Use of case-based reasoning in the domain of building regulations

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: In traditional legal decision support systems, it has been regarded as natural to represent statutes in terms of decision rules and to link these to a separate case-based reasoning system for handling precedent. Statutory legal rules used in these systems are formal and prescriptive. Building regulations in Scotland are part of statute law and constitute part of a legal system together with case histories. In recent years, the regulations have been becoming less prescriptive and more emphasis has been put onto the interpretive use of the regulations. In developing a system which can assist domain experts in interpreting the regulations, this trend has presented us with difficulties in employing this traditional approach and has led us to a unified case-based model of the regulations and case histories. In this paper, we first describe the characteristics of the regulations and the activities involved in this domain. Second, we explain the reason why we abandoned the traditional approach. Third, we describe the system which is under development using this case-based model.

1 Introduction

Legal and regulatory reasoning is one of the AI application domains where substantial amounts of research have been done over the last twenty years [5]. The law contains “legal rules”, which state what should be done in certain circumstances in the context of a legal system. Most of these are expressed in the form of statutes and cases. Building regulations in Scotland are part of statute law and constitute part of a legal system together with case histories.

Since it is normal to view statutory regulations as “legal rules”, it is also natural to expect these to be most easily expressed formally in terms of decision rules (perhaps in some propositional language). However, it is becoming increasingly common to make the regulations less prescriptive, with the text of the regulations merely providing a general framework within which the interpretation of the regulations is established through precedent. In these types of system the legal rules do not in themselves allow any decision to be made. Much greater importance is attached to the means by which precedent is accessed and interpreted. A good example of this approach is found in the building regulations of the Scottish Office. These will be used throughout this paper as a concrete example of the more general application.

First of all, we describe the characteristics of building regulations and the activities involved in this domain. Second, we explain the reason why we decided to abandon the more traditional rule-based approach even for statutory building regulations. Then we describe the system itself [16, 15], and conclude with some remarks about the system.

2 The domain of building regulations

The Building Directorate of The Scottish Office is responsible for the drafting and maintenance of building regulations— the Building Standards (Scotland) Regulations and their associated Technical Standards [7]— which set out the statutory technical requirements affecting building design and construction in Scotland. The Building Directorate is also responsible for the administration of the appeals procedures under the building control system in Scotland.

Building regulations have the generic objectives of safeguarding public health and safety and of energy conservation in the built environment. These objectives are the government policy and the principles which bring building regulations into existence. The Building (Scotland) Act 1959 enables the Secretary of State for Scotland to make the Building Standards (Scotland) Regulations and their associated Technical Standards, in which these issues or principles are translated into more detailed technical requirements.

The Building Standards (Scotland) Regulations contain 24 regulations which state requirements for the sixteen subject areas to achieve the generic objectives mentioned above. These requirements are very general and expressions are highly abstract. For example, Regulation 22 states that:

> 22. In a building to which this regulation applies, reasonable provision shall be made for the conservation of fuel and power.

The bounds of applicability of this requirement are, intentionally, left unclear at this point. In addition to the 24 main regulations, there are regulations and schedules detailing exemptions, classifications of building, rules of measurements and so forth. One of these regulations provides ways of narrowing down the applicability of requirements such as Regulation 22. According to this regulation, these requirements can be met by compliance with the relevant standards and provisions set out in the Technical Standards document. Requirements stated in these relevant standards and provisions are a refinement of requirements in the Building Standards (Scotland) Regulations, and can be regarded as the description of how requirements in Building Standards (Scotland) Regulations should be interpreted. For example, the Standard J2.2 describes how the building fabric of buildings can meet Regulation 22.

> J2.2* A building must have provision for conservation of fuel and power in accordance with J2.3 or J2.4 or J2.5.

The asterisk attached to J2.2 means that there are provisions deemed to satisfy the standard J2.2. Therefore there are at least four ways of complying with Regulation 22. For example, one way is described in J2.3 (see Figure 1) in terms of U-values of building fabric and the areas of openings in walls and roofs.

Requirements are specified only for certain aspects or features of buildings that are related to generic objectives. Other aspects or features unrelated to those generic objectives are disregarded. In the examples above, requirements to achieve energy conservation specify only U-values of floors, walls and roofs and the total areas of openings in walls and roofs. Other

J2.3 Under this method every part of the specified floors, walls and roofs (excluding any opening in them) must be constructed so that they do not exceed the appropriate U-values, and the total areas of single glazed openings in those walls and roofs do not exceed the areas, prescribed in the table to this standard.

Table to J2.3: U-values of elements and areas of glazed openings							
Purpose group	Building type	U-values of elements (W/m^2K)				Maximum area of single glazed openings in walls and roofs assuming a U-value of 5.7	
		Floors next to the ground	Exposed floors and walls	Semi-exposed floors and walls	Roofs	Windows	Roof-lights
1	Dwelling	0.45	0.45	0.6	0.25	Windows and rooflights together: 15 % of total floor area	
2	Other residential	0.45	0.45	0.6	0.45	25 %	20 %
3,4 or 5	Places of assembly offices, shops	0.45	0.45	0.6	0.45	35 %	20 %
6 or 7	Industrial storage	0.45	0.45	0.6	0.45	15 %	20 %

Figure 1: The Standard J2.3

factors, for example the material of building fabric or the shape and position of openings in walls and roofs, are regarded as irrelevant in the subject of energy conservation. Cases are also described in the same manner. They do not provide a design of whole building but provide only the description of aspects or features of building related to a particular subject area in question.

The relevancy between requirements, which plays a central role in interpretation of the document, is sometimes clearly expressed in the document but in some cases may rely on experts' experience in applying requirements to particular problems.

The interpretation of written requirements changes through time. At the time of legislation, it is impossible to anticipate all possible circumstances in which a case may arise after legislation. New legal rules are set out in unexpected circumstances or slightly different circumstances and existing legal rules may be relaxed in exceptional circumstances. Therefore new ways of interpretating of requirements such as Regulation 22 should be introduced when a case arises in circumstances not covered by standards and provisions or not explicitly stated in it.

Furthermore, if it is unreasonable for specific requirements to be applied and these requirements can be met in a less onerous way, existing requirements may be relaxed. In this sense, cases may set out new legal rules but legal rules set out in case histories may or may not apply to new cases. As legal rules from case histories accumulate, they are reviewed when revising statutory regulations. The rulings made in case histories may be generalised and included in the new version of regulations

An application for relaxation of particular requirements is first submitted to the Local

Building Control Authority and if there is disagreement about interpretation between the applicant and the Local Building Control Authority the appeals procedure follows. The argument surrounding each appeal decision is often complex, involving the balancing of a variety of assertions by the parties concerned. Experts make a decision after (if necessary) consulting all information relevant to the given case. In addition to statute and case histories relevant to the given case, experts should refer to all information available to provide justification for their decision within the legal system in which they work. It includes government policies, principles taken for granted in the given field of law, and other related law. It would be ideal to build a knowledge base which includes all these information but that would be too ambitious. The problem is therefore to support the consultation of these loosely expressed legal requirements without being forced to make explicit all the information needed to arrive at a decision.

We do not employ a rule-based approach in our system and employ a purely case-based approach for both statute and precedent. In the next section we explain the reasons why we totally rejected a rule-based approach.

3 Why not a traditional legal decision support system in this domain?

Legal rules are the basic unit of knowledge to be represented in the knowledge base. Legal rules are requirements, which describe a required design *solution* in certain *circumstances*. Given this description of legal rules, conventional if-then rules seem a natural choice for representing legal rules—for example, in the form:

> if *circumstances* then *solution*

For example, the Standard J2.3 can be described in this form as shown below. For ease of understanding, *circumstances* and *requirements* are given not in computer-executable form but in approximate English.

> if *the type of building is dwelling and there are single-glazed windows in the wall*
> then *the total area of windows must not exceed 15 % of floor area*
> .....

Then some form of rule interpreter might be used to derive information from legal rules represented in the knowledge base. This approach has been taken in Waterman and Paterson's LDS [14], Sergot et al's representation of the British Nationality Act 1981 [8, 9] and Susskind's system [11]. The Scottish Office's Building Directorate has also experimented with this approach in their in-house prototype systems [10]. Legal rules used in these systems are formal and prescriptive. However, the regulations are becoming less prescriptive and more emphasis is put onto the interpretive use of the regulations. This presents us with several difficulties in employing a traditional decision support system approach.

1. Legal rules are described at different levels of detail.

In addition to legal rules which are prescriptive and state requirements at a very detailed level, such as the Standard J2.3, there are legal rules which are less prescriptive and which explain why these detailed requirements should be satisfied in the context of the legal system. Regulation 22 and the Standard J2.2 are examples of these legal rules.

Legal rules are independently described at various levels of detail. Legal rules at the upper level describe requirements using more abstract terms which have broader meaning. Terms like "reasonable provision" and "adequate means" are used at this level. Then one or more legal rules are introduced to provide various ways of interpreting these abstract legal rules. Refined legal rules are described using more detailed and concrete terms. For example, what "reasonable provision" means in the context of achieving "energy conservation" is defined in terms of "heat loss", and then the "building fabric" and the "building services", and then "walls", "floor", and so forth. In other words, legal rules at lower levels are a refinement or specialisation or decomposition of legal rules at upper levels.

2. There exists a terminological gap between legal rules and cases.

Although appropriate rule structuring might overcome the problem of different levels of detail in statute, the language used in statutory regulations is generally at a more general level of description than that used in describing the objects (buildings) to which the rules apply. The description of the circumstances and required solutions in statutory regulations is the description of requirements generalised from many particular circumstances and solutions in these particular circumstances. Cases arise in one of those particular circumstances and propose one particular solution. For example, in cases, the type of building may be given as "flat" or "house", which are of type "dwelling", or instead of "rooflight" the term "window" may be used. It would be possible to bridge these terminological gaps by introducing domain-specific "bridging" rules, but it is not always possible reliably to bridge this terminological gap in this way, as we demonstrated below.

3. The meaning of terms is often in debate.

It is sometimes a matter of debate as to what the regulations mean by basic concepts, such as "floor area". For example, does "floor area" include or exclude raised platforms? or is embankment "a place of safety"? This is one of the most common reasons for appealing against the regulations and seeking relaxation, and there are prolonged arguments as to the meaning of terms in the context of the given circumstances in many appeal cases provided by the Scottish Office's Building Directorate. Although it would be possible to encode in rules how these terms are interpreted in particular circumstances, there would remain the serious practical problem of maintaining consistency across the vast collection of these "bridging" rules, given that interpretation of the regulations constantly changes over time and there is always the possibility of conflicts between these bridging rules.

4. New legal rules are constantly emerging.

Even if the above theoretical and practical objectives could be addressed, we would still be left with the problem of accommodating new, unexpected or exceptional circumstances within the decision support system, for example if someone invented a new type of building material.

5. Legal rules are the generalisation of the rulings made in case histories.

One possible way of solving these problems (1-4) is to use the rule-based model for statutory regulations and to use the case-based model for case histories such as in CABARET [6], GREBE [1] and PROPLEXS [13]. These systems assume that legal rules in statutes are open-textured and may (in a limited extent) conflict each other but they are formal and fairly stable. The problems with open-texturedness and conflicting rules are compensated by referring to cases.

But we take a different view of statutes and cases. Legal rules in the statutes are the result of accumulation and generalisation of the rulings made in case histories. A case must comply with legal rules in the statutes which are applicable. In this sense, legal rules in the statutes act as "strong" constraints on new cases. Cases provide a way of interpreting a particular set of legal rules in the statutes under particular circumstances, in other words more detailed legal rules than those in the statutes. There may be similar cases and it is possible to derive more generalised legal rules from those similar cases. The interpretation of legal rules in the statutes changes over time, legal rules set out in case histories may or may not apply to new cases. In this sense, legal rules set out in case histories act as "weak" constraints to new cases.

Once we view legal rules in this way (i.e. legal rules are generalised cases) legal rules from both statutes and cases can be represented using only a case-based model. In the system described in the next section, cases are kept in a separate case library, and legal rules, i.e. the generalised cases, are represented as *models*. The case-based approach has the following advantages over the rule-based approach in this domain:

- It allows more flexibility in organising models than in a traditional rule-based approach.
- Terminological gaps between legal rules and between legal rules and cases can be bridged by using matching algorithms.
- It does not insist on consistency among models and cases.
- New legal rules can be introduced into the system more easily.

A system which can assist domain experts in the appeal process is under development using this approach. The system does not make any decision about relaxation but is intended to provide the experts with relevant information, i.e. relevant legal rules and cases, so that they can arrive at a decision.

4 Overview of the system

The overview of the system is shown in Figure 2. There are three knowledge bases: the Model Knowledge Base, the Domain Knowledge Base and the Case Library. Legal rules are repre-

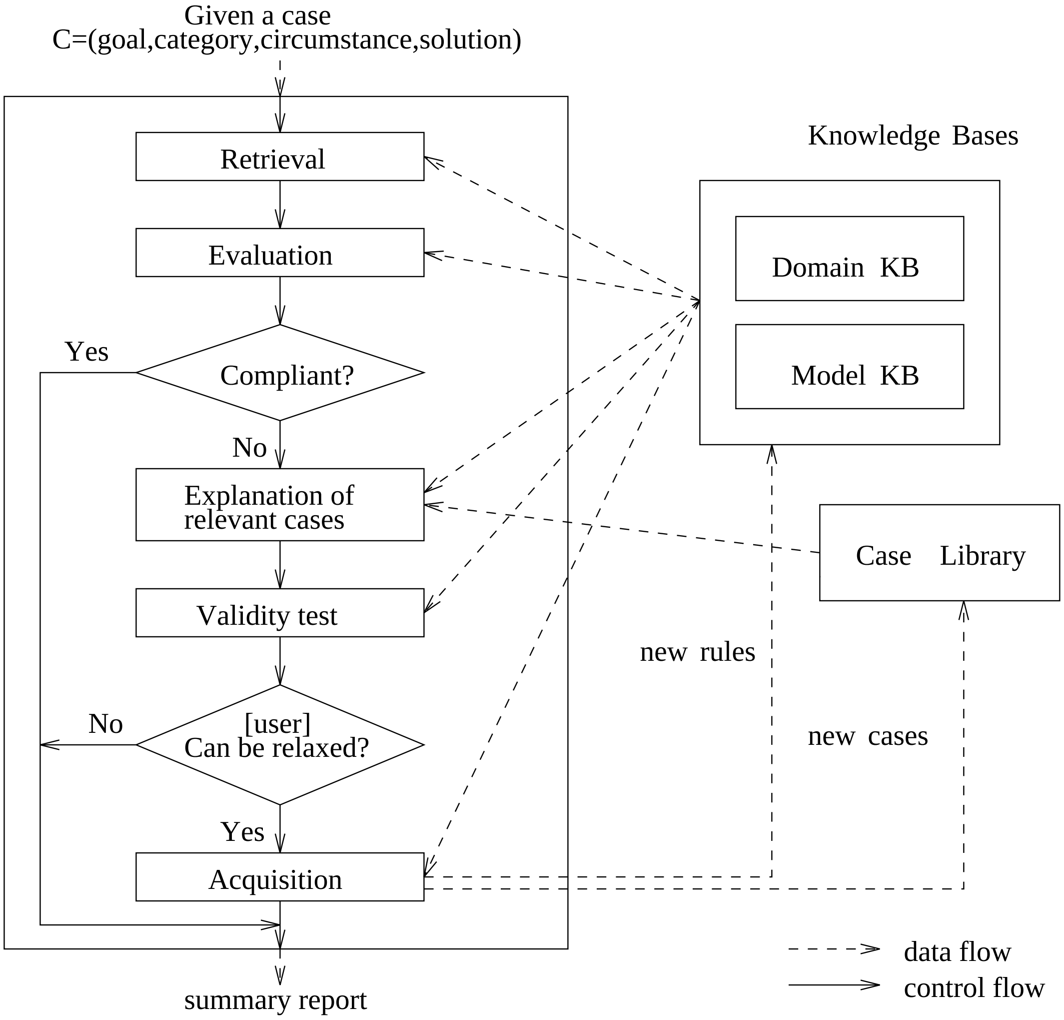

Figure 2: System overview

sented in the Model Knowledge Base. In addition to legal rules, background domain knowledge must be used in the retrieval and maintenance of legal rules, and cases and argumentation surrounding each case must be provided to the user. Argumentation surrounding each case is represented in the form of an IBIS-like argumentation network [4]. Background domain knowledge is represented in the Domain Knowledge Base, and cases and argumentation surrounding each case are represented in the Case Library.

A case will be given as $C = (C_{id}, C_g, C_c, C_{cd}, C_{sd})$ where C_{id} is the id number of the case, C_g is the goal given in the case, C_c is the category of the building, C_{cd} is the circumstances description, and C_{sd} is the proposed solution. Given a case C, the system first retrieves relevant models by comparing goal, category and circumstance parts of the given case with models in the MKB. Once relevant models are found, the solution part of the case will be evaluated against the required solution of the retrieved model. The case is said to be *compliant* if the proposed solution satisfies the required solution of the retrieved strong models.

If the case is not compliant, there is a possibility of relaxation and the system will explain

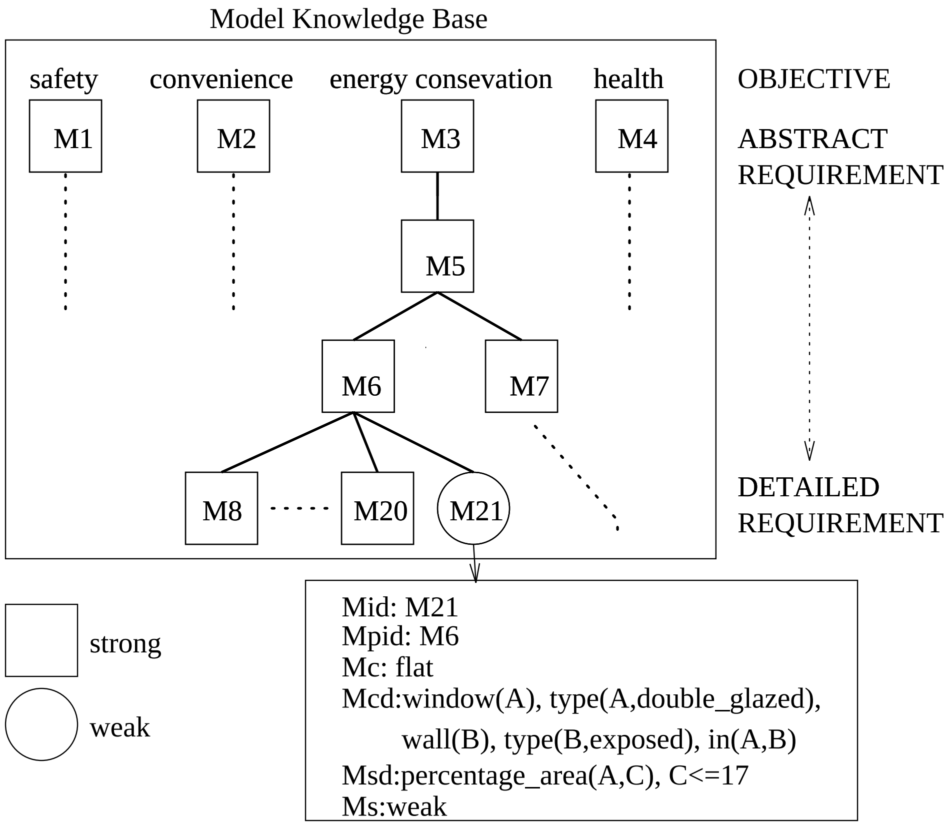

Figure 3: Model knowledge base

cases associated with retrieved models. An argumentation structure is used to explain key points in argumentation surrounding the relevant cases. Then the system checks the validity of the solution given in the case by checking whether there is conflict between the case and existing models. If the expert decides to grant relaxation, new legal rules set out by the case will be integrated into the existing abstraction hierarchy by combining new rules with relevant weak models. The case itself will also be stored in the Case Library.

4.1 The Model Knowledge Base

Legal rules are represented as *models* and then an abstraction hierarchy is built for each generic objective. In each abstraction hierarchy, a model M is represented as a set of six parts: $M = (M_{id}, M_{pid}, M_c, M_{cd}, M_{sd}, M_s)$ where M_{id} is the id number of the model, M_{pid} is the id number of the parent model in the abstraction hierarchy, M_c is the category of buildings 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 solutions, and M_s is the strength, i.e. "strong" or "weak". The strength will be given as "strong" if the model is taken from statutory regulations, or as "weak" otherwise. M_{cd} and M_{sd} are described in terms of objects, their attributes and relations between objects. An example of a model is shown in Figure 3.

Abstraction hierarchies of models are also shown in Figure 3. The generic objectives are represented in M1-M4: all buildings should be built to ensure the safety (M1), convenience (M2) of people in and around buildings and the conservation of fuel and power (M3) and not to harm the health of people living in or visiting buildings (M4). The experience of working in this domain provides an interpretation of M3 as all buildings should be built to limit heat loss (M5) to achieve the conservation of fuel and power. M5 again is interpreted as: reasonable provision should be made for the building fabric (M6) and for the building services (M7). M8-M20 represent requirements which can be found in the Standard J2.3. The Standard J2.3 prescribes the maximum total area of single-glazed window. The maximum required total area of other types of window are set out in case histories. M21 represents a legal rule generalised from such cases: when the type of building is flat and the window is double-glazed, the total area of window must not exceed, for example, 17 % of total floor area. M21 can be regarded as another interpretation of M6.

Each abstraction hierarchy provides a framework for the interpretation of legal rules under a particular legislative intention. Although legal rules set out in a case are integrated into models in abstraction hierarchies, the actual case with all argumentation surrounding the case is stored in the Case Library. Therefore these abstraction hierarchies serve as indexes to cases stored in the case library as well as the interpretation hierarchies of information about regulations.

4.2 The Case Library and argument structure

The Case Library is a collection of cases processed by the system. In addition to an argumentation network, a case C is represented as a set of eight parts: $C = (C_{id}, C_g, C_c, C_{cd}, C_{sd}, C_d, C_m, C_t)$ where C_{id} is the id number of the case, C_g is the goal given, C_c is the category of the building, C_{cd} is the circumstance description, C_{sd} is the proposed solution, C_d is the decision taken by the Building Directorate (BD), i.e. "granted" or "refused", C_m is the id number of the model which is relevant to the case, and any other information will be stored in texts in C_t. For example, Case 90 is represented as follows.

C_{id}: C90
C_g: safety
C_c: house
C_{cd}: stair(a), exit_stairway(b), is(b,enclosed), part_of(a,b)
C_{sd}: material(a,timber), fire_resistance(a,1,hour)
C_d: granted
C_m: M23
C_t: nil

Argumentation for the relaxation of specific regulations starts with usually two *positions*, one suggested by the applicant and the another one or two suggested by the Local Building Control Authority (the LBCA). In addition to these two *players*, there may be other players who make *arguments* to *support* or *oppose* the suggested positions such as BD itself or the firemaster. The role of BD is to arbitrate between these two main players—the applicant and the LBCA—and decide which position is reasonably acceptable in the context of the given

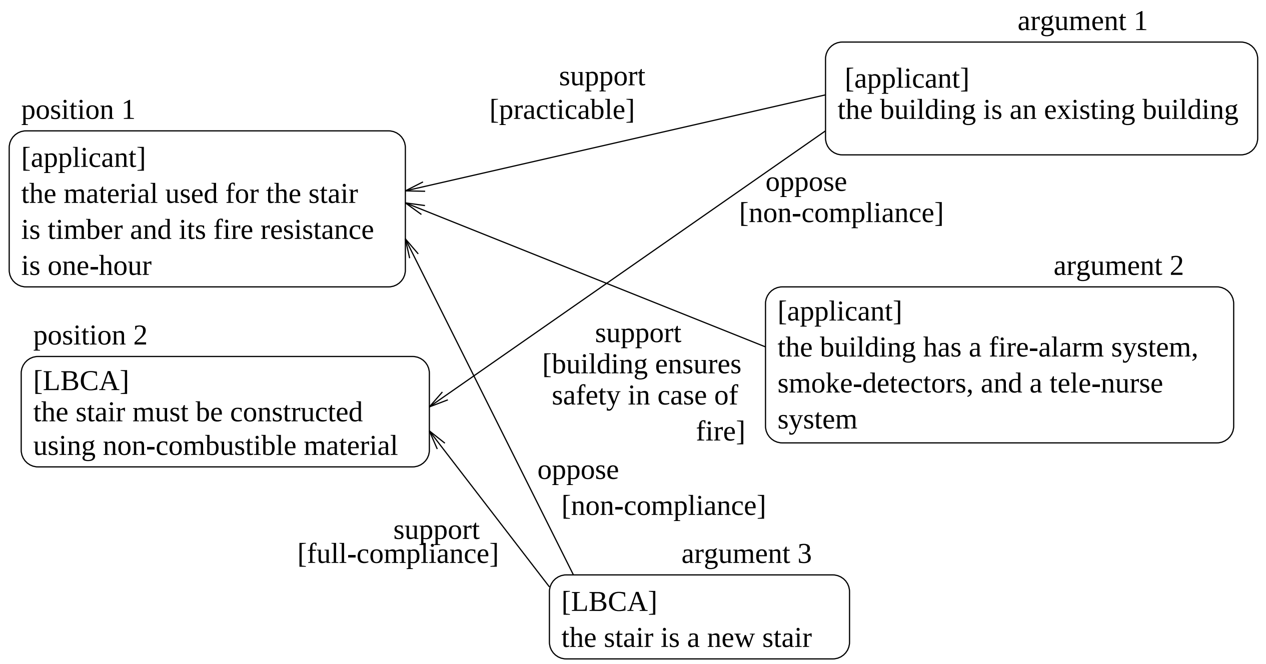

Figure 4: Argumentation surrounding Case 90

case. Such argumentation is recorded in the Case Library along with cases in the form of the following IBIS-like argumentation structure [4].

$$position(C_{id},P_{id},Pl,Position)$$
$$argument(C_{id},A_{id},Pl,Argument)$$
$$support(C_{id},A_{id},P_{id},Reasons)$$
$$oppose(C_{id},A_{id},P_{id},Reasons)$$

where C_{id} is the id number of the case in which position with the id number P_{id} and argument with id number A_{id} are given, Pl is the name of the player, and $Reasons$ are reasons for providing arguments for supporting or opposing a particular position. An example of an argumentation structure is shown in Figure 4. Reasons to *support* or *oppose* a position are enclosed in brackets ([...]) in this Figure.

4.3 The Domain Knowledge Base

Knowledge in the Domain Knowledge Base (the DKB) is used in classifying a given case, i.e. to find models whose M_c and M_{cd} are most similar to C_c and C_{cd} of the given case. Specialisation hierarchies are used to represent building categories and concepts used to describe models and cases. Aggregation hierarchies are used to represent structural knowledge about building components. Examples of specialisation/aggregation hierarchies are shown in Figure 5.

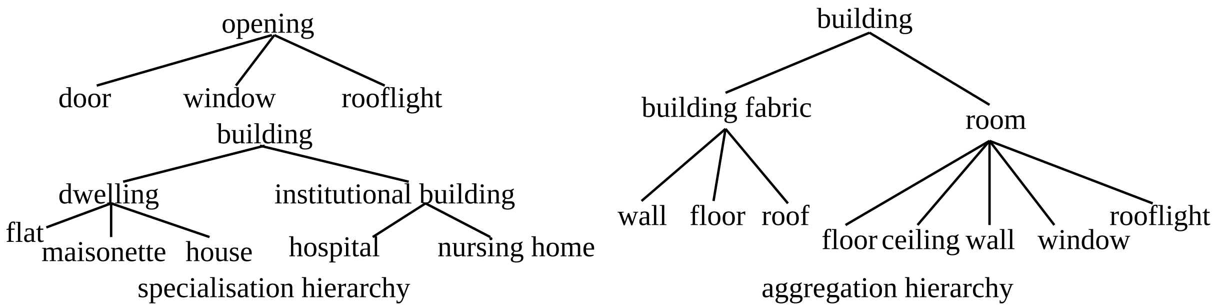

Figure 5: Examples of specialisation/aggregation hierarchy

4.4 Retrieval of relevant models

A case will be given as $C = (C_{id}, C_g, C_c, C_{cd}, C_{sd})$ where C_{id} is the id number of the case, C_g is the goal given, C_c is the category of the building, C_{cd} is the circumstance description, and C_{sd} is the proposed solution. Given a case C, the retrieval is done in three steps.

1. Retrieval by goal and category

The system selects an abstraction hierarchy of the given C_g. Then the system selects only models whose M_c *subsumes* the C_c of the given case using the specialisation hierarchy of building categories in the DKB.

2. Selection of models at the right level

Since models in the selected abstraction hierarchy describe legal rules using terms at different levels of detail, the next step is to find the *right level* in the abstraction hierarchy which has models comparable to the C_{cd} of the given case. This is done by selecting the level at which objects in M_{cd} are on the same level of aggregation hierarchies as objects in C_{cd}.

3. Selection of models by similarity of circumstances

Once models at the right level are found, the system assesses similarity by comparing the details of M_{cd} of selected models with C_{cd} of the given case. The system first applies the notion of structural mapping [2, 3, 12]. If this attempt fails, the system tries to bring out similarity hidden under the surface description of C_{cd} using knowledge stored in the DKB (*implied similarity*). The system transforms C_{cd} into an alternative description using knowledge in the DKB and repeats the similarity test using the notion of structural mapping. Heuristic rules used to transform C_{cd} are as follows.

1. Introduce objects which exist in M_{cd} but not in C_{cd} and add "part_of" relations between existing and new objects.
2. Add default properties if properties of objects in C_{cd} are missing. For example, if the occupancy capacity of a storage room is not given in C_{cd} then the default occupancy capacity,

0, is added to C_{cd}. Another example is, if there is a storey on the ground/first/second level, then the default height of the floor above the ground/first/second level is added to C_{cd}.

3. Infer relations between objects in C_{cd} and add them to C_{cd}. For example, if an object A is connected to an object B and B is connected to another object C, a new relation "A is connected to C" is added to C_{cd}. Another example is, if an object A is part of an object B and B is part of another object C, a new relation "A is part of C" is added to C_{cd}.

4.5 Evaluation

Once relevant models are found, the system compares the C_{sd} of the given case with the M_{sd} of retrieved models. If the solution described in C_{sd} is within the range of the solution described in M_{sd} of a model, then the case is said to satisfy the model. For a case to be determined to be *compliant* with the regulations, all "strong" models among retrieved models should be satisfied but "weak" models need not be satisfied.

4.6 Explanation and validity test

When the proposed solution C_{sd} does not satisfy any "strong" models retrieved as relevant, there is a possibility of relaxation. The decision of whether the relaxation is to be granted will be made by a human expert. The system only provides relevant information stored within the system: models in the upper levels of retrieved models, which provide the reason why the retrieved models should be satisfied, and cases attached to relevant models. Argumentation recorded using an IBIS-like argumentation structure can also be used to provide an explanation of the key points in the debate surrounding the case. For example, the system explains argumentation on the proposal for relaxation suggested by the applicant in Figure 4 as follows.

> The proposal [the material used for the stair is timber and fire-resistance is one-hour] suggested by the applicant was granted.
>
> The proposal was supported by the following arguments:
>
> Argument1 by the applicant: The proposal is a practicable solution in the circumstances [the building is an existing building]
>
> Argument2 by the applicant: Under the circumstances [the building has a fire-alarm system, smoke-detectors, and a tele-nurse system] the proposal has the following consequences [the building ensures safety in case of fire]
>
> The proposal was opposed by the following arguments:
>
> Argument3 by the LBCA: The proposal constitutes non-compliance based on the facts [the stair is a new stair]

If the human expert decides to grant relaxation, the system will check the validity of the new rules set out by this new case. This is done by checking whether there is conflict between the new rules and rules described in the same criterion in different abstraction hierarchies. This validity test is needed because the regulations deal with various issues and it is very rare for any part of a building to serve only one purpose. Different standards may be given which use the same criterion to resolve different issues. Therefore, when a criterion is evaluated for requirements in one issue, all requirements described using the same criterion with respect to other issues must be checked to ensure that there is no conflict.

4.7 Acquisition

If there is no conflict, then the new rules will be integrated into the abstraction hierarchies. First, the system will hypothesize the category of buildings to which the new rules should apply by looking at the M_c of "strong" sibling models which are similar to the case. Second, the system will collect "weak" sibling models which are relevant to the case and generalise the solutions from the C_{sd} of the case and the M_{sd} of the collected "weak" models. A new model will be created from the hypothesised category, the circumstance description of the case and the generalised solution. The strength will be set to "weak". This new model will replace the "weak" models whose solutions are integrated into the new model.

5 Concluding remarks

The Scottish Office records each appeal case in one file which contains documents, letters, drawings and photographs related to the case. 24 appeal cases were selected randomly in different subject areas and the summarised version of 24 appeal cases were provided in plain texts by the Scottish Office. The system is being implemented in Prolog using these cases. 24 appeal cases provided are appeal cases against three different versions of the regulations: 1971, 1981 and 1990. Since the system does not provide a versioning facility, 22 appeal cases which are applicable to the 1981 version were formalised into 25 cases in the form described in Section 4.2. The regulations related to these cases were also formalised into 102 models and added to the MKB. Background domain knowledge needed for processing 25 cases was also gathered and formalised in the DKB. We are currently implementing the retrieval procedure.

6 Acknowledgements

We would like to thank the Scottish Office for allowing us to use appeal cases.

References

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

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

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

[4] W. Kunz and H.W.J. Rittel. Issues as Elements of Information Systems. Technical Report 131, Centre for Planning and Development Research, University of California, Berkeley, USA, 1970. Working Paper.

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

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

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

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

[9] 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.

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

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

[12] 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.

[13] 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.

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

[15] Soon-Ae Yang and Dave Robertson. A Case-Based Reasoning System for Regulatory Information. In *Digest of the Colloquium on Case-Based Reasoning: Prospects for Applications, London, UK*, March 1994.

[16] Soon-Ae Yang, Dave Robertson, and John Lee. KICS: A Knowledge-Intensive Case-Based Reasoning System for Statutory Building Regulations and Case Histories. In *Proceedings of the Fourth International Conference on AI and Law, Amsterdam, The Netherlands*, June 1993.