KICS: A Knowledge-Intensive Case-Based Reasoning System for Statutory Building Regulations and Case Histories

April 1993

Soon-Ae Yang† **Dave Robertson**†
†Dept of Artificial Intelligence
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN
United Kingdom

John Lee‡
‡EdCAAD, Dept of Architecture
University of Edinburgh
20 Chambers Street
Edinburgh EH1 1JZ
United Kingdom

To be presented at the Fourth International Conference on Artificial Intelligence and Law, Amsterdam, The Netherlands, June 15–18, 1993.

KICS: A Knowledge-Intensive Case-Based Reasoning System for Statutory Building Regulations and Case Histories

Soon-Ae Yang† **Dave Robertson†**
†Dept of Artificial Intelligence
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN, UK

John Lee‡
‡EdCAAD, Dept of Architecture
University of Edinburgh
20 Chambers Street
Edinburgh EH1 1JZ, UK

Abstract

There have been several knowledge-based systems for statutory building regulations during the last decade, such as Fenves *et al* 's systems using the SASE model, Stone and Wilcox's system using a rule-based approach, and Waard's system using Cornick *et al* 's model-based approach. However, they take into account only one side of building regulations, considering them only in the context of design systems and ignoring the existence of case histories. Building regulations are also part of a legal system and have characteristics of law. In this paper, we propose a Knowledge-Intensive Case-based reasoning System which can be used for the retrieval and maintenance of building regulations and case histories. First, we propose a unified knowledge representation scheme for both statutory building regulations and case histories. Second, we describe the retrieval of regulations information, which uses the notion of implied similarity as well as structural mapping. Finally, we describe knowledge acquisition from case histories, which is guided by knowledge gained from statutory regulations and case histories.

1 Introduction

Building regulations prescribe standards for a defined set of issues affecting building design and construction and are part of a statutory building control system. There have been several knowledge-based systems to help the authoring and consultation process of building regulations during the last decade, such as Fenves *et al* 's systems [7, 8, 12] using the SASE model, Stone and Wilcox's system [24] using a rule-based approach, and Waard's system [4, 5] using Cornick *et al* 's model-based approach [3].

However, they take a narrow view of building regulations. They consider building regulations only as statements of what requirements building elements satisfy in terms of their attributes and relations between building elements. They take into account only one side of building regulations, considering them only in the context of design. Building regulations have characteristics of legal information as well as of information needed in design. As part of a legal system, there is another source of information, case histories, which must be available in the authoring and consultation process of building regulations. We will call statutory building regulations and standards building *regulations*, and we will describe statutory building regulations including standards and case histories collectively as building *regulations information* throughout this paper.

As legal knowledge, building regulations information deals with several levels of concepts, from very abstract to very detailed and concrete. Abstract concepts provide intentions that must be satisfied in the context of law [6, 16], or design goals that must be achieved in the context of design [14, 23, 9, 15]. More detailed and concrete concepts provide requirements or design solutions with which buildings must comply in particular circumstances to achieve a certain intention or goal. A building needs to comply with certain requirements only when a certain part of a building is matched with circumstances in which those particular requirements are stipulated. However, deviation from requirements described in detailed and concrete terms is allowed in exceptional or unexpected circumstances, provided that the intention or goal can still be achieved [27, 6, 16, 25, 17]. Cases which allow such deviation must be available for the consulation of new cases which occur in similar circumstances or in the revision of statutory regulations. We regard legal rules as the results of the accumulation and generalisation of knowledge from case histories. [1]

Building regulations information deals with various issues (intentions or design goals) [14, 15]. It is very rare for any part of a building to serve only one purpose. The window is required to let in daylight and sunlight, to provide a view while retaining privacy, and to offer natural ventilation. The window in the external wall poses problems of structural stability, heat loss and noise transmission. Requirements for different issues may complement (e.g., letting in daylight and offering natural ventilation), inhibit (e.g., letting in daylight and retaining privacy) or have no effect upon each other (e.g., letting in daylight and noise transmission) [14]. Different standards may be given in the same criterion (e.g., the size of the window) to resolve different issues. Therefore, when a criterion is evaluated for requirements in an issue, all requirements described in the same criterion with respect to other issues must be checked to ensure that there is no conflict.

In this paper, we propose a **K**nowledge-**I**ntensive **C**ase-based reasoning **S**ystem (KICS) which makes use of two classes of AI techniques, case-based reasoning and machine learning.

First of all, we describe a unified knowledge representation scheme for both statutory building regulations and case histories. In previous approaches to knowledge representation in the domain of law, two different knowledge representation schemes have been used to represent statute and case histories: mostly production rules for statute [30, 21, 22, 25, 19], and frame-based or semantic network representations for case histories [28, 29, 19, 1, 2]. In our approach, legal rules are regarded as the results of the accumulation and generalisation of knowledge from case histories, and rules from both statute and case histories are represented as models. To represent hierarchical and multidimensional building regulations information, an abstraction hierarchy is constructed for each particular goal or intention.

[1] The reader must not confuse production rules with legal rules, which we will refer to many times in this paper. Production rules are used to represent knowledge in expert systems. Production rules have a simple *If-Then* format, which is interpreted as "*If* certain conditions are known to hold *Then* draw the stated conclusion or take the stated action", and a forward and backward chaining method is used to reason with these rules. On the other hand, legal rules are statements of what should be done in certain circumstances in the context of a legal system. Legal rules can be represented in many ways in computer, and production rules are just one of them.

Second, we describe the retrieval of models relevant to the given case. The system first prunes irrelevant models by category of buildings and then selects models applicable to the given case by assessing similarity. In similarity assessment, in addition to the notion of structural mapping which has been used in [11, 10, 13, 26, 2], we introduce the notion of implied similarity which assesses similarity using domain theory. Evaluation of cases against retrieved models is also described.

Finally, we describe knowledge acquisition from case histories. The case must be decided as valid for acquisition before being added to the knowledge base. The system then generalises cases and integrates them into existing models. This process is guided by experience, that is knowledge gained from statutory regulations and previous cases and accumulated in the model knowledge base.

2 Knowledge Representation

Development of regulations information is very similar to designing an object. Legislators create regulations which achieve certain intentions; regulations provide possible ways of achieving such intentions (or constraints) at different levels of concepts concerning various issues. One view of design is as follows [14, 23, 9, 15]. Design begins with a set of goals or objectives that represent the highest level design problems (intentions in regulations) and these problems are refined into subproblems (constraints in regulations). A problem may be decomposed into several subproblems or specialised using more specific terms. Those refined subproblems are themselves partial design solutions which serve as a major stimulus for suggesting to the designer what problem should be attended to next [23].

We can start with a set of intentions or goals and build a knowledge base which contains structures of intentions and constraints by decomposing and specialising the initial set of intentions or goals. New knowledge from case histories can be added to the knowledge base by modifying the contents and structure of the knowledge base. In our system, knowledge from statutory regulations and case histories will be represented as abstraction hierarchies of *models* in the Model Knowledge Base (MKB). Each abstraction hierarchy will represent a structure of a particular intention and its constraints.

In addition to the MKB, there will be the Domain Knowledge Base (DKB). The DKB will include vocabulary used to describe models and cases and knowledge needed in the retrieval of relevant cases and the acquisition of cases.

2.1 The Model Knowledge Base

The model knowledge base (MKB) will consist of *models*, which are used to represent knowledge from two sources in one form: rules from statutory regulations and rules from case histories. Rules from statutory regulations act as strong constraints on new cases in the sense that a case must comply with regulations which are applicable to it. We can discover how regulations used to be interpreted and are interpreted in case histories. Each case provides a way of interpretating a particular set of regulations under particular circumstances.

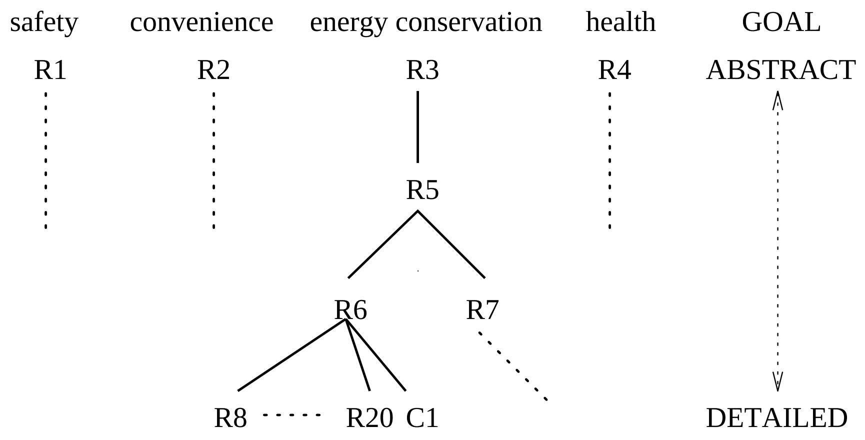

Figure 1: Interpretation Hierarchies

At the time of legislation, it is impossible to anticipate all possible circumstances in which a case can arise after legislation. Therefore new rules are set out when a case arises in circumstances unanticipated at the time of legislation. Furthermore, if it is unreasonable in the particular circumstances for specific rules to be applied and these rules can be met in a less onerous way, existing rules may be relaxed. As the interpretation of written regulations changes in time, all rules set out in case histories may or may not apply to new cases. In this sense, rules set out in previous cases act as weak constraints to new cases. However, how cases were handled in similar circumstances needs to be considered to decide the conformance with regulations for the new case.

Rules from statutory regulations and case histories can be described at different levels. First, rules in particular circumstances are described using more abstract terms, which have broader meaning and can be interpreted in different ways. Each of these abstract rules can serve as a goal or an intention which provides a reason for existence of such rules. One or more rules can be generated to provide ways of achieving a specific goal. In other words, refined rules explain how the original rule is to be interpreted. There may be other ways of achieving the specified goal but certain rules are chosen and set out as regulations. Refined rules are usually described using more detailed and concrete concepts. Since there may be different rules to be satisfied in slightly different circumstances, each refined rule may apply, not to the same circumstances to which the original rule should apply, but to the subdivided or reduced circumstances. Each of these refined rules can be refined again in as much detail as necessary.

For example, the Scottish building regulations[20] implicitly require that all buildings be built to ensure the safety (R1) and convenience (R2) of people in and around buildings and the conservation of fuel and power (R3), and not to harm the health of people living in or visiting buildings (R4). These four rules (R1–R4) can be considered four intentions for which the regulations are legislated. The regulations provide an interpretation of the rule R3 as: all buildings should be built to limit heat loss (R5). R5 is again interpreted as: reasonable provision should be made for the building fabric (R6) and for the building services (R7).

Again, a building is considered to satisfy R6 if floors, walls and roofs are constructed so that they do not exceed the appropriate U-values (R8-R14), and if the total areas

M15:

M_c:	dwelling, industrial_building, storage, warehouse
M_{cd}:	window(A), type(A,single_glazed), wall(B), type(B,exposed), in(A,B)
M_{sd}:	percentage_area(A,C), C $\leq$ 15
M_s:	strong

M19:

M_c:	dwelling
M_{cd}:	rooflight(A), type(A, single_glazed), roof(B), in(A,B)
M_{sd}:	percentage_area(A,P), P $\leq$ 15
M_s:	strong

Figure 2: Model M15 and M19

of single-glazed openings in those walls and roofs do not exceed the prescribed areas (R15-R20). R6 does not apply to all floors, walls and roofs and the regulations prescribe different U-values and different areas of openings in different situations. The regulations also prescribe different U-values and areas of openings to those components of different categories of buildings.

Rules should apply to buildings which fall into certain situations in which these rules are stipulated. However, as mentioned earlier in discussing case histories, new rules are set out in some unexpected or slightly different circumstances and existing rules are relaxed in exceptional circumstances.

Here is an example. Suppose that statutory regulations prescribe the total area only for single-glazed openings (R15). When the window is double-glazed, it is unreasonable to apply the same rule. In such cases, the authority will permit the construction of a larger window (C1), for instance, the maximum 17 percent of the total floor area, provided that this is reasonable provision to limit heat loss (i.e., the building satisfies R6, consequently, R5 and R3). Such cases can be considered as providing another interpretation of R6. Figure 1 shows the structure of rules as a hierarchy. Note that in the top and second levels of the hierarchy rules are applied to a building, in the third level the building fabric which is part of a building, and in the fourth level floors, walls, roofs and openings which are part of the building fabric. In other words, as levels go down, components to which the rule is to apply become more concrete and therefore the scope of application of rules becomes more restricted.

Each rule from statutory regulations and case histories will be represented as a *model*. A model M consists of four parts: $M = (M_c, M_{cd}, M_{sd}, M_s)$ where M_c is the category to which the model applies, M_{cd} is the description of key features of circumstances to which the model applies, M_{sd} is the description of (required or proposed) solutions, and M_s is the strength ("strong" or "weak"). Circumstances can be described in terms of objects, their attributes and relations between objects. Solutions can be described in

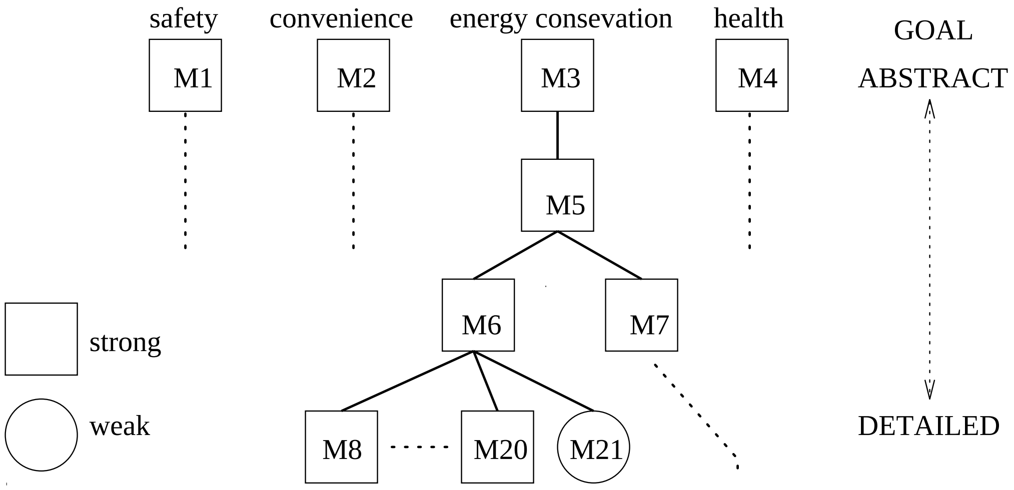

Figure 3: Abstraction Hierarchies of Models

terms of attributes or relations between objects usually but sometimes not included in the description of circumstances. The strength will be given as “strong” if the model is taken from statutory regulations, or as “weak” if the model is taken from case histories.

The category description of a parent model always subsumes the category description of all child models. In other words, if a model does not apply to a case because the category in this model does not include the category described in the given case, this means that descendant models of this model also do not apply to the given case. The category description together with such hierarchical structure of models will serve as effective indices in the retrieval of relevant models (described in Section 3).

For example, R15 and R19 in Figure 1 can be represented as M15 and M19 respectively as shown in Figure 2. [2]

Once all rules are converted into models, an abstraction hierarchy can be built from the interpretation hierarchy of rules. It is also possible to construct several hierarchies, each of which represents a hierarchy of one topic. For example, four hierarchies can be built for four different subjects (i.e., R1 to R4). Figure 3 shows model hierarchies constructed from interpretation hierarchies in Figure 1. It is assumed that if there are two models in different levels the model in the upper level has greater strength than the model in the lower level.

However, exceptions to a rule are sometimes permitted to a particular category of buildings and this information is inherited along the abstraction hierarchy. For example, limited life buildings which are used for dwelling do not need to satisfy the model M3 in Figure 3 and therefore do not need to satisfy all descendant models of M3 either. Such inheritance mechanisms need to be implemented in the proposed knowledge representation scheme.

[2] In the description of models and cases throughout this paper, arguments starting with an uppercase character are variables and arguments starting with a lowercase character are constants.

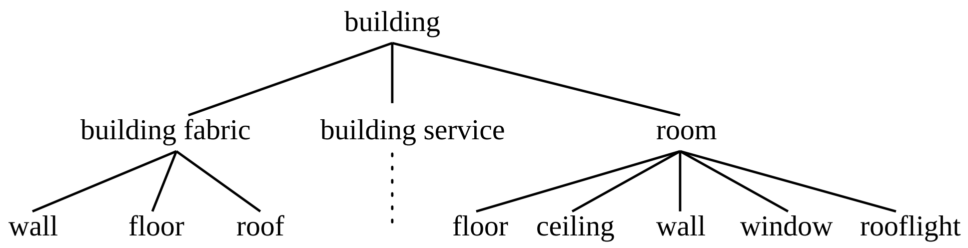

Figure 4: Aggregation Relations

2.2 The Domain Knowledge Base

Knowledge in the DKB will be used to classify a given case, i.e., to find models whose category and circumstance descriptions are most similar to those of the given case. All vocabulary used to describe models will be included in the DKB: objects and their synonyms, properties which an object can have, and relations which can exist between objects.

Specialisation and aggregation relations will be used to represent the object hierarchies. For example, components in a building can be represented using aggregation relations as in Figure 4.

Objects can relate to more general objects by specialisation relations. For example, specialisation hierarchies of components in a building can be constructed as in Figure 5. The category of buildings can also be represented using specialisation relations.

Knowledge in the DKB will also be used in the similarity test (described in Section 3.2.2) to reveal similarities which cannot be found in the surface description but can be inferred from the given description of the circumstances using domain knowledge. For example, if the circumstance description includes "there is a window in a bedroom" and the system fails to find models whose circumstance description matches, then this can be rephrased as "there is a wall in the bedroom and the window is in that wall" and the system will try to match again. Here are three rules which can be used to rephrase the given circumstance description:

- A wall X is exposed, then X is also external.
- A window X is in the roof, then X is a rooflight.
- There is a window X in a bedroom, then there is a wall Y in the bedroom and X is in Y.

3 Retrieval of Relevant Models

A new case C will be represented in the same way as models in the MKB without the strength: $C = (C_c, C_{cd}, C_{sd})$ where C_c is the category, C_{cd} is the circumstance description, and C_{sd} is the proposed solution. A new case C will be given with a specific goal G and the system will retrieve relevant models in the model hierarchy of the specified goal H_G.

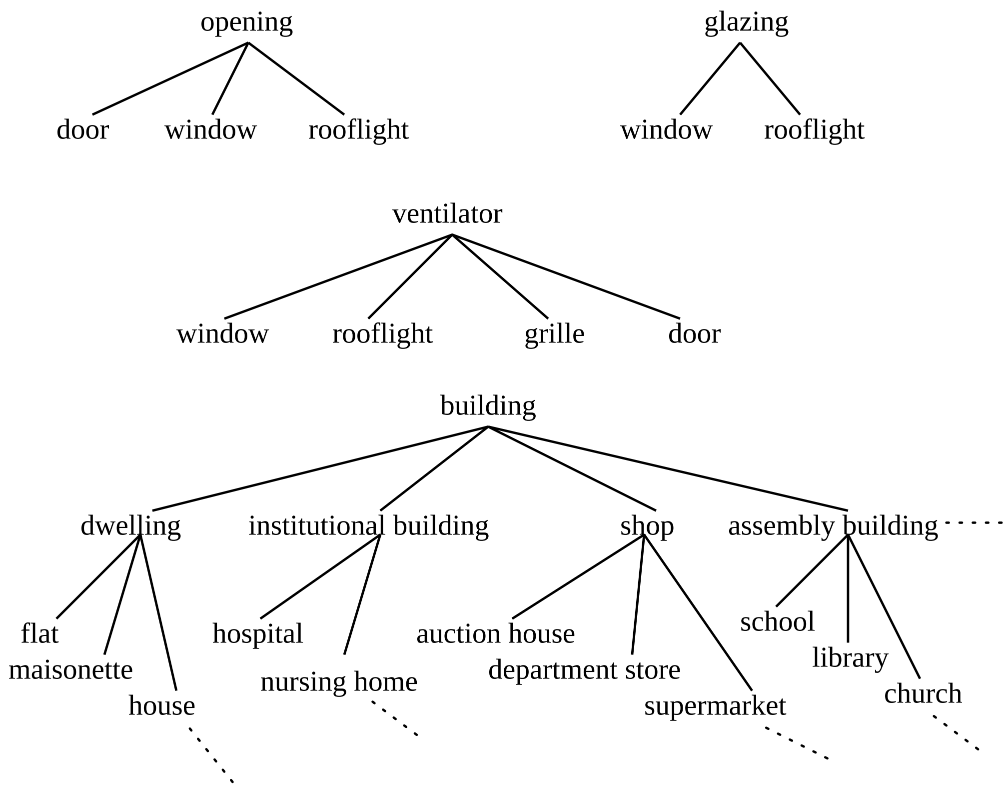

Figure 5: Specialisation Relations

This will be carried out in two phases:

1. retrieve models whose category description subsumes the category description of the given case

2. among retrieved models, select models whose circumstance description is similar to the circumstance description of the given case

3.1 Retrieval by Category

The system will compare C_c of C and M_c of models in H_G to retrieve models whose category description M_c *subsumes* the category description C_c of the given case.

M_c is said to *subsume* C_c when:

- objects in M_c and C_c are matched; or

- objects in M_c are matched with objects in the higher level of the specialisation hierarchy than objects in C_c

The system will start by comparing C_c with the root model of H_G. The system will go down until the lowest level L in which M_c subsumes C_c is found. Once the system finds a model whose M_c does not subsume C_c, the system will stop going down to descendant models of the current model. Only models in the path from the root

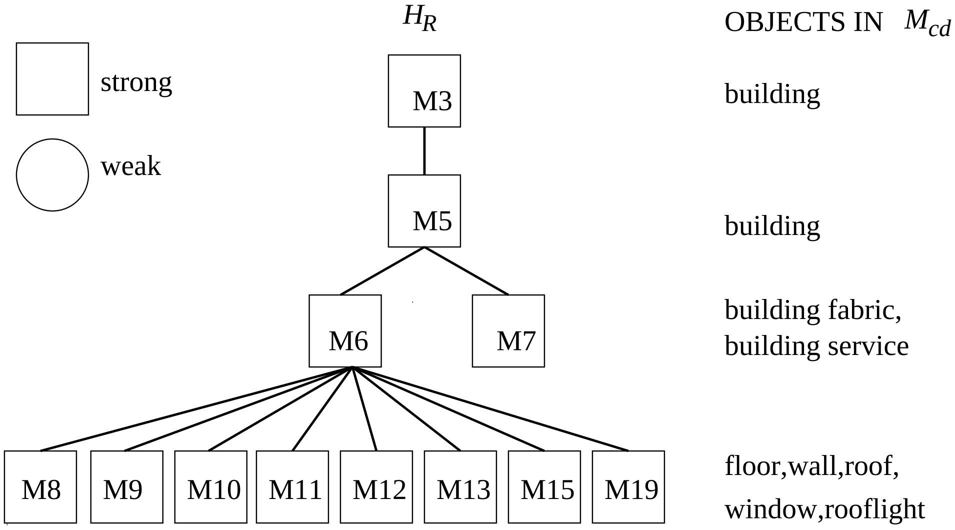

Figure 6: Hierarchy of Retrieved Models

model to the parent model of the current model will be marked "relevant". The system will continue comparing the remaining models in H_G until there is no model to compare. The system will then retrieve models marked "relevant". Retrieved models will again constitute a hierarchy H_R which is a subhierarchy of H_G.

For example, suppose that a case C2 is given as follows with the goal "energy conservation".

C2:
C_c: house
C_{cd}: window(a), type(a,single_glazed),
wall(b), type(b,exposed), in(a,b)
C_{sd}: percentage_area(a,14.5)

Based on the specialisation relations shown in Figure 5, models whose category description includes "building", "dwelling" and "house" will be retrieved. The system will come up with models $H_R = \{$M3, M5, M6, M7, M8, M9, M10, M11, M12, M13, M15, M19$\}$. The hierarchy of retrieved models H_R is shown in Figure 6.

3.2 Selection of Models by Similarity of Circumstances

Among models in H_R, the system will select models whose circumstance description M_{cd} is *similar* to the circumstance description C_{cd} of the given case. Like M_{cd}, C_{cd} is described in terms of objects, their attributes and relations between objects. Similarity will be measured by comparing those objects, their attributes and relations between those objects. This will be carried out in two phases:

1. find the right level of models in H_R
2. among models in the right level in H_R, select models whose circumstance description is similar to the circumstance description of the given case

3.2.1 The Right Level of Models

First, the system will look for the *right* level in H_R. The *right* level means the level in which objects in M_{cd} and objects in C_{cd} are on the same level of aggregation hierarchies so that objects in M_{cd} and objects in C_{cd} are comparable.

For example, in the case C2 in the previous section, the circumstances are described in terms of "windows" and "walls". The system will begin to compare these objects with M_{cd} of the root model in H_R shown in Figure 6, and go down until the system reaches the fourth level on which M_{cd} is described in terms of "floors", "walls", "roofs", "windows" and "rooflights". The models on the fourth level, H_{R_4} = {M8, M9, M10, M11, M12, M13, M15, M19} will be retrieved for the next process, the similarity test.

3.2.2 The Similarity Test

Once the right level are found, the system will compare the details of circumstance descriptions. A model which satisfies any of the following *similarity criteria* will be selected.

- literal similarity
- analogy
- semantic similarity
- embedded similarity
- implied similarity

First, the system will try to find model(s) whose M_{cd} is similar to C_{cd} of the given case by the first four criteria (structural mapping). If this attempt fails, the system will try to bring out similarity hidden under the surface description of C_{cd} (implied similarity). The system will transform C_{cd} into an alternative description using domain knowledge and repeat the similarity test with this alternative description.

Literal Similarity: All objects and their attributes are matched and all relations between objects are also matched. This has been called *literal similarity* by Gentner [11, 10] or *structural consistency* by Holyoak and Thagard [13, 26].

Analogy: Objects and relations between objects are matched but few or no object attributes are matched. This has been called *analogy* by Gentner [11].

Semantic Similarity: Objects or their attributes are not exactly the same but similar. All relations are matched. *Semantic similarity* described by Holyoak and Thagard [13, 26] includes this kind of similarity. Holyoak and Thagard include similar relations in *Semantic similarity* as well as similar objects and attributes. Semantic similarity will be measured using domain knowledge stored in the DKB (see Section 2.2).

Embedded Similarity: M_{cd} is similar to only part of C_{cd}. C_{cd} contains extra objects with or without their attributes and extra relations which do not match. This extra information can be regarded as not critical in determining the uniqueness of the particular circumstances to which a model should apply. Therefore, this extra information will be ignored in the similarity test.

An example is shown below.

C3:
C_c: house
C_{cd}: window(a), type(a,single_glazed),
wall(b), type(b,exposed), in(a,b),
bedroom(r), in(a,r),
floor_area(r,6,square_metre)
C_{sd}: height(a,1,metre), width(a,0.6,metre)

In C_{cd} of C3, the floor area of the bedroom is specified instead of the percentage area of the window. The floor area is not one of the key features of M_{cd} of M15 but is only needed to calculate the percentage area in the evaluation of the case. Therefore M_{cd} of M15 shown in Figure 2 can be said to be similar to C_{cd} of C3 and M15 will be selected.

Implied Similarity: Some objects are matched and some objects are not matched. But relations between objects imply that the situation is similar.

For example, no M_{cd} of any model in H_{R_4} matches C_{cd} of the case C4 by the criteria described so far.

C4:
C_c: house
C_{cd}: apartment(a), roof(b),
window(c), type(c,single_glazed),
have(a,c), in(c,b)
C_{sd}: percentage_area(c,11)

The window "c" is in the roof "b" and this means that "c" is a rooflight. From this fact, M19 is relevant to this case and must be selected. This can be achieved by redescribing C_{cd} as follows and repeating the similarity test:

C4′:
C_{cd}: apartment(a), roof(b),
rooflight(c), type(c,single_glazed),
have(a,c), in(c,b)

Then M_{cd} of M19 shown in Figure 2 is similar to this description by the embedded similarity criterion and M19 can be selected.

Here is another example.

C5:
C_c: house
C_{cd}: window(a), type(a,single_glazed),
bedroom(b), in(a,b)
C_{sd}: percentage_area(a,14.5)

From the fact that the window "a" is in the bedroom "b", it can be inferred that there is a wall "x" in the bedroom "b" and the window "a" is in the wall "x". Therefore, the circumstances can be redescribed as:

C5′:
C_{cd}: window(a), type(a,single_glazed),
bedroom(b), in(a,b),
wall(x), in(a,x)

Then M_{cd} of M15 shown in Figure 2 is similar to this description by analogy and embedded similarity criteria and M15 will be selected.

Alternative descriptions of the given circumstances can be produced by referring to domain knowledge. To be able to select a model which is similar by the implied similarity criterion, the system must have a sufficient amount of knowledge. This knowledge must also be reliable to get correct results. The more domain knowledge the system has, the more cases will be covered by the retrieval procedure.

3.3 Evaluation of a Case against Retrieved Models

To determine the conformance of the given case, the system will compare the proposed solution C_{sd} of the given case C with the solution M_{sd} of each retrieved model M. If the solution described C_{sd} is within the range of the solution M_{sd}, C is said to satisfy M. When a criterion is evaluated for requirements in an issue, all requirements described in the same criterion for other issues must be checked to ensure that there is no conflict.

For a case to be determined as compliant with the regulations, among models retrieved as relevant, all "strong" models should be satisfied but "weak" models need not be satisfied. When the proposed solution does not satisfy any "strong" models retrieved as relevant but satisfies "weak" model(s), those "weak" model(s) can be used as materials for justifying the relaxation in the case. The system will provide a list of models retrieved and the result of the evaluation of each model against the case.

4 Acquisition of New Cases

A new model will be created or existing models will be updated by acquiring new rules from cases, which are evaluated as not compliant but determined as compliant by the human expert. A case must pass the validity test for acquisition before being integrated into existing models.

4.1 Validity Test for Acquisition

The decision on whether the case is valid for acquisition depends mainly on the human expert. The system will assist the human expert by providing information needed in making such a decision.

For a case to be valid for acquisition, there are two conditions to be satisfied. First, the case needs to be *decided compliant*. Models in the path from the root model M_1 to the model M_n retrieved for evaluation provide the reason why the retrieved model should be satisfied. If the case does not violate any of rules represented as models in the path from

M_1 to M_{n-1}, [3] then the case must be regarded as compliant and valid for acquisition. If a case is valid for acquisition, the case C will be integrated with other "weak" models at the level of M_i, which is the model at the lowest level in the path among the models which provide reason for justification. Once such cases are integrated into the existing hierarchy of models, it is also possible to provide the expert with this model as backing materials for justification in similar cases in which the proposed solution does not satisfy any "strong" models (whether relevant or not) but the proposed solution satisfies this acquired "weak" model.

Second, there should be *no conflict* with existing models marked "strong" in model hierarchies other than H_G. To identify conflicts, the system will look for models in other model hierarchies whose M_{sd} is provided in terms of the same attributes or relationships between the objects matched with those in C_{sd} within the same category with C_c. If there exist such models, C_{sd} will be evaluated against M_{sd} of those models. It is assumed that there is no conflict between the models already in the MKB.

For example, when the following case is given, only M15 will be retrieved by the analogy criterion but the solution proposed in this case does not satisfy M15.

C6:
- C_c: house
- C_{cd}: window(a), type(a,double_glazed),
  wall(b), type(b,exposed), in(a,b)
- C_{sd}: percentage_area(a,18)

The only difference is the type of the window "a", which is "double-glazed". There are three models, M3, M5 and M6, that provide the reasons why M15 is to be satisfied. In order for C6 to be added to the MKB, first, the proposed percentage of double-glazed window, "18", must be proved by the expert to be reasonable provision (M6) to limit heat loss (M5) from the house. Second, there must be no conflict with existing strong models. In C6, the solution is described in terms of the "percentage area" of the window. Suppose that there are two strong models which describe M_{sd} in the same category with C_c of C6 and also describe their required solution in the same terms with C_{sd} of C6: M30 in the model hierarchy of "convenience" and M43 in the model hierarchy of "health".

M30:
- M_c: dwelling
- M_{cd}: apartment(A),
  wall(B), type(external),
  window(C), have(A,C), in(C,B)
- M_{sd}: percentage_area(C,P), P $\geq$ 6.67
- M_s: strong

[3]This will be decided by the human expert. The decision process is often political rather than logical, and decisions are often made on the ground of political policy. It could be possible to build a computational model of such a decision process but that is beyond the bounds of this paper.

M43:
M_c: dwelling
M_{cd}: room(A), not kitchen(A),
floor(C), in(C,A),
area(C,N,square_metre), N $\geq$ 4,
ventilator(B), in(B,A)
M_{sd}: percentage_area(B,P), P $\geq$ 3.33
M_s: strong

C6 does not conflict with either of M30 and M43, and therefore C6 is valid for acquisition.

However, solutions may be too specific to apply to new cases so that they need to be transformed into less specific solutions. This will be done by integrating with other "weak" models.

4.2 Integration of Cases with Existing Models

Acquisition of a case is done by integrating the case into existing "weak" models. If there is no "weak" model which represents rules in the same context, a "weak" model will be created from the case. If there are "weak" models which represent rules in the same context, the case will be integrated into those models. The category and the proposed solution of the case will be combined with the category and the solution of those models.

First, the system will guess the category of buildings to which rules in the case should apply. The system will guess the category by looking at "strong" sibling models at the same level as the model which the case is supposed to satisfy but does not. The system will look for "strong" models:

- whose category subsumes the category of the case;
- whose circumstance description is similar to the circumstance description of the case; and
- whose solution is given in the same criteria.

If there are such models, then the category of selected models will be the category to which rules in the case may apply.

Second, the system will collect "weak" models with which the case can be integrated. Among "weak" sibling models at the same level as the model which the case is supposed to satisfy but does not, the system will look for models:

- whose category is subsumed by the guessed category;
- whose circumstance description is similar to the circumstance description of the case; and
- whose solution is given in the same criteria.

Finally, the system will generalise the solutions from the case and the collected "weak" models, and create a new model from the guessed category, the circumstance description of the case and the generalised solution. The strength will be set to "weak".

The "weak" models whose solutions are integrated into the new model will be replaced with this new model.

For example, in Section 4.1, C6 was decided to be valid for acquisition. M15 was the model that C6 was supposed to satisfy. Among "strong" sibling models of M15, M15 is the only "strong" model which represents a rule in the same context as C6. Therefore, the category of M15 ("dwelling", "industrial building", "storage" and "warehouse") will be the category of the new model.

Then the system will look for "weak" models which represent rules in the same context as the case. Suppose that there is only one "weak" model M21 which meets those conditions.

M21:
M_c: flat
M_{cd}: window(A), type(A,double_glazed),
wall(B), type(B,exposed), in(A,B)
M_{sd}: percentage_area(A,C), C $\leq$ 17
M_s: weak

The solution in M21 is given in terms of the *maximum* percentage area of windows. "18" percent of window area, which is greater than the maximum percentage are of windows prescribed in M21, is permitted in C6.

M21:
C_{sd}: percentage_area(A,C), C $\leq$ 17
C6:
C_{sd}: percentage_area(a,18)

Therefore the solutions from M21 and C6 can be generalised as "the percentage area of double-glazed windows in walls should not exceed 18 percent of the total floor area". M21 will be replaced with the new model M21′.

M21′:
M_c: dwelling, industrial_building,
storage, warehouse
M_{cd}: window(A), type(A,double_glazed),
wall(A), type(B,exposed), in(A,B)
M_{sd}: percentage_area(A,C), C $\leq$ 18
M_s: weak

5 Concluding Remarks

In this paper, we proposed a **K**nowledge-**I**ntensive **C**ase-based reasoning **S**ystem (KICS) which can be used for consultation and maintenance of building regulations information. There are several features worth pointing out as likely contributions.

First, both statute and case history are integrated in one knowledge base using a single representation scheme. In previous approaches to knowledge representation in the domain of law, two different knowledge representation schemes have been used [18, 30, 21, 22, 25, 28, 19, 1]. In our approach, legal rules are regarded as the results of the accumulation and generalisation of knowledge from case histories, and rules from both statute and case histories are represented as models.

Second, intentions of building regulations can be expressed explicitly by representing building regulations information as abstraction hierarchies of models. In this way, the semantics of building regulations information can be represented more clearly than previous approaches such as the SASE model [7, 8, 12], a rule-based approach [24], and the information model-based approach [4, 5].

Third, retrieval of relevant models is carried out in the abstraction hierarchies of models. These abstraction hierarchies make it possible to retrieve relevant models in a very efficient way by pruning irrelevant models at an early stage in retrieval (retrieval by category). In addition to the notion of structural mapping (literal similarity, analogy, semantic similarity, embedded similarity) which has been widely used for similarity assessment [11, 10, 13, 26, 2], similarity is also assessed by using domain theory (implied similarity).

Finally, as regulations information develops in time, the system maintains an up-to-date interpretation of regulations information by integrating cases into existing models in the MKB. Integration of cases into the MKB is guided by knowledge stored in the MKB, i.e., knowledge from statutory regulations and previous cases.

We are just beginning to implement the system in Prolog. Along with implementation, there are two issues that need further investigation. First, inheritance mechanisms for category information in the abstraction hierarchies need to be implemented in the proposed knowledge representation scheme. Second, the system needs to provide a means of acquiring new domain knowledge, which includes checking correctness and consistency of domain knowledge.

Acknowledgements

This research has been done as part of the project sponsored by the UK Science and Engineering Research Council under Grant No. GR/F/89022. Thanks to all members of the project for their advice and discussion.

References

[1] Kevin D. Ashley. Reasoning with cases and hypotheticals in HYPO. *International journal of man-machine studies*, 34:753–796, 1991.

[2] L. Karl Branting. Building Explanations from Rules and Structureed Cases. *International journal of man-machine studies*, 34:797–837, 1991.

[3] Steven M. Cornick and Debbie A. Leishman. Integrating Building Codes into Design Systems. In *Intenational Workshop on Computers and Building Standards in Espoo, Finland*, May 1991.

[4] Marcel de Waard. Computer Aided Conformance Checking. In *Computers and Building Standards Workshop in Montreal, Canada*, May 1992.

[5] Marcel de Waard. *Computer Aided Conformance Checking.* PhD thesis, Delft University of Technology, The Netherlands, 1992.

[6] Ronald M. Dworkin. *Law's Empire.* Fontana, 1986.

[7] Steven J. Fenves and James H. Garrett. Knowledge Based Standards Processing. *Artificial Intelligence*, 1(1):3–14, 1986.

[8] Steven J. Fenves, Richard N. Wright, Fred I. Stahl, and Kent A. Reed. Introduction to SASE: Standards Analysis, Synthesis, and Expression. Technical Report 87-3513, National Bureau of Standards, 1987. NBSIR.

[9] R. Ganeshan, S. Finger, and J. H. Garrett. Representing and Recording Design Intent: A Progress Report. Technical report, Department of Civil Engineering, Carnegie Mellon University, 1992.

[10] D. Gentner and R. Landers. The Mechanisms of Analogical Learning. In Stella Vosniadou and Andrew Ortony, editors, *Similarity and Analogical Reasoning*, pages 199–241. Cambridge University Press, 1989.

[11] Debre Gentner. Structure Mapping: A Theoretical Framework for Analogy. *Cognitive Science*, 7:155–170, 1983.

[12] M. Maher Hakim and James H. Garrett. Issues in Modelling and Processing Design Standards. In *Computers and Building Standards Workshop in Montreal, Canada*, May 1992.

[13] Keith J. Holyoak and Paul Thagard. Analogical Mapping by Constraint Satisfaction. *Cognitive Science*, 13:295–355, 1989.

[14] Bryan Lawson. *How Designers Think.* Butterworth Architecture, second edition, 1990.

[15] Brian Logan and Tim Smithers. The Role of Prototypes in Creative Design. Technical Report 453, Department of Artificial Intelligence, University of Edinburgh, 1989. DAI Research Paper.

[16] Neil McCormick. *Legal Reasoning and Legal Theory.* Clarendon Press, 1978.

[17] Neil McCormick and Zenon Bankowski. Some Principles of Statutory Interpretation. In Jan van Dunne, editor, *Legal Reasoning and Statutory Interpretation*, pages 41–53. Gouda Quint BV Arnhem, 1989.

[18] Edwina L. Rissland. Artificial Intelligence and Law. *The Yale Law Journal*, 99:1957–1981, 1990.

[19] Edwina L. Rissland and David B. Skalak. CABARET: Rule Interpretation in a Hybrid Architecture. *International journal of man-machine studies*, 34:839–887, 1991.

[20] Scottish Office. *Technical Standards for Compliance with the Building Standards (Scotland) Regulations 1990*, 1990.

[21] Marek Sergot. Representing Legislation As Logic Programs. Technical report, Department of Computing, Imperial College of Science and Technology, 1985.

[22] Marek Sergot, F. Sadri, R.A. Kowalski, F. Kriwaczek, P. Hammond, and H.T. Cory. The British Nationality Act as a Logic Program. *Communications of the ACM*, 29(5), May 1986.

[23] Herbert A. Simon. *The Sciences of the Artificial.* MIT Press, second edition, 1981.

[24] David Stone. Intelligent Information Systems for Building Standards. In *Proceedings of EuropIA 88, Paris*, 1988.

[25] Richard Susskind. *Expert Systems in Law.* Oxford University Press, 1987.

[26] Paul Thagard and Keith J. Holyoak. Why Indexing is the Wrong Way to Think about Analog Retrieval. In Kristian Hammond, editor, *Proceedings of DARPA Case-Based Reasoning Workshop*, pages 36–40. Morgan Kaufmann, 1989.

[27] Chris Tweed. Information Systems Study of Relaxation Procedures in Branch 3 of the Building Directorate of the Scottish Office. Technical report, EdCAAD, University of Edinburgh, February 1990.

[28] Anne von der Leith Gardner. *An Artificial Intelligence Approach to Legal Reasoning.* MIT Press, 1987.

[29] R. F. Walker, A. Oskamp, J. A. Schrickx, G. J. Van Opdorp, and P. H. Van Den Berg. PROLEXS: Creating Law and Order in a Heterogeneous Domain. *International journal of man-machine studies*, 35:35–67, 1991.

[30] D. A. Waterman, J. Paul, and M. A. Peterson. Expert Systems for Legal Decision Making. *Expert Systems*, 3(4), 1986.