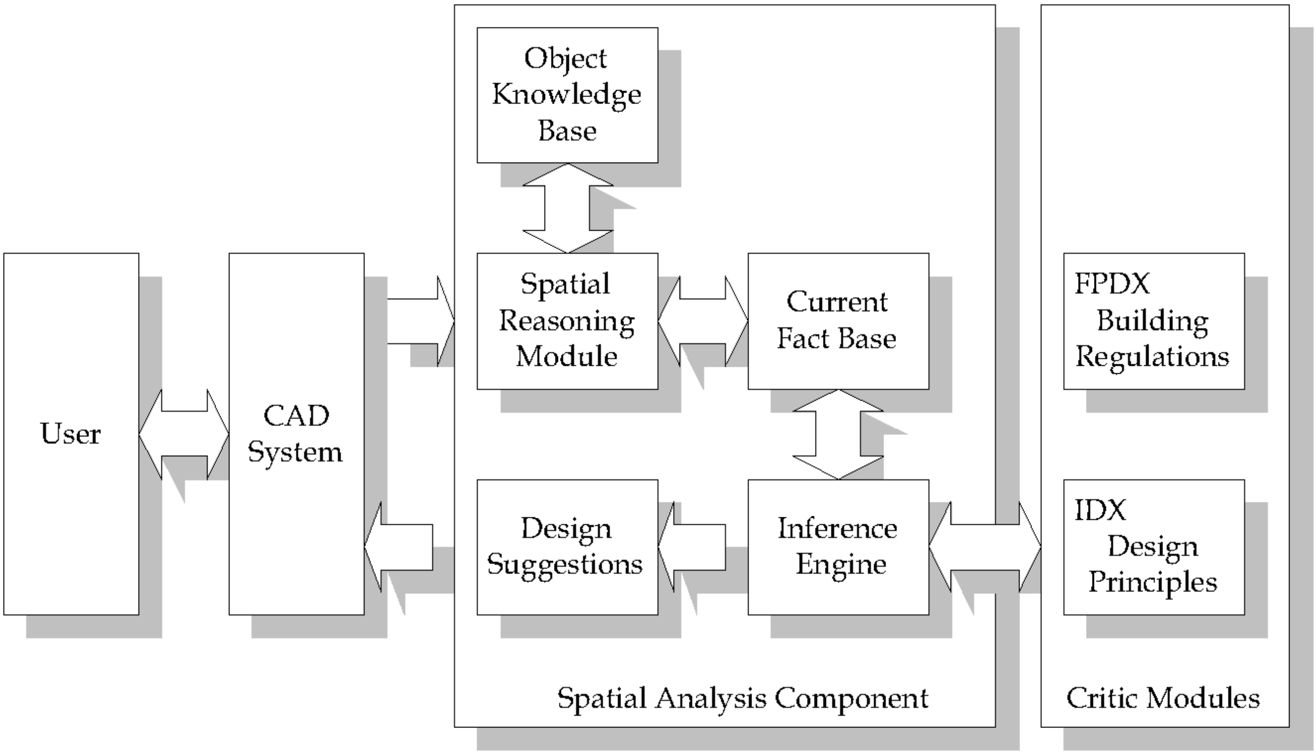


Fig. 1. Overall system architecture for ICADS.

Modules. The Spatial Analysis Component forms the foundation and backbone for the Critic Modules. This component is responsible for taking a graphic drawing and converting it into an internal knowledge representation that the critic modules can understand. This section provides a description of the modules in the Spatial Analysis Component.

ICADS is designed to be an embedded system in a conventional CAD package. Additional object libraries and commands are encoded into the CAD system itself. For our research, we have been using AutoCAD[1] as the front-end CAD system. However, the research developed is independent from the front-end CAD system and can potentially be applied to other systems as well. The following describes the main modules of the Spatial Analysis Component.

4.1 Spatial Reasoning Module

The "Spatial Reasoning Module" is implemented using the front-end CAD system to analyze an architect drawing that was drawn using the ICADS object library. This module identifies all relevant objects in the drawing and computes geometric information about these objects [26], [45], [46]. Based on the low-level geometric information, higher level spatial relationships among objects are extracted and stored using a set of ICADS spatial primitives. Domain knowledge about architectural objects, used to analyze a drawing, is taken from the "Object Knowledge Base." The result produced by using this module is then stored as the "Current Fact Base."

4.2 Object Knowledge Base

This module is a repertoire of information on all the objects in the architecture domain that ICADS understands. This includes knowledge of elevators, stairs, apartments, rooms, walls, windows, doors, fireplaces, furniture, fixtures, and appliances. These objects are defined as a hierarchy of classes [5], [51]. Fig. 2 shows a high-level class diagram of the ICADS classes using the Booch Notation [7], [50].

The "Spatial Analysis Component" represents each object in a floor plan as a subclass of "`ICADS Object`." Attribute values of each instance are either obtained from the drawing

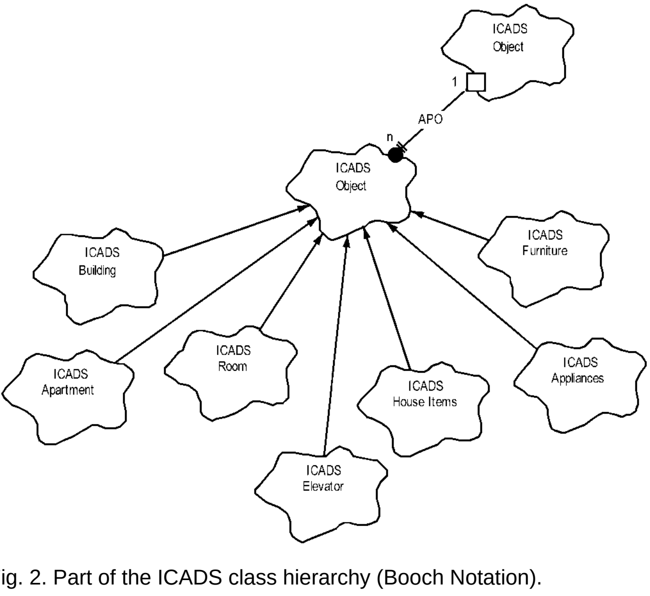


Fig. 2. Part of the ICADS class hierarchy (Booch Notation).

---

[1] AutoCAD is a registered trademark of AutoDesk Inc.

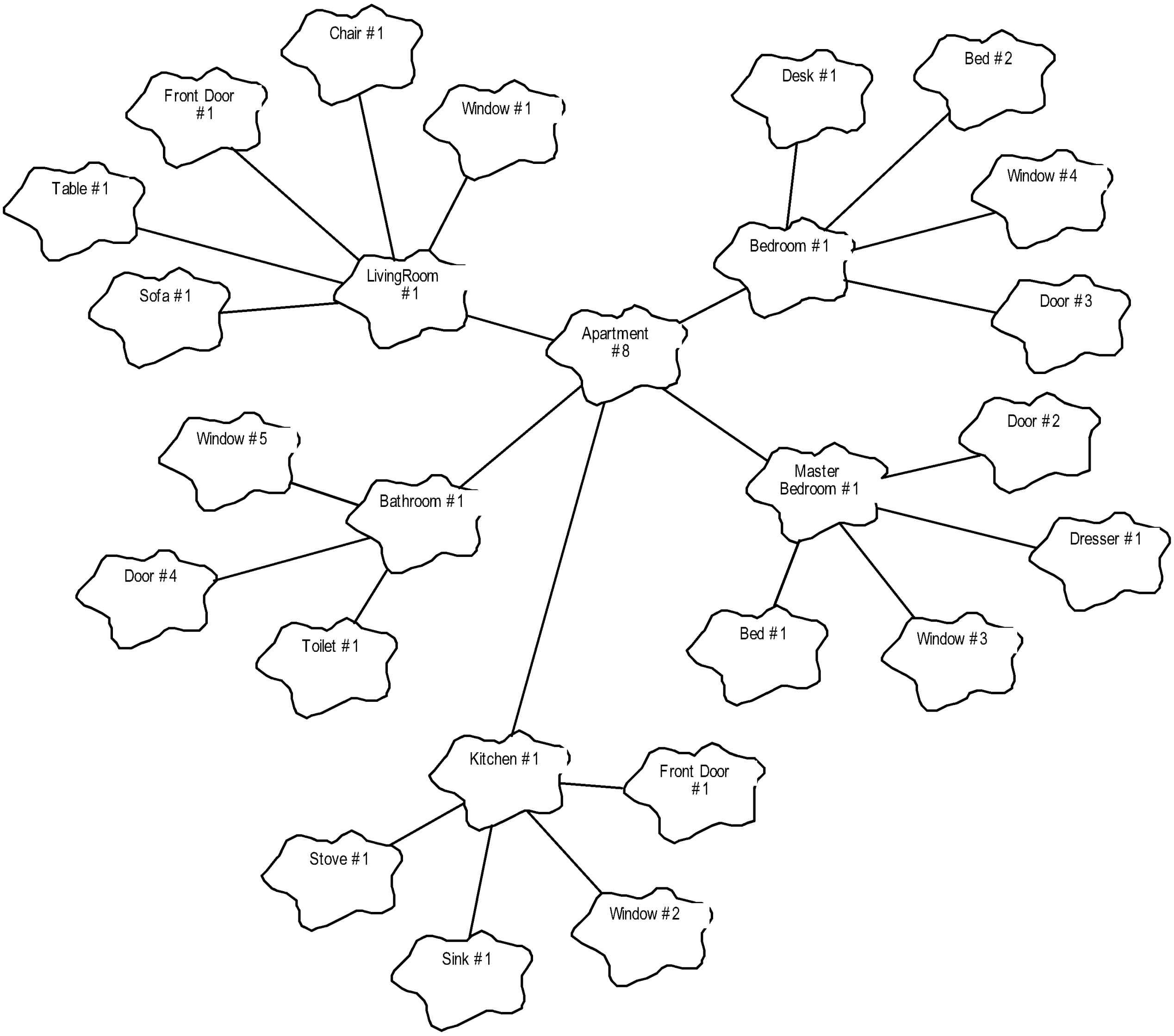


Fig. 3. An example of an APO hierarchy for a particular apartment.

directly, inherited from the parent (e.g., the typical depth of a room or the area of a window), or computed using daemons (e.g., the actual site coverage is computed from the site-coverage daemon of the building frame). Default class information is retrieved from the "Object Knowledge Base."

Since the "`ICADS Object`" has an a-part-of (APO) relationship (see Fig. 2), all its subclass will also inherit this relationship. In other words, there are two main hierarchies—the class inheritance hierarchy and the APO hierarchy. After object instantiation, the "Spatial Analysis Component" generates the APO hierarchy and computes the *spatial relationship graph*. The class inheritance hierarchy is defined in the "Object Knowledge Base" while the APO hierarchy is instantiated and stored in the "Current Fact Base" with the *spatial relationship graph*. Fig. 3 shows the partial APO hierarchy created from an example floor plan. Fig. 4 is a portion of the class inheritance hierarchy for the "ICADS Room" class shown previously in Fig. 2.

Fig. 5 is a portion of the class hierarchy for the "`ICADS Furniture`" class. In ICADS, objects like sofas and chairs, are modeled as two-dimensional shapes such as circles or rectangles of different orientations and sizes [8], [10], [11], [12]. Although objects are approximated by simple shapes in computation, they are displayed more realistically in the CAD front-end.

The geometric data structure used by ICADS is stored as part of the object instance. The type of information stored in the data structure varies for different types of objects, but it is typically contains geometric information such as the center of the object, its length and width, its diameter, its shape, and the direction the object is facing. These geometric attributes are easily extracted from low-level geometric data provided by the front-end CAD. Fig. 6 shows the detailed structure of the geometrical data used by ICADS using the Booch Notation [7]. The geometric data includes attributes such as "`direction`" (represented as a vector), "`region`" (represented as a set of points), "`shape`," "`coordinate system`," "`area`," "`center`," "`diameter`," "`length`," and "`count`." (a class attribute). From this set of low-level geometrical data, higher level relationships are extracted, such as intersections of regions, visibility, proximity, relative distances, relative size, or relative location.

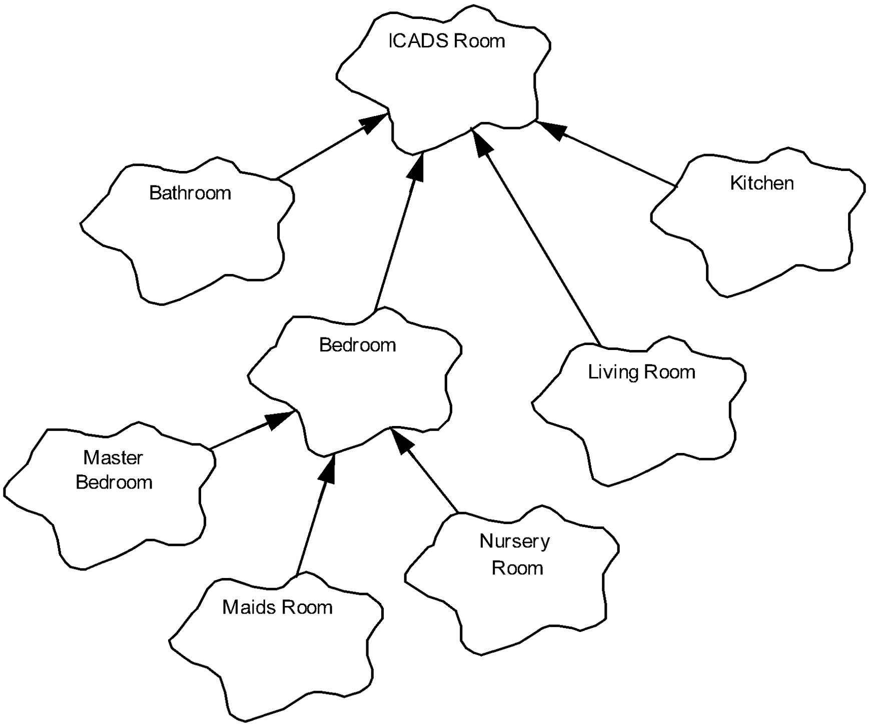


Fig. 4. Part of the class hierarchy for the ICADS Room class.

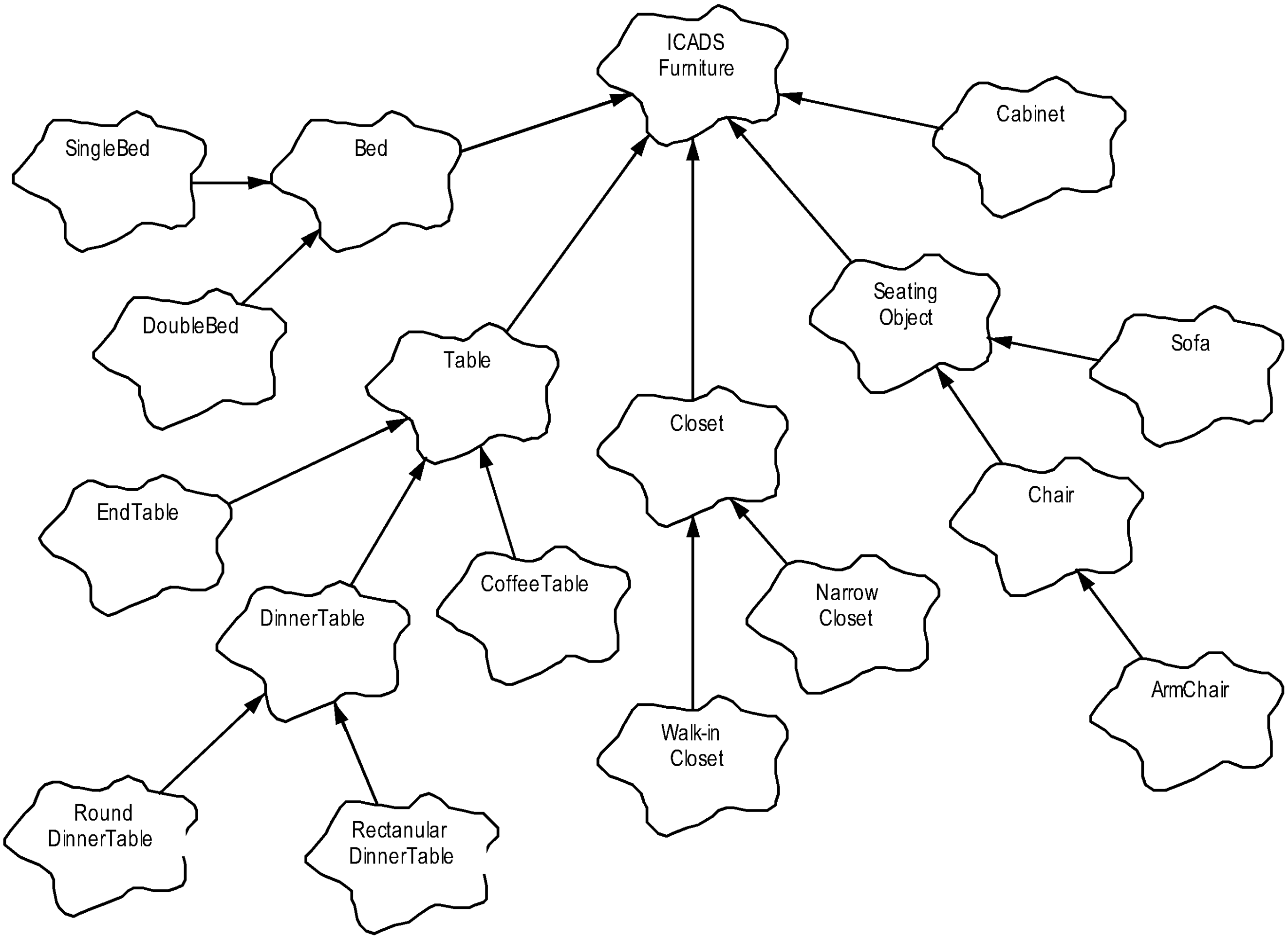


Fig. 5. Part of the ICADS `Furniture class` hierarchy.

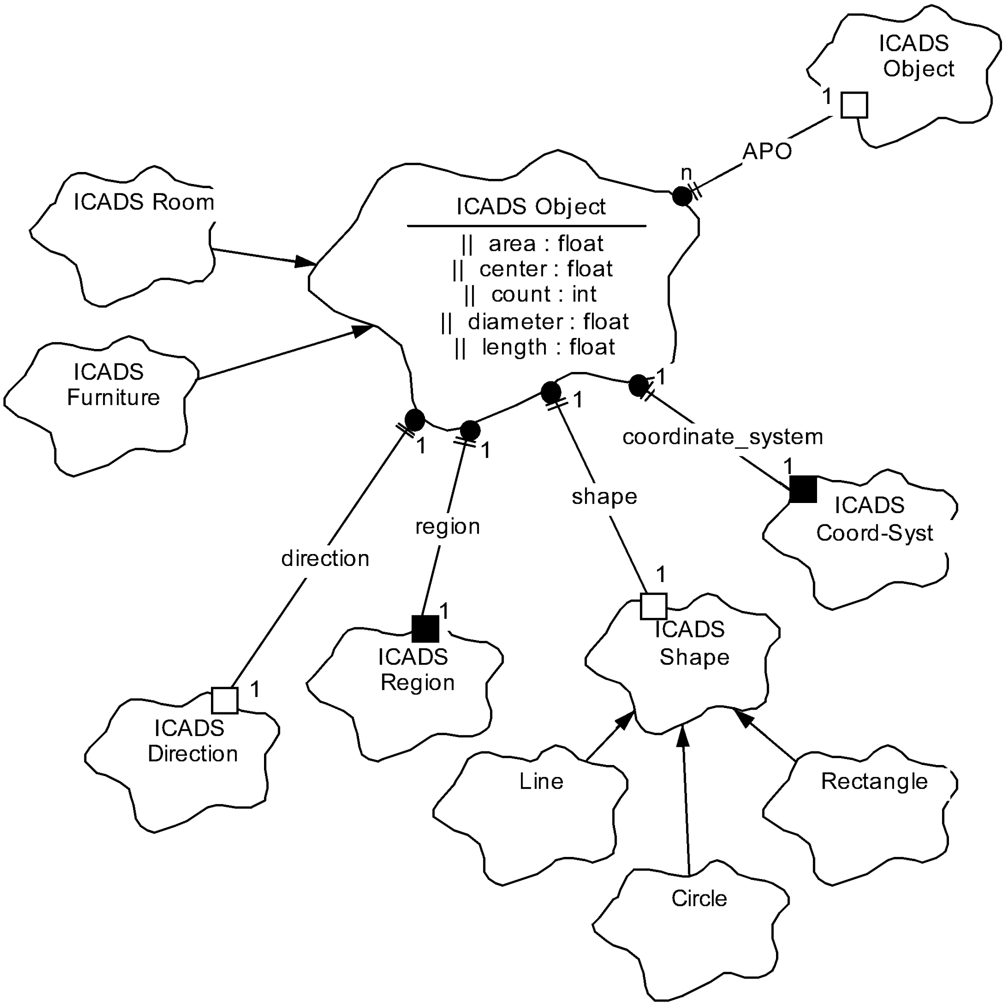


Fig. 6. The structure of the ICADS object (Booch Notation).

4.3 Current Fact-Base

The "Current Fact-Base" stores the results generated by the "Spatial Reasoning Module" in analyzing a drawing file. The fact-base contains object instances of all the objects in the floor plan with their attributes and values, the partitions of the objects, and the spatial relationships among the objects. Currently, the fact-base is represented as a set of Prolog predicates [6], [24] with object-oriented extensions. The APO hierarchy (see Fig. 3), is part of the "Current Fact-Base."

4.4 Inference Engine

The inference engine matches rules stored in the critic modules to the spatial representation stored in the "Current Fact-Base." FPDX rules will fire if building regulations are violated while IDX rules will fire if design principles are not met. The ICADS inference engine is built using the inference capabilities of Prolog [24]. Each fired rule generates a proposal to the "Design Suggestions" module. Currently, suggestions are displayed as textual messages. We are investigating the potential of having ICADS modify the drawing automatically to satisfy the critic rules. Fig. 7 is a detailed flow-chart of the ICADS inference engine. The key steps are:

- **Step 1**. The user creates or modifies the floor plan or interior design using the CAD front-end and ICADS graphic objects.
- **Step 2**. After the user is done, the floor plan design is saved into an external file. This file contains the primitive geometric data of each object in the floor plan.
- **Step 3**. The "Spatial Reasoning Module" analyzes this external file together with knowledge of the domain stored in the "ICADS Object Knowledge Base." The "ICADS Object Knowledge Base" provides the class hierarchy used for default knowledge. The geometric data obtained from the CAD system together with default knowledge is used to generate the APO hierarchy and the *spatial relationship graph*. These two components are stored into the "ICADS Current Fact-Base."
- **Step 4**. The "ICADS Inference Engine" built using Prolog pattern matching, matches the spatial knowledge encoded in the "ICADS Current Fact-Base" with the rules from the FPDX or IDX critic.
- **Step 5**. If the spatial relationship between the objects of the floor plan violates the critic rules, then design suggestions will be generated and stored away in a file. The ICADS design suggestions are currently flat structured. We are extending the system to add a hierarchy

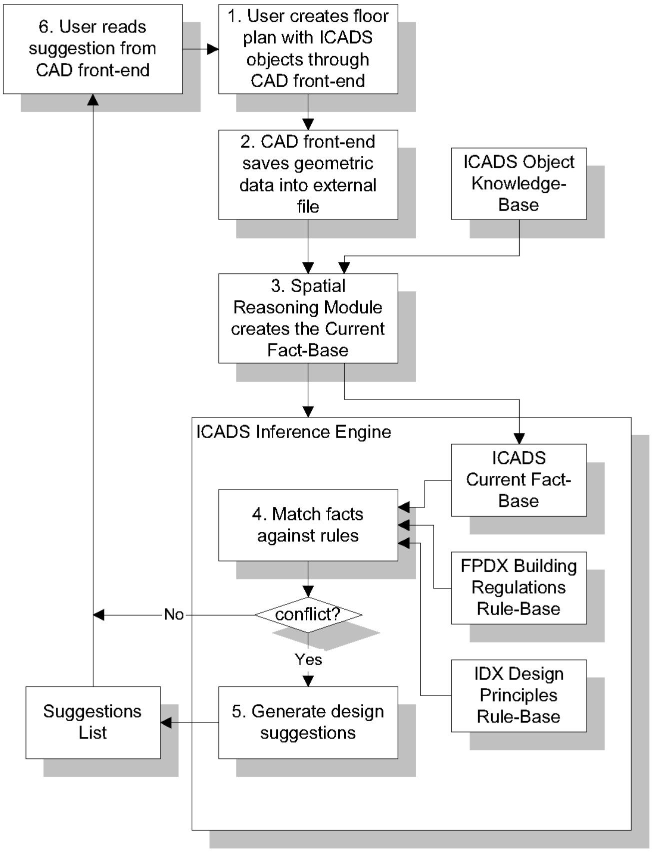


Fig. 7. Flow-chart of ICADS processing.

to the suggestions. The hierarchy will be constructed based on the severeness of the rule violated. Suggestions that can solve violations of higher priority rules will be proposed first.

- **Step 6**. The user retrieves the suggestions generated by ICADS through the CAD front-end. The user can immediately make modifications to his floor plan design and call ICADS again to critique his work.

5 ICADS SPATIAL REPRESENTATION

Current spatial representations are mainly derived from the field of natural language understanding with emphasis on the common prepositions used in daily dialogs, such as ON, IN, IN-FRONT, BEHIND, OVER, UNDER, and BETWEEN [1], [13], [18], [30], [48]. Formalization is difficult since prepositions are ambiguous, and contextual factors (e.g., salience, relevance) are usually present [13]. However in architectural work, precise analog model of the scene is readily available and the representational focus is in encoding higher level spatial relationship among objects from low-level geometric data. For instance, one basic design rule stored in the IDX critic module says:

IDX Rule D4

Condition: A door should not directly face a window.

Reason: People do not usually like rooms with a draft.

Suggestion: Place object between door and window OR move door or window.

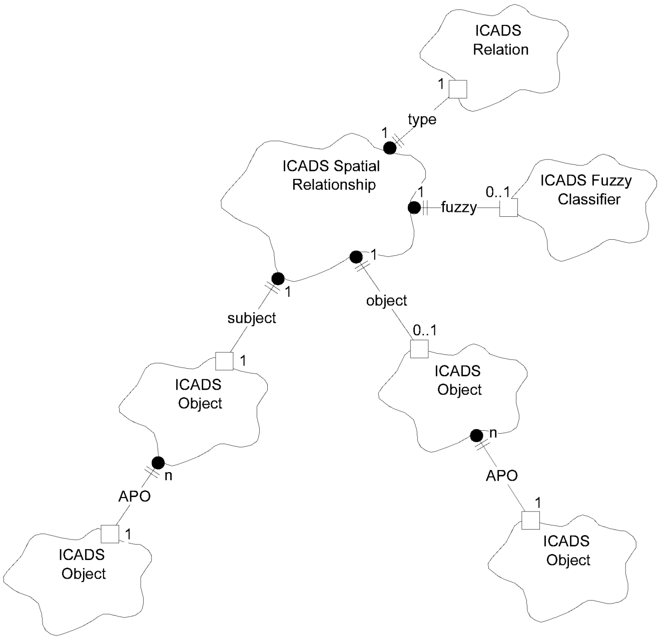


Fig. 8. The ICADS spatial relationship (Booch Notation).

We can see that the concept of "directly facing" cannot be captured by the developed preposition formalization because of its abstractness. Many concepts needed in interior design are quite high-level, such as the visibility of an object from others, the relative position of an object with respect to its locating environment, a room, for example. All these suggest that an extension to the current spatial representation is needed in order to perform design reasoning. For our previous example of "directly facing," instead of representing this concept with a complicated set of low-level primitives, we introduced a group of fuzzy "facing" concepts: **directly facing**(A, B), **slightly facing**(A, B), and **a bit facing**(A, B).

In ICADS, all spatial relationships are represented as an in-stance of the "ICADS Spatial Relationship" class, see Fig. 8. These high-level relationships that are extracted from low-level geometric data, such as length, width, region, direction, angle, etc. Since the low-level geometric data can be extracted directly from the user input through the ICADS graphic user interface, no line detection or feature extraction algorithms were needed. The spatial relationships between objects can be extracted by means of simple geometrical computation under different coordinate systems and generalized into an internal set of predicates. In computing the various spatial predicates, the key issue is to determine whether a point is located within an arbitrary polygon when the polygon's coordinate sequence is given. The approach taken by ICADS is to first triangulate the polygon, i.e., divide it into triangles [26] and use the convex combination property to test if the point is located in any of these triangles.

There are basically two main types of ICADS spatial relationships—binary or unary. Binary relationships are related to two "`ICADS Objects`" whereas unary relationships are only related to a single "`ICADS Object`." As described before, a "`ICADS Object`" can be either a room,

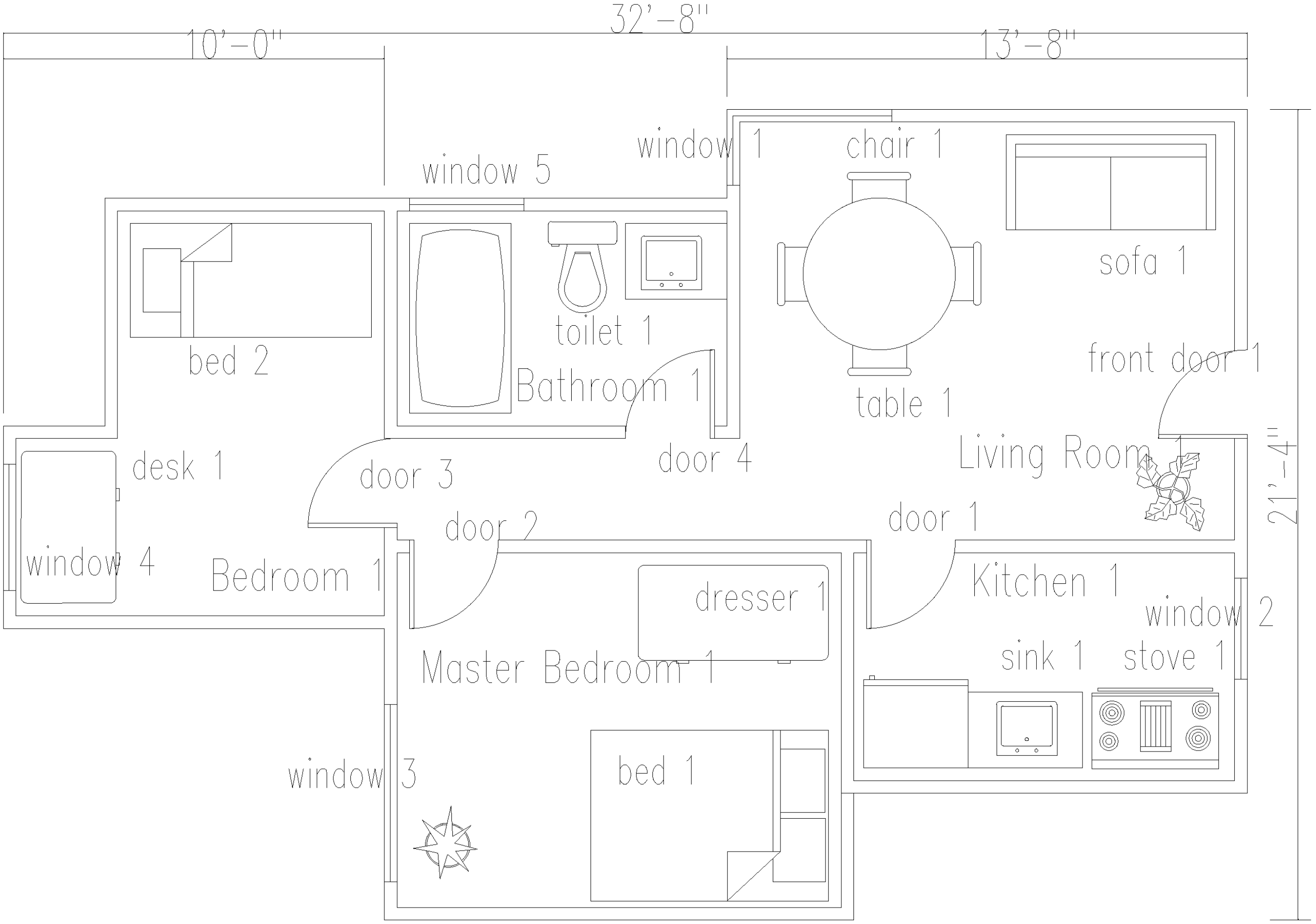


Fig. 9. A typical apartment floor plan analyzed by ICADS.

apartment, furniture, appliance, or stairway; i.e., any object within the ICADS domain is a subclass of the "`ICADS Object`." Each object may in turn be a part of another "`ICADS Object`." Each spatial relationship has a "type" that is one of the high-level predicates described before, such as **facing**, **visible**, **far**, **near**, or **congestion**. In addition, each spatial relationship may have a fuzzy qualifier.

With high-level predicates, computation used to match rules in the critic modules is greatly reduced and the resulting rule-base is easier to understand and maintain. Note that predicates may be object-dependent, which means the computation required to extract the predicate relationship might be different for different combination of objects.

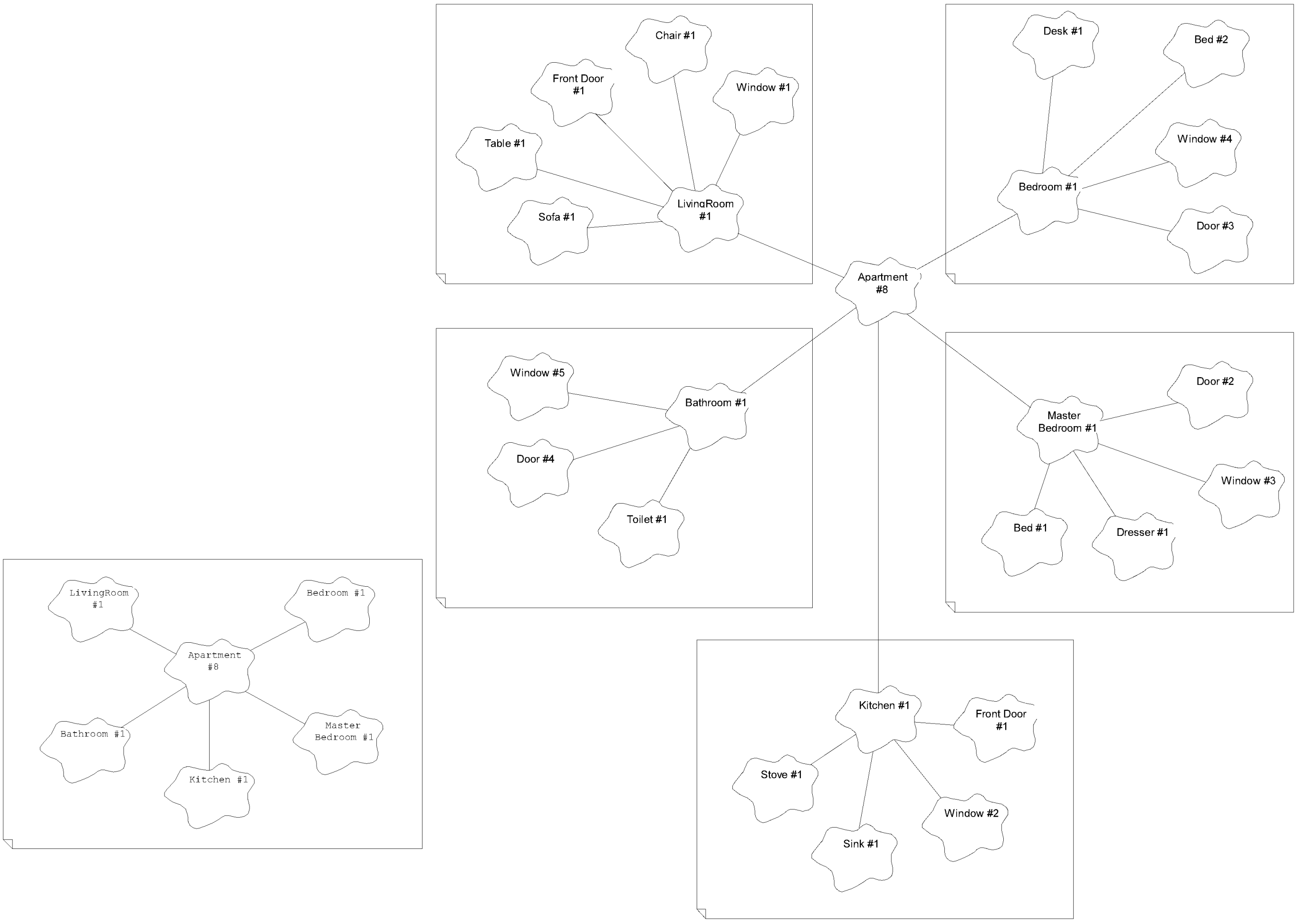
5.1 A Detailed Example

The following is a simple example to illustrate how ICADS processes a drawing to produce the internal knowledge representation. The drawing of a simple apartment will be used (see Fig. 9). For illustration, only a minimal amount of furniture and appliances are shown in this floor plan.

After the ICADS user develops and draws a floor plan using the ICADS object library, the "Spatial Analysis Component" extracts objects and their spatial relationships from this drawing. This component generates the instances of objects in the drawing, the a-part-of (APO) hierarchy, and the spatial relational links. The APO hierarchy for this example is shown in Fig. 3.

5.2 Forming Partition Space

The ICADS spatial predicates are divided into several levels: in the level of objects within a room, the rooms within an apartment, the apartments within a floor, etc. In our example floor plan, there are five rooms and 20 key objects. Instead of exhaustively enumerating all the spatial relationships among all the objects, the objects are first partitioned into groups according to the objects' APO hierarchy. Only objects within the same partition space have their spatial relationships enumerated. Fig. 10 is a diagram of the resulting partition space. This hierarchical partitioning corresponds nicely to the reference intervals introduced in [2], [3] when adopted from the time domain to the space domain. In our representation, *reference nodes* might be a better description. In ICADS, we have an advantage that the choice of reference is natural by the fact that objects are inside rooms and rooms are, in turn, inside houses. In the time domain, the reference has to be selected carefully from, say, important key events. This technique is also similar to partitioned semantic networks.



(a) (b)

Fig. 10. Partition space of objects in the example floor plan: (a) partition of rooms; (b) partition of objects within each room.

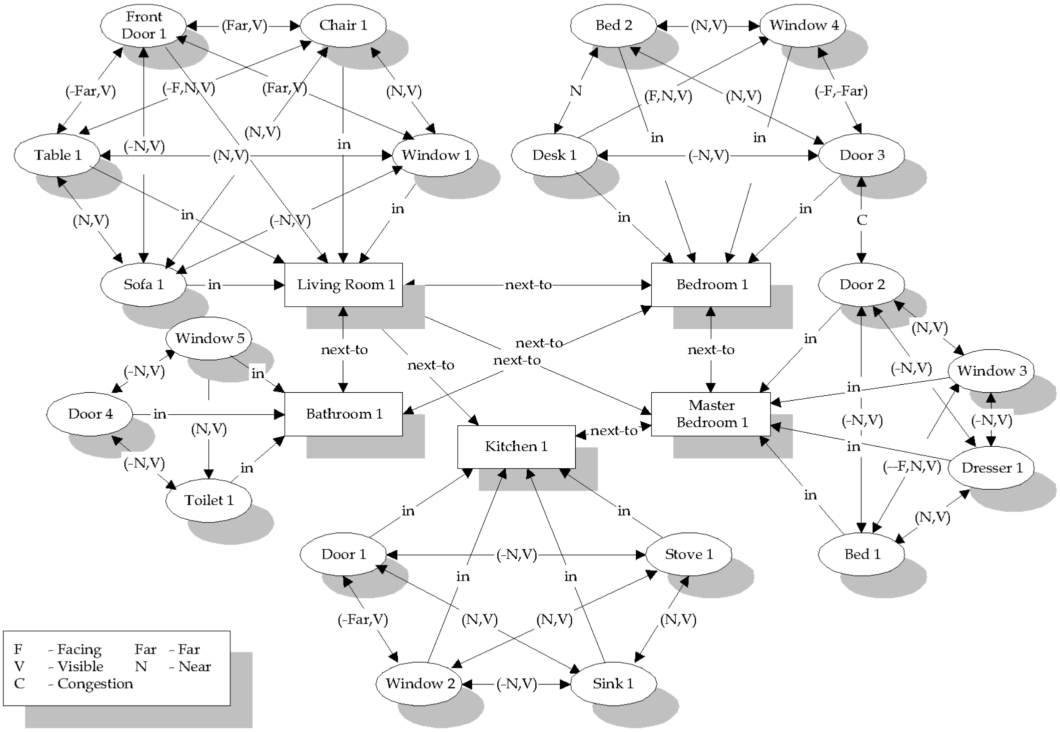


Fig. 11. ICADS *spatial relationship graph* of the example floor plan.

5.3 Enumerating Spatial Relationships

Once the partition space is defined, the “Spatial Analysis Component” of ICADS enumerates the spatial relationships among all the objects within the same partition space using a set of high-level spatial primitives. ICADS uses different sets of spatial primitives to represent spatial relationships within different levels of partition space. For example, in the level of rooms and furniture, there are spatial primitives that deal with concepts such as whether one object is **facing** another, how **near** or **far** one object is from another, and whether one object is **visible** from another. In addition, each primitive can be qualified with a fuzzy predicate, such as “object A is only **slightly facing** object B.” The definition of these primitives may depend on the context of the objects involved. In the level of rooms, ICADS uses spatial primitives such as whether a room is **next-to** another, the relative **position** of the room within an apartment, the relative **size** of a room, and the **connectivity** of the rooms [38].

Fig. 11 is a simplified diagram of a *spatial relationship graph* generated by ICADS from the example floor plan. Links labeled “in” refers to the APO hierarchy. In the *spatial relationship graph*, instead of representing each spatial relationship as a class instance, they are shown simply as a link and a label. Fuzzy qualifiers are represented as a prefix of “–,” “- -,” “+,” or “+ +” to the spatial relationship, e.g., “- - Far” means two objects are not too **far** from each other. In addition to spatial relationships that apply within a partition space, there are also certain relationships that must be defined across partitions, such as the “C” link in Fig. 11 between “Door 2” and “Door 3” that indicate that two doors, from different rooms and hence different partitions, are close to each other and might cause a problem.

The FPDX critic module reasons mainly with attributes stored in the objects and the APO hierarchy, whereas the IDX critic module reasons mainly with the spatial primitives.

6 THE FPDX CRITIC MODULE

In any modern city, there are various government building regulations that restrict how apartment buildings can be built. These regulations ensure the buildings are safe and comfortable to live in. For example, there are regulations that ensure there are adequate passages for fire escape, the stairways are wide enough, or there is enough sunlight in an apartment. In Hong Kong, there are hundreds of building regulations related to planning, construction, lifts, reuse storage chambers, ventilation systems, or fire safety. For our initial research, the FPDX critic module contains only knowledge related to fire escape routes and stairs [14], and sunlight requirements within a room [36].

These encoded rules are matched against the ICADS “Current Fact-Base” to verify that government regulations are met. If not, it will explain why there is a violation and propose solutions to the problems. Although contradicting rules should not exist in legal regulations, FPDX inherently handles them since each rule violation will be flagged, regardless whether they are contradicting, and highlighted to the user. Since there is a large number of regulations that deal with various aspects of building design and these regulations might change from time to time, it is very hard for an architect to creatively concentrate on designing a floor plan and at the same time try to adhere to these regulations. The main goal of the FPDX critic is to act as an architect’s regulation advisor in the designing process and thus freeing the architect to concentrate on his design work.

Fig. 12 is a typical floor plan analyzed by the FPDX critic module. FPDX will verify that the stairs satisfy government requirements and limitations, such as the maximum number of steps and dimensions. It will also verify that each room has a safe exit route. FPDX will then check each room to see if they meet government sunlight requirements. The syntax and structure of the FPDX rules are very similar to that of the KB-1 building safety rules found in the EKSPRO system [33].

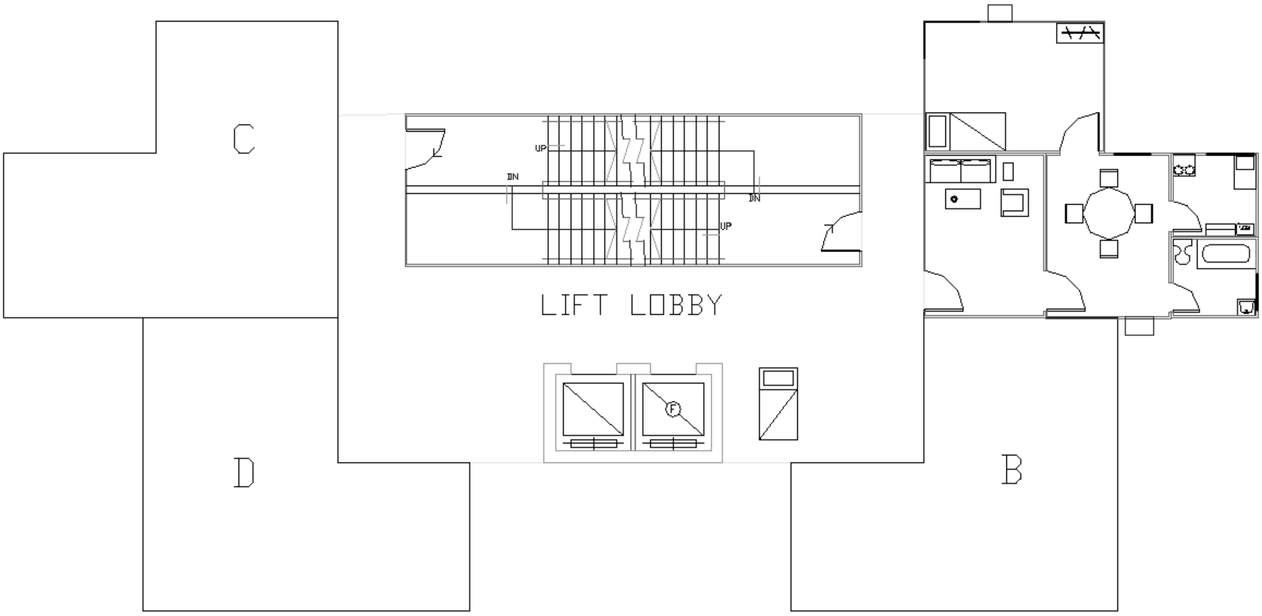


Fig. 12. Typical floor plan analyzed by the FPDX critic module.

The FPDX critic rule-base contains encoded interpretations of Hong Kong government regulations provided by the Buildings Ordinance Office and the Buildings and Lands Department. Currently, FPDX contains the following types of rules:

6.1 Dimensional Restrictions

There are numerous government restrictions on the minimum or maximum dimensions of various objects in a building. These are mainly to ensure the living environment is safe. Some examples:

FPDX Regulation F10b.1
The minimum width of each exit route is 900 mm.

FPDX Regulation F10b.2
The minimum width of each exit door is 750 mm.

6.2 Stairway Restrictions

Building regulations limit the maximum and minimum number of steps in a stairway. Examples:

FPDX Regulation F19.3.1
Each flight shall consist of not more than 16 risers nor less than two risers.

FPDX Regulation F19.6
No stair shall exceed 1,800 mm in width unless it is divided by a central handrail.

6.3 Window and Sunlight Requirements

In order to provide a healthy and comfortable living environment, there are building regulations that stipulate when a window is required and the minimum size of such a window. Example:

FPDX Regulation P30.1
Every room used for habitation shall be provided with natural lighting and ventilation by means of one or more windows.

6.4 An Example Using the FPDX Critic Module

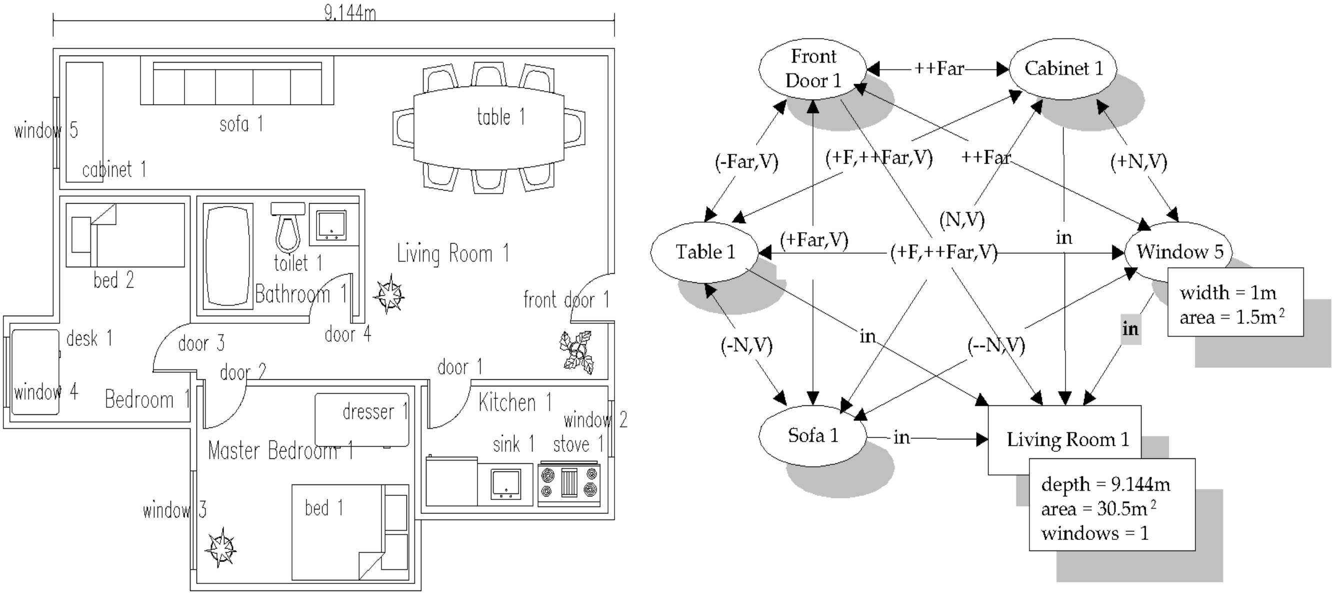
The following is a simple example to illustrate how the FPDX critic module verifies the window and sunlight regulations. Fig. 13a is the floor plan of an apartment with a somewhat strangely shaped and long living room. There is a window in the far end. Fig. 13b is part of the resulting ICADS *spatial relationship graph*. Note that the FPDX critic module will use mainly the APO hierarchy and the object's geometric data for its reasoning. The relevant nodes in this example are expanded to show the frame structure.

After evaluating all the FPDX rules, it finds three window-size related violations. The first two are related to the size of "window 5" in "living room 1." The window is too small and too far away from some part of the living room.

FPDX Regulation P30.2a.1
The aggregate superficial area of glass in the window(s) is not less than one-tenth of the area of the floor of the habitation room.

and

FPDX Regulation P32
No part of any room used for habitation shall be more than 9 m, measured within the room, from a prescribed window.



(a) (b)

Fig. 13. An example floor plan for the FPDX critic module: (a) a long living room; (b) ICADS spatial relationship graph.

The third rule that is violated is related to “bathroom 1” that does not have a window frame instance as part of its APO hierarchy:

FPDX Regulation P36.1
Every room containing a soil fitment or waste fitment shall be provided with a window.

To fix these problems, the architect must move “bathroom 1” to a side of the apartment where a window can be opened. The living room must either be shortened or extra windows should be added.

7 THE IDX CRITIC MODULE

Once a floor plan is designed and is found to fully meet all government regulations, the interior designer steps in to fill the apartment with furniture, fixtures, and appliances. The art of interior design involves creating an overall image or style for an apartment or building interior. It also involves selecting colors for the rooms, fabric for furniture and drapery, art work, and decorative accessories. Most importantly, it involves deciding where to place furniture and household objects within the rooms, i.e., how a room can be efficiently and comfortably utilized [37], [49]. Although much of interior design is an art that varies with individual style and taste, there are still some basic guidelines or rules to follow.

Our IDX critic module encodes these types of design rules and uses them to analyze an interior design plan, indicate design violations, and offer suggestions for improvements. IDX rules are similar to KB-2 rules of EKSPRO [33] which also uses Prolog to encode design knowledge. However, the rules encoded in IDX reasons with object placement within an apartment, whereas the KB-2 rules propose placement of heating, ventilation, or air conditioning systems. The IDX rule base currently has about 60 designs, or placement rules taken from common design principles. These rules can be divided roughly into the following categories; each describing a specific high-level relationship between objects:

7.1 Directions of Objects

This set of rules forbids objects from being placed face-to-face or requires some object to be placed within the facing scope of another object. Example:

IDX Rule R1
Condition: The kitchen should not face the bathroom.
Reason: People usually consider this not pleasant.
Suggestion: Relocate either doors.

7.2 Proximity of Objects

This imposes a restriction on how close objects should or should not be. An example:

IDX Rule B2
Condition: The bed should not be too close to the door.
Reason: Might be noisy and inconvenient.
Suggestion: Move bed away from door.

7.3 Spatial Relations of Rooms

People naturally think that some rooms should be or should not be next to particular rooms, for instance the bathroom should not be facing or inside the kitchen, or the kitchen should be next to the living room or dining room for convenience of serving meals. An example:

IDX Rule R14
Condition: Avoid having to pass through kitchen to get to toilet.
Reason: Inconvenient, potentially dangerous for children, and not healthy to have kitchen near toilet.
Suggestion: Relocate the door to the toilet.

7.4 Dimensions of Objects

The dimensions of objects will sometime affect the occupants; for example, too large a window in a small room is not welcome, since it might produce a large draft. In another case, a room should not appear to be too long and narrow, or else it looks empty and lonely. This overlaps in some way with regulations in FPDX that restricts the size of rooms based on window size and sets a minimum limit on the size of windows.

7.5 Locations of Rooms

This set of rules restricts how rooms should be arranged in the floor plan. An example:

IDX Rule R4
Condition: The master bedroom should be behind the central lines of the house, i.e., away from the main entrance.
Reason: People prefer a more private and quiet location for the bedroom.
Suggestion: Move master bedroom away from front part of house.

7.6 An Example Using the IDX Critic Module

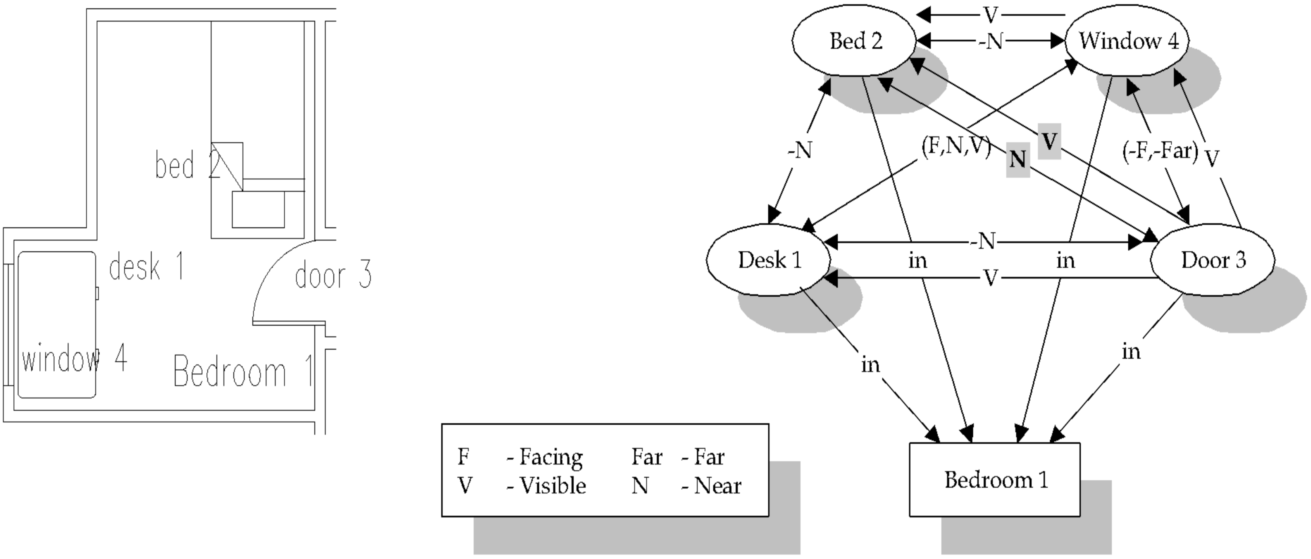
The following is a simple example to illustrate how the IDX critic module works. Fig. 14a is floor plan for a bedroom in which the bed faces the wall. Fig. 14b is part of the resulting ICADS *spatial relationship graph* generated by the “Spatial Analysis Component.”

Two particular spatial relationships are highlighted: “Bed2 is **near** to Door3.” and “Bed2 is **visible** from Door3.” Since the **visible** link is only unidirectional, Door3 is assumed to be **not visible** from Bed 2 because a closed-world logic is used.

When the IDX critic rules are applied to this simple graph, the highlighted spatial relationships cause two IDX rules to be fired:

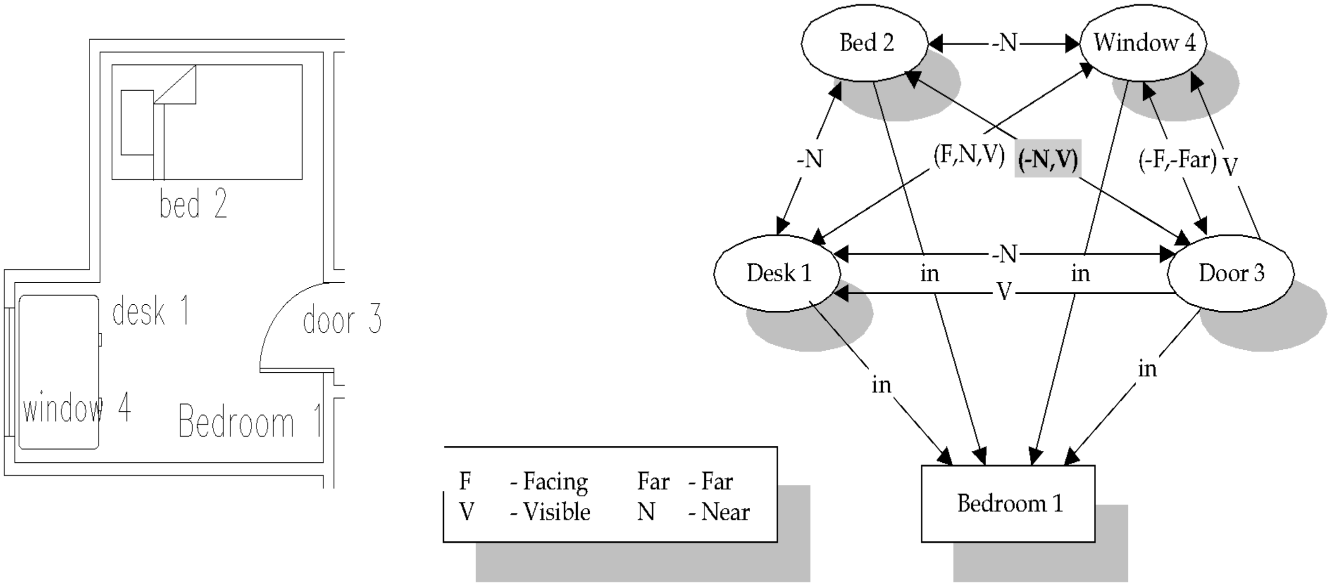
IDX Rule B1
Condition: Entrance to room is not visible from bed.
Reason: People usually like to know if someone is entering the room.
Suggestion: Rearrange bed OR add mirror.

and



(a) (b)

Fig. 14. An example that violates IDX critic rules: (a) bedroom floor plan; (b) ICADS spatial relationship graph.



(a) (b)

Fig. 15. Previous example with violation corrected: (a) bedroom floor plan; (b) ICADS spatial relationship graph.

IDX Rule B2

Condition: The bed should not be too close to the door.

Reason: Might be noisy and inconvenient.

Suggestion: Move bed away from door.

To remedy this situation, we simply follow the suggestions of IDX, i.e., rearrange the bed and move it away from the door. Fig. 15a is one possible solution. Bed 2 is now only **somewhat near** to Door 3 and Door 3 is now **visible** from Bed2. Fig. 15b is part of the resulting ICADS *spatial relationship graph.* The highlighted spatial relationships are those that have changed due to relocating the bed. Applying IDX critic rules again to this spatial graph no longer produces any violations.

8 CONCLUSION

This paper documents our research and development work in creating a more intelligent CAD system through “critic modules.” Although a lot of effort and development have gone into the ICADS, there are still many areas that can be further explored in a research project. The scope and depth of knowledge needed to fully understand all types of government regulations and design principles are quite large. Therefore, we have limited our investigation to regulations and design rules that deal only with two-dimensional reasoning and only with objects within one floor or within one apartment. As part of the growth of our research, we are expanding the scope of ICADS to other types of knowledge. Fortunately, with the ICADS modular design, we can incrementally widen the scope of knowledge of ICADS by the gradual addition of more critic modules.

We also intend to supplement knowledge found in critics with case-based knowledge [19], [20], [21], [32], [42] of previously created "good" floor plan designs. These cases can be used to generate a skeleton "sketch" of a floor plan given the current physical restrictions and design requirements. This sketch can then be enhanced by an architect and reviewed by our critic modules.

ACKNOWLEDGMENTS

This research was supported, in part, by a Hong Kong UGC Earmarked Grant and a Strategic Research Grant from the City University of Hong Kong. The authors would like to thank Wai Lun Fung for his contributions to IDX and Yvonne Mei-yee Yau for her efforts in FPDX.

REFERENCES

[1] G. Adorni, A. Boccalatte, and M. Dimanzo, "Cognitive Models for Computer Vision," *Proc. Ninth COLING*, July 1982.

[2] J.F. Allen, "Maintaining Knowledge about Temporal Intervals," *Comm. ACM*, vol. 26, no. 11, Nov. 1983.

[3] J.F. Allen, "An Interval-Based Representation of Temporal Knowledge," *Proc. IJCAI-81*, pp. 221-226, 1981.

[4] O. Akin, "How Do Architects Design?" *Artificial Intelligence and Pattern Recognition in Computer Aided Design*, Latombe, ed., North-Holland, pp. 806-809, 1978.

[5] D.T. Bar, "Practical Consequences of Formal Definitions of Inheritance," *J. Object-Oriented Programming*, pp. 43-49, Aug. 1992.

[6] A.H. Bond, "A Predicate Logic Approach to CAD/CAM Modeling," *AI EDAM*, vol. 6, no. 1, pp. 39-58, 1992.

[7] G. Booch, *Object-Oriented Analysis and Design with Applications*, second edition, Benjamin/Cummings, 1994.

[8] R.A. Brooks, *Model-Based Computer Vision*, UMI Research Press, 1981.

[9] D.M. Chorafas, *Knowledge Engineering: Knowledge Acquisition, Knowledge Representation, the Role of the Knowledge Engineer, and Domains Fertile to AI Implementation*, Van Nostrand Reinhold, 1990.

[10] E. Davis, "The Mercator Representation of Spatial Knowledge," *Proc. IJCAI 83*, pp. 295-301, 1983.

[11] E. Davis, "A Logical Framework for Commonsense Predictions of Solid Object Behaviour," *AI in Eng.*, vol. 3, no. 3, 1983.

[12] E. Davis, *Representations of Commonsense Knowledge*, Morgan Kaufmann, 1990.

[13] M. Dimanzo, G. Adorni, and F. Giuhchiglia, "Reasoning About Scene Descriptions," *Proc. IEEE*, vol. 74, no. 7, July 1986.

[14] *Code of Practice on Provision of Means of Escape in Case of Fire and Allied Requirements*, Building Ordinance Office, Buildings and Lands Dept., Hong Kong, 1986.

[15] G. Fischer and K. Nakakoji, "Making Design Objects Relevant to the Task at Hand," *Proc. AAAI-91*, 1991.

[16] G. Fischer and T. Mastaglio, "A Conceptual Framework for Knowledge-Based Critic Systems," *Decision Support Systems*, vol. 7, pp. 355-378, 1991.

[17] F.J. Garijo and L.A. de Garrido, "A Knowledge Based System for House Design," *Proc. 1988 IEEE Int'l Conf. Systems, Man, and Cybernetics*, 1988.

[18] F. Giunchiglia, C. Ferrari, P. Traverso, and E. Trucco, "Understanding Scene Descriptions by Integrating Different Sources of Knowledge," Univ. of Genoa, Technical Report MRG/DIST No. 9101-02, Jan. 1991.

[19] A.K. Goel, J.L. Kolodner, M. Pearce, R. Billington, and C. Zimring, "Towards a Case-Based Tool for Aiding Conceptual Design Problem Solving," *Proc. Workshop Case-Based Reasoning (DARPA)*, Washington, D.C., Morgan Kaufmann, 1991.

[20] A.K. Goel, "Integrating Case-Based and Model-Based Reasoning," *AI Magazine*, vol. 13, no. 2 , pp. 50-53, 1992.

[21] K.J. Hammond, "Planning and Goal Interaction: The Use of Past Solutions in Present Situations, *Proc. Third Nat'l Conf. Artificial Intelligence*, pp. 127-138, 1983.

[22] V.D. Hunt, *Artificial Intelligence and Expert Systems Sourcebook*, Chapman and Hall, 1986.

[23] *Notes on Architecture*, Information Design Inc., Crisp Publications, 1990.

[24] *IF/Prolog Manual*, version 4.1, InterFace Computer, GmbH, 1992.

[25] D. Jain and M.L. Maher, "Combining Expert Systems and CAD techniques," *Proc. Artificial Intelligence Developments and Applications, Australian Joint Artificial Intelligence Conf.*, North-Holland, p. 65, 1987.

[26] X. Kong, H. Everett, and G. Toussaint, "The Graham Scan Triangulates Simple Polygons," *Pattern Recognition Letters*, vol. 11, no. 11, pp. 713-716, 1990.

[27] B. Kuipers, "Representing Knowledge of Large-Scale Space," MIT AI Lab, Technical Report No. TR-418, July 1977.

[28] S.-Y. Lye and H.K. Ho, "Knowledge-Based CAD System for Protective Packaging Design," *AI EDAM*, 1991.

[29] B. MacKeller and J. Peckham, "Representing Design Objects in Sorac," *Artificial Intelligence in Design '92*, J.S. Gero, ed., Kluwer Academic Publishers, pp. 201-219, 1992.

[30] J. Malik and T. Binford, "Reasoning in Time and Space," *Proc. IJCAI-83*, pp. 333-345, 1983.

[31] R. Oxman and J.S. Gero, "Using an Expert System for Design Diagnosis and Design Synthesis," *Expert Systems*, vol. 4, no. 1, 1987.

[32] R.E. Oxman, "Case-Based Design Support: Supporting Architectural Composition Through Precedent Libraries, *J. Architectural Planning Research*, 1993.

[33] L.F. Pau and S.S. Nielsen, "A Knowledge-Based System for Computer-Aided Architectural Design for Energy Savings and Thermal Comfort," *AI EDAM*, vol. 4, no. 2, pp. 71-88, 1990.

[34] R.N. Pelavin, J.F. Allen, H.A. Kautz, and J.D. Teneberg, *Reasoning About Plans*, Morgan Kaufmann, 1990.

[35] D.N Perkins, *Knowledge As Design*, Erlbaum, 1986.

[36] *Building (Planning) Regulations*, Buildings Ordinance Office, Buildings and Lands Dept., Hong Kong, 1984.

[37] S. Rossbach, *Interior Design With Feng Shui*, Rider, 1987.

[38] R.C. Schank and C.J. Rieger III, "Inference and the Computer Understanding of Natural Language," *Artificial Intelligence*, 1974.

[39] S.C. Shapiro, *Encyclopedia of Artificial Intelligence*, John Wiley and Sons, 1990.

[40] E. Shaviv, "Layout Design Problems: Systematic Approaches," *Computer-Aided Architectural Design Futures*, A. Pipes, ed., pp. 28-52, 1985.

[41] H. Shimodaira, "Basic Structure of a Building Model for Representing and Using Knowledge of Buildings in CAAD Systems," *Artificial Intelligence in Design '92*, J.S. Gero, ed., pp. 241-263, 1992.

[42] S. Slade, "Case-Based Reasoning: A Research Paradigm," *AI Magazine*, pp. 42-55, 1991.

[43] K.W. Smithies, *Principles of Design in Architecture*, Van Nostrand Reinhold, 1981.

[44] C. Stanfill and D.L. Waltz, "The Memory-Based Reasoning Paradigm," *Proc. Case-Based Reasoning Workshop*, J. Kolodner. ed., pp. 414-424, 1988.

[45] G.T. Toussaint and J.A. McAlear, "A Simple O(n log n) Algorithm for Finding the Maximum Distance between Two Finite Planar Sets," *Pattern Recognition Letters*, vol. 1, pp. 21-24, 1982.

[46] G.T. Toussaint and B.K. Bhattacharya, "Optimal Algorithms for Computing the Minimum Distance between Two Finite Planar Sets," *Pattern Recognition Letters*, vol. 2, pp. 79-82, 1983.

[47] P.R. Wallach, D.E. Hepler, and D.J. Hepler, *Architecture Drafting and Design*, McGraw-Hill, 1993.

[48] D. Waltz and L. Boggess, "Visual Analog Representations for Natural Language," *Proc. IJCAI-79*, 1979.

[49] D. Walters, *Feng Shui Handbook*, Aquarian Press, 1991.

[50] I. White, *Using the Booch Method*, Benjamin/Cummings, 1994.

[51] W. Wolf, "Object-Oriented Implementation Issues in an Experimental CAD System," *Software—Practice and Experience*, vol. 22, no. 4, pp. 287-304, Apr. 1992.

Hon Wai Chun received the BS degree from the Illinois Institute of Technology in 1981, and the MS and PhD degrees in electrical engineering from the University of Illinois at Urbana-Champaign in 1983 and 1987, respectively. He was a senior scientist at Ascent Technology, Boston, before joining the Information Engineering Department of the Chinese University of Hong Kong as a lecturer in 1989. Dr. Chun joined the Department of Electronic Engineering at the City University of Hong Kong in March 1994 and is now an associate professor. He is jointly a consultant for CityU Consultants Limited. His research interests are in scheduling, resource allocation, personnel rostering, and knowledge representation. He is a member of the IEEE.

Edmund Ming-Kit Lai received the BE with honors and PhD degrees in electrical engineering from the University of Western Australia in 1982 and 1992, respectively. Dr. Lai was a faculty member of the Department of Electrical and Electronic Engineering at the University of Western Australia between 1984 and 1990. From August 1990 to December 1994, he was with the Chinese University of Hong Kong, first as assistant lecturer, then as a lecturer in the Department of Information Engineering. In 1995, he was a research fellow with the Australian Telecommunications Research Institute in Perth, Western Australia. In January 1996, he joined the Edith-Cowan University as a lecturer in the Department of Electronics and Communication Engineering. His current research interests include applications of artificial intelligence, telecommunication networks, and signal processing. He is a member of the IEEE.