KICS: Representation of Regulatory Information and the Use of Case-Based Reasoning to Support the Relaxation Process

Soon-Ae Yang

Ph.D.
University of Edinburgh
1995

Abstract

This thesis presents a case-based reasoning system (KICS) which can assist domain experts in interpreting building regulations in relaxation cases. In traditional legal decision support systems, it has been regarded as natural to represent legal rules in statutes in terms of If-Then decision rules and to link these rules to a separate case-based reasoning system for handling cases. However, we take a view that legal rules in the statutory regulations are the results of accumulation and generalisation of rulings made in case histories and this has led us to a unified case-based approach to handle both statutory regulations and cases.

First, we propose a unified case-based model of regulatory information. In this model, regulatory information, i.e., legal rules from statutes and precedent cases, are represented as *models*, and interpretation hierarchies of legal rules are represented as abstraction hierarchies of models in the Model Knowledge Base. Actual cases are stored in the Case Library together with arguments debated. Background domain knowledge used in classifying input cases are represented as semantic networks and heuristic rules in the Domain Knowledge Base.

Second, we propose to use case-based reasoning to access and maintain regulatory information. Models relevant to the input case are retrieved by identifying the level of abstraction at which the input case is described and by selecting models similar to the input case. If the input case is not compliant with the retrieved models, principles behind retrieved legal rules and previous similar cases are explained to the user and ask whether relaxation can be granted. Decision on relaxation is made by the user, and rulings made in cases in which relaxation is granted are acquired by generalising them and (if possible) combining them with existing models in the abstraction hierarchies.

Third, we describe the results of the evaluation of the prototype system carried out using the Scottish building regulations and 22 appeals cases from the Building Directorate of The Scottish Office. The proposed framework has been successfully used to process 22 appeal cases and has been demonstrated to be a flexible and efficient indexing mechanism. The evaluation with real-world cases has also suggested that, as the system moves towards the real-world environment, the retrieval algorithm needs to be refined and supplied with more domain-specific knowledge.

The case-based approach proposed in this thesis provides a representational framework in which legal rules at various abstraction levels including principles or legislative intentions can be explicitly represented and used to provide explanation. It also allows both statutes and precedent cases to be integrated in one knowledge representation scheme. Therefore, up-to-date interpretation of regulations can be maintained by accumulating new rulings along with existing legal rules.

Acknowledgements

I am most grateful to Dave Robertson, my first supervisor, who suggested to apply case-based reasoning to the domain of building regulations, and who has been generous with his advice and suggestions and also patient with my English.

I am also grateful to John Lee, my second supervisor, who has shown great interest in my research and has always found time for comments on my work.

Part of the work was done while I was still working as a full-time Research Assistant in the SERC Authoring Project (Grant Number: GR/F/89022). I thank the members of the project, Chris Tweed, Dave Robertson, Andrew Casson, Paul Chung, Jussi Stader, Alistair Watson, Mike Ward, Jarlath Trainer and David Stone, and the SERC for their support. My thanks also go to the British Council in Korea for supporting my work.

I am indebted to members of KBS group in the department, Alberto Castro, Peter Funk, Mandy Haggith, Jane Hesketh, Weiru Liu, Edjard Mota, Namseog Park, Maria Vargas-Vera and Wamberto W. Vasconcelos.

I am also indebted to The Scottish Office for allowing appeal cases to be used in this research. However, any views expressed in this thesis about the Scottish building regulations are those of the author alone.

Finally, I would like to dedicate my thesis to my parents.

Declaration

I hereby declare that I composed this thesis entirely myself and that it describes my own research.

Soon-Ae Yang
Edinburgh
May 31, 1995

Contents

List of Figures

List of Tables

Chapter 1

Background and Introduction

1.1 Introduction

Building regulations prescribe standards for a defined set of issues affecting building design and construction. The texts of building regulations provide a general framework for interpretation and more detailed interpretation of regulations is derived from experience in applying them in particular cases. As case histories of interpretation of regulations accumulate, previous cases are consulted in subsequent similar cases. This description of the domain suggests case-based reasoning (CBR) as an ideal methodology for developing an expert system which can assist domain experts in interpreting building regulations.

A case-based reasoner solves problems by reasoning with cases. In CBR systems, a case is "a contextualised piece of knowledge representing an experience" [Kolodner 93]. A case-based reasoner approaches a new problem by reminding of similar experiences in the past. New situations are understood in terms of old experiences by recalling previous similar cases and comparing them. Then cases are used to formulate solutions to a new problem or to generate the necessary justification or supporting argumentation. Cases in a case library (or a case memory) are indexed so that applicable cases are retrieved in subsequent similar cases. New knowledge is gained by accumulating new cases in a case library and assigning appropriate indexes.

This thesis presents a case-based reasoning system which shows how experts could be helped in interpreting building regulations by accessing and maintaining regulatory information, i.e., statutory regulations and precedent cases. We use the Scottish building regulations and relaxation cases in which the applicant seeks the relaxation of a set of particular requirements and therefore rulings set out alternative or different interpretation of regulations if relaxation is granted. Since CBR has not hitherto been applied to the domain of building regulations, we first introduce some research work done in relevant fields: AI in law in Section 1.2 and other approaches taken in this domain in Section 1.3. We then summarise the relevance of other work to this thesis and describe the outline of the thesis in Section 1.4.

1.2 AI in Law

The law contains "legal rules", which state what should be done in certain circumstances in the context of a legal system. Many of these are expressed in the form of statutes and cases. Building regulations are part of statute law and constitute part of a legislative system together with case histories. This section describes some of the characteristics of law relevant to this thesis and introduces systems which model some of these characteristics of law.

1.2.1 Legal Knowledge and Legal Reasoning

Legal concepts including legal rules are constantly changing and new concepts emerge in time. Levi describes change in the law as a cyclical process [Levi 48]:

> The first stage is the creation of the legal concept which is built up as cases are compared. The period is one in which the court fumbles for a phrase. Several phrases may be tried out; the misuse or misunderstanding of words may have an effect. The concept sounds like another, and the jump to the second is made. The second stage is more or less fixed, although reasoning by example continues to classify items inside and out of the concept. The third stage is the breakdown of the concept, as reasoning by examples has moved so far ahead as to make it clear that the suggestive influence of the word is no longer desired.

Legal rules are normally expressed in natural language. The legislature may originally have had a particular case in mind, but legal rules in statutes are usually expressed in general terms, for example, vehicle, reasonable, fair, etc. The bounds of their applicability are intentionally unclear. In other words, concepts described are often open-textured and vague. This is mainly due to the incomplete definition of legal concepts. This is inevitable because "there is no way of framing rules which are ready for all imaginable possibilities" [Susskind 87] and because legal rules and concepts "must be left ambiguous in order to permit the infusion of new interpretation" [Levi 48] as circumstances change. Therefore the application of statutes is often seen as the process of classification and subsumption: the classification of the facts of the particular case in fairly general terms, and then the subsumption of that classification within terms of one or more legal rules [Susskind 87].

Since statutory legal rules are incomplete and ambiguous, new legal rules are often made through precedent. In this sense the parties involved in cases participate in the law-making [Levi 48]. There are cases that can be decided by applying existing legal rules (clear cases) and cases that require the exercise of discretion (hard cases). The cases often become hard to decide in terms of the legislature because of the open-texturedness and incompleteness of legal rules or because of arguments from principles or consequences described below. In hard cases, issues involved in the case should be identified, and assertions by the parties involved in the case should be considered and balanced.

Legal rules provide guidance in the decision making process. Lawyers and judges, while solving legal problems as well as reasoning with rules, often have recourse to "non-standards" [McCormick 78, McCormick & Bankowski 89, Bell 89, Bell 91] or "principles" in Dworkin's terminology [Dworkin 86]. Dworkin uses this term in generic fashion so as to embrace all "principles, policies and other sorts of standards" used by lawyers. These principles state reasons that argue in one direction, but do not necessitate a particular decision. Principles, therefore, have a dimension of weight that can conflict and be balanced against one another.

Laws are, like laws in natural sciences, empirical generalisations and are not verifiable but only falsifiable [Susskind 87]. Exceptions to a legal rule may be implied on grounds of principles. All legal rules are subject to overriding principles and may be circumvented in particular circumstances. Exceptions may also be implied on account of the intention of legal rules. Even the clearest legal rule may be held not to apply to a case where that would frustrate the purpose of the law or produce absurd consequences [McCormick 78, McCormick & Bankowski 89].

1.2.2 Rule-Based Approach

This approach is the earliest steps toward expert systems in law. In this approach, legal rules are represented in terms of decision rules in If-Then format—If certain conditions are known to hold Then draw the stated conclusion or take the stated action—and then to use some form of rule interpreter to derive information from these rules.

Waterman and Paterson's LDS [Waterman *et al.* 86] use If-Then rules to encode legal doctrine and legal expertise. Given a description of a product liability case, LDS calculates defendant liability, case worth, and an equitable settlement amount. Sergot *et al* [Sergot 85, Sergot *et al.* 86] model the British Nationality Act 1981 as a set of Horn clauses [1] extended to allow negated conditions in rules. Susskind takes the Scottish law of divorce and represents some part of it in antecedent/consequent format [Susskind 87]. Susskind divides the law into small manageable units in such a way to make clear the issues involved in fixing the law's scope ("individuation"). The prototype system developed uses deductive reasoning and legal heuristics to allow for the temporal and spatial limitations and the conjunctive and exceptive qualifications which govern the condition, and for deontic content (obligation, permission, prohibition).

Legal rules represented and used in these systems are formal and prescriptive, and it is assumed that these legal rules are stable, i.e., legal rules represented are used in the middle period of the development of the law. The rule-based approach is conceptually clear and simple enough to represent such legal rules. However, from the legal standpoint, as Rissland points out in [Rissland 90], there are acknowledged difficulties in representing even formal and prescriptive and stable legal rules in If-Then rules such as the presence of imprecise terms, conflicting rules, incomplete legal rules, definitional circularity, recursion and self-referencing in statutes. Furthermore, as described in the previous section, there are cases where legal rules stated as statutes and precedents

[1] Horn clauses are rules of the form:

A if B_1 and B_2 and $\ldots$ and B_n

Each clause has zero or more conditions B_i, each of which is an atomic relationship among individuals. But each clause has exactly one conclusion A.

provide only guidance, and experts need to refer to other contextual information in making decisions.

1.2.3 Mixed Paradigm Approach

In the mixed paradigm approach, legal rules are represented in terms of If-Then rules as in the rule-based approach, and these legal rules are linked to a separate case-based reasoning system for handling precedent. Gardner's system, CABARET, GREBE and PROLEXS use this paradigm to tackle problems such as open-texturedness and incompleteness of legal rules using knowledge from rulings in case histories.

Gardner [von derLeith Gardner 87] represents legal rules and rules extracted from common sense knowledge ("CSK rules") in If-Then format, and facts of example cases ("exemplars" or "simplified precedents") as augmented transition networks. When a hard legal question arises, in other words, it is impossible to characterise the events of the input problem or competing rules are encountered or rules fail to produce a solution, the program uses exemplars to derive the answer and to check the validity of the answer. In Gardner's program, the rule-based part and the case-based part of the program are not explicitly divided. CABARET [Rissland & Skalak 91], GREBE [Branting 91] and PROLEXS [Walker *et al.* 91] have two explicitly separate systems, a rule-based reasoning system to handle legal rules and a case-based reasoning system to handle cases.

CABARET [Rissland & Skalak 91] uses a HYPO style case-based reasoner. HYPO [Ashley 91a, Ashley 91b] is a case-based system which generates legal arguments citing prior cases as justifications for legal conclusions. Ashley identifies 13 factors that affect trade secrets and related claims, and both prior cases and a problem case are represented as a collection of factors with magnitude in frames. "On-point" cases (the most relevant cases) are cases which have maximal overlaps of factors in common with the problem case. CABARET switches to the case-based reasoner when the rule-based reasoner reaches an impasse due to imprecise or open-textured terms in legal rules, incomplete legal rules, or circularity in definition. CABARET has a dedicated monitor that uses control heuristics to make observations on reasoners and to make progress toward a solution.

GREBE [Branting 91] uses both a rule-based reasoner and a case-based reasoner to construct explanations of possible conclusions about cases. The Texas workers' compensation law and related common-sense law are represented in rules, and a horn-clause resolution theorem prover is used as a rule-based reasoner. GREBE represents a case as a semantic network in which links represents the relationship between case facts. The notion of structural congruence (or structural mapping) is used to find most similar precedent cases, i.e., two structured representation of case facts are similar to the extent that they share similar configuration [Gentner 83, Gentner & Landers 89, Holyoak & Thagard 89, Thagard & Holyoak 89]. GREBE receives a new case with a proposition as a goal and tries to construct both rule-based and case-based explanation. A rule-based explanation consists of all of the antecedents of the rule whose consequent is unified with a proposition of the new case. A case-based explanation is constructed using an explanatory structure, EBE or an "exemplar-based explanation", which establishes connection between a set of critical facts of a case and the open-textured predicate to be explained. Once GREBE retrieves relevant cases, i.e., precedent cases whose configuration of critical facts of an EBE is similar to that of facts of the new case, GREBE sorts EBEs of retrieved precedent cases in order of the strength of an EBE, which is the proportion of critical facts that were matched, and prints a natural-language equivalent of EBEs.

PROLEXS [Walker *et al.* 91] represents statutes, legal doctrines from papers and books and expert knowledge in production rules and a rule-based reasoner is used to operate these rules. A case is represented in a frame as a set of facts, that is a set of conditions associated with a fixed weight. Weights reflect the degree to which satisfying the condition contributes to the relevancy of the stored case. An abstraction finder is used to detect similarity between cases. Conditions of a new case and a stored case are compared, and for each satisfied condition the case is awarded with the weight associated with the condition. If the sum of the weights of the satisfied conditions exceeds a case dependent threshold, the case is considered to match. The agenda handler keeps agenda for two reasoners and decides which reasoner may proceed next using the expert supplied strategy.

Systems described in this section open the possibility of representing a wider range of legal knowledge than rule-based systems described in the previous section. In addition to statutory legislation, Gardner's program and GREBE represent related common sense knowledge and PROLEXS represents legal doctrines and expert knowledge. They also provide possible solutions to issues related to open-texturedness and incompleteness of legal rules and determining relevancy and measuring similarity between cases. However, as Rissland points out [Rissland 90], there are limitations in those systems. First, although they represent a wider range of legal knowledge than rule-based systems, they do not take account of the importance of underlying intent or principles of legal rules in legal decision making. Second, they cannot accommodate the dynamic development of legal rules.

1.3 Related Research in the Domain of Building Regulations

Building regulations are part of a legal system together with case histories. Section 1.2.1 describes the characteristics of legal information and building regulations share these characteristics as part of legal information. Building regulations create constraints within which designers must work in designing buildings. This adds another character to building regulations: as part of design knowledge. This section describes the characteristics of building regulations as design knowledge and introduces some knowledge-based systems which address the issue of knowledge representation for building regulations.

1.3.1 Building Regulations as Design Knowledge

One view of design by [Simon 81, Logan & Smithers 89, Ganeshan *et al.* 92, Lawson 90] is as follows. Design is a goal-oriented activity that produces a description of an artifact which satisfies a set of certain goals or functional requirements. In order to produce a description of an object which satisfies these goals or functional requirements, designers have to resolve various problems that arise at different stages of the design process. Designers begin with a set of goals or objectives that represent the highest level design problems and these problems are refined into subproblems. 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.

Design problems deal with various issues [Lawson 90, Logan 89]. It is very rare for any part of a designed object to serve only one purpose. For example, Lawson [Lawson 90] identifies several issues related to window design: daylighting, sunlight, view, privacy, ventilation, sound transmission, heat loss, and structural stability. A window is required to let in daylight and sunlight, to provide a view while retaining privacy, and to offer natural ventilation. A window in an 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).

Design problems are created by designers, clients, users and legislators. Legislators of building regulations create constraints by legislating building regulations and standards and bye-laws or by making guidelines and recommendations. These constraints concern various levels of design problems, such as from the issue of energy conservation to the size of windows and the materials to be used. These constraints also concern various issues related to buildings, such as public health, fire safety, convenience of people in and around buildings, and energy conservation. Different standards may be given in the same criterion (e.g., the size of the window) to resolve different issues.

Building regulations provide intermediate solutions to design problems which emerge during the design process in the form of requirements. They are intermediate because requirements in the building standards do not specify definite and final solutions to any design problem but rather a range of solutions within which the designer must choose a solution, and these intermediate solutions, in turn, serve as subproblems to be solved to produce a final design.

1.3.2 Representation of Building Regulations in Existing Systems

The issue of knowledge representation of building regulations has been addressed in several systems. These systems were developed to help the authoring and compliance checking of building regulations.

SASE model and Hakim and Garrett's system

The SASE model [Fenves & Garrett 86, Fenves *et al.* 87] was developed to provide a formal methodology for representing the logic and organisation of building regulations so as to ensure the following qualities: individual provisions are unique, complete and correct, relations among provision are connected, acyclic and consistent, and the organisation of regulations is complete and clear. Although, as law (as described in Section 1.2.1) it is impossible for building regulations to have those qualities, the SASE model is the first major step in the computerisation of building regulations and has inspired many other systems including systems described in this section.

In the SASE model, a regulation is considered as consisting of provisions which a design must satisfy and each provision stipulates that a product or process shall have or be assigned some quality. There are four components in the SASE model: data items, decision tables and trees, information networks, and organisational systems. Data items represent all the variables occurring in the regulations. Decision tables represent the rules for assigning values to a data item. Decision trees are generated from decision tables and represent the same information in the decision tables. Decision trees prescribe the most efficient order in which to evaluate conditions and choose an appropriate action. These decision tables and trees can be regarded as a set of If-Then rules. The difference is that the order of evaluation is prescribed by the decision trees, unlike production systems which usually have no fixed order of evaluation.

The information network represents precedent relationship between data items. The SASE model provides an organisational (or classification) system in which the arrangement and scope (the subjects covered) of provisions can be represented. The organisational system provides a way of determining which provisions of a regulation apply to the situation at hand. The scope of the regulation is decomposed into several trees of classifiers. Several independent classification areas are called fields and each field is subdivided into a hierarchy of classifiers, called a facet. After the classifier facets have been built, appropriate classifiers from each facet are associated with each provision, providing a medium for accessing the provisions.

Hakim and Garrett [Hakim & Garrett 92] developed this classification system further to help designers to identify the requirements that apply to the problem which they are trying to address. The contextual concept lattice is used to represent the context in which each provision is applicable. The contextual concept lattice consists of several hierarchies of classes that are merged together to form a class lattice. Each concept hierarchy addresses a facet of classification within the standard. Subtype relationships are used to create the contextual concept lattice. All data items that have been defined in a concept superclass are inherited by its concept subclass.

Stone and Wilcox's system

Stone and Wilcox [Stone & Wilcox 88] employed the SASE model of regulations and developed two prototype systems: the writer system and the user system. The writer system provides functions for the analysis of the structure of information, the evaluation of conformance of requirements and the ordering of information based on the SASE model. The information from the writer system is converted into production rules which form a knowledge base of the user system. The user system uses a backward chaining method to provide a consultation dialogue for compliance checking.

Cornick *et al*'s model and Waard's system

Cornick *et al* [Cornick & Leishman 91] provides a more refined view of building regulations than the SASE model. Cornick *et al*'s model was developed for the integration of building regulations within CAD systems [2]. They describe building regulations as consisting of two conceptual entities: models, which describe the objects that are subject of regulations, and constraints, which limit the attributes and characteristics of the objects defined in a regulation and also place limits on the relationships that can hold between objects.

Waard employed this view of building regulations and developed a system for automatic compliance checking [deWaard 92a, deWaard 92b]. Residential buildings are modelled as a product type model based on the General AEC (Architecture, Engineer-

[2] Computer-Aided Design systems

ing and Construction) Reference Model (GARM). Building regulations are modelled as an entity-type model which consists of the entity-types involved in building regulations and constraints over those entity-types. Compliance checking is carried out in four steps. First, the system generates a filled product model database from the description of a design using the product type model. Second, from the entity-type model of building regulations, the system generates an empty regulation database which contains the entity definitions, as given in the entity-type mode of building regulations, and source code which can check a filled regulation database for conformance. Third, the system fills the empty regulation database with necessary information from the filled product model database. Fourth, the system takes the generated source code and the filled regulation database and performs the compliance checking process.

1.3.3 Discussion

Systems described in the previous section address the issue of representation of building regulations and Waard's system provides a way of integrating building regulations within CAD systems. However, the assumption behind these systems is that building regulations prescribe what requirements building elements should satisfy in terms of their attributes and relations between building elements. However, this is not the whole story of building regulations.

First, building regulations deal with several levels of concepts (hierarchical), from very abstract and descriptive to very detailed and concrete, related to various issues (multidimensional). Abstract concepts (e.g., energy conservation) provide intentions or principles that must be satisfied in the context of a legal system, or design goals that must be achieved in the context of design. More detailed and concrete concepts (e.g., the size of window and the material to be used for the window) provide requirements or intermediate design solutions with which buildings must comply in particular circumstances to achieve a certain intention or goal.

Second, there is another source of information: case histories. As law, building regulations are not complete and new legal rules are made through precedent. Relaxation of requirements prescribed in detailed and concrete terms is permitted in exceptional or unexpected circumstances provided that the intention or goal can still be achieved.

Models described in the previous section propose possible ways of representing building regulations. But they do not take into account the hierarchical and multidimensional and dynamic nature of building regulations.

1.4 Summary and the Outline of the Thesis

Building regulations have the characteristics of legal knowledge and design knowledge. Section 1.2 described the characteristics of law, and introduced systems which model some of these characteristics of law. LDS [Waterman *et al.* 86], Sergot *et al'* representation of the British Nationality Act 1981 [Sergot 85, Sergot *et al.* 86] and Susskind's system [Susskind 87] represent formal, prescriptive and stable legal rules, and they cannot cope with open-textured and incomplete legal rules. Gardner's system [von derLeith Gardner 87], CABARET [Rissland & Skalak 91], GREBE [Branting 91] and PROLEXS [Walker *et al.* 91] provide solutions to these problems by using cases and knowledge from other sources. However, the problems of accommodating changes in the legal knowledge base and modelling principles underlying legal rules have not been addressed in these systems.

Section 1.3 described the characteristics of building regulations as design knowledge, and introduced knowledge-based systems which address the issue of knowledge representation of building regulations, such as the SASE model [Fenves & Garrett 86, Fenves *et al.* 87] and Hakim and Garrett's system [Hakim & Garrett 92], Stone and Wilcox's system [Stone & Wilcox 88] using the rule-based approach, and Waard's system [deWaard 92a, deWaard 92b] using Cornick *et al*'s model [Cornick & Leishman 91]. However, the assumption behind these systems is that building regulations prescribe what requirements building elements should satisfy in terms of their attributes and relations between building elements. Therefore they do not take into account the hierarchical, multidimensional and dynamic nature of building regulations.

This thesis describes KICS [3], a system which can assist experts in interpreting building regulations in appeal cases, in which the applicant seeks the relaxation of particular requirements in statutory building regulations. Appeal cases are hard cases to decide

[3] **K**nowledge **I**ntensive **C**ase-based reasoning **S**ystem for regulatory information

and experts need to refer to principles (in Dworkin's terminology) as well as statutory building regulations and relevant cases.

Statutory building regulations can be viewed as a set of legal rules, which state a set of requirements (or a set of required design solutions) in certain circumstances. Alternative or different interpretation of statutory regulations may be set out in case histories. New legal rules may be extracted from one case or from more than one similar cases. Some legal rules from statutes and case histories state principles which are underlying intents behind legal rules or government policies. The term *regulatory information* will be used in this thesis to refer to legal rules from statutory regulations and legal rules set out in case histories.

In order to help experts to make decision in appeal cases, the system needs to provide information relevant to the case, i.e., relevant legal rules from statutes and cases. In traditional legal decision support systems, statutes and cases have been viewed as two different sorts of knowledge, and therefore it has been regarded as natural to represent legal rules in statutes in terms of If-Then rules and to link these rules to a separate case-based reasoning system for handling cases. But the representation of legal rules in If-Then rules has itself presented problems, as described in Section 1.2.2, and the same problems have raised difficulties in employing the rule-based approach in our domain (see Chapter 2). This has led us to develop a unified case-based model of statutes and cases (see Chapter 3), and to use a case-based reasoning method to handle both statutes and cases (see Chapters 4–7). The view that building regulations have the characteristics of both legal knowledge and design knowledge is reflected in the unified model of regulatory information described in Chapter 3.

The outline of the remaining part of the thesis is as follows.

- Chapter 2 first describes the characteristics and relaxation of building regulations as part of a legal system. Second it describes difficulties in using the rule-based approach in our domain and explains why we employ only the case-based approach in the system described in Chapters 3–7.

- Chapter 3 describes the unified case-based model of regulatory information and how this model is represented in the model knowledge base.

- Chapter 4 describes how cases including argumentation surrounding those cases are represented in the case library.

- Chapter 5 describes the representation of background domain knowledge which is used in classifying cases.

- Chapter 6 describes the retrieval of relevant regulatory information, which uses the notions of implied similarity as well as structural mapping, and the case evaluation procedure.

- Chapter 7 describes the explanation of relevant information i.e., principles behind regulatory information retrieved as relevant and similar cases, and the acquisition of new legal rules from relaxed cases. Only cases in which human experts decide to grant relaxation are integrated into the model knowledge base and the case itself is also stored into the case library.

- Chapter 8 describes the evaluation of the prototype system using 22 appeal cases from the Building Directorate of The Scottish Office.

- Chapter 9 provides the summary of the thesis and and proposes the possible extensions of the system.

Chapter 2

Regulatory Information and Case-Based Reasoning

The inherent nature of legal rules described in Section 1.2.1 implies problems of representing legal rules in If-Then rules and has led us to use the case-based reasoning approach rather than the mixed paradigm approach. This chapter consists of the following sections:

- Section 2.1 describes the nature of regulatory information using examples from the Scottish building regulations and the activities involved in the relaxation process.
- Section 2.2 explains the difficulties of representing legal rules from regulatory information in If-Then rules and the reason why we have employed a more uniform case-based approach in our system.
- Finally, Section 2.3 provides the overview of the system described in Chapters 3–7.

2.1 Building Regulations

The Building Directorate of The Scottish Office (the BD) is responsible for the drafting and maintenance of building regulations—the Building Standards (Scotland) Regulations and their associated Technical Standards [Sco90]—which set out the statutory technical requirements affecting building design and construction in Scotland. An ap-

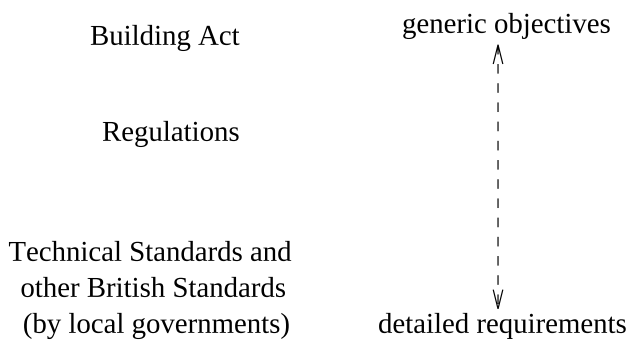

Figure 2.1: Structure of building regulations

plication for relaxation of particular requirements is first submitted to the Local Building Control Authority. If there is disagreement about interpretation, the applicant may submit an appeal application to the BD which is also responsible for the administration of the appeals procedures under the building control system in Scotland.

2.1.1 Structure of Building Regulations and Interpretation

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 [Her59] legislated by parliament explicitly states these generic objectives:

> An Act to make as respects Scotland new provision for safety, health and other matters in respect of the construction of buildings and for safety in respect of the conduct of building operations; $\cdots$

The Building (Scotland) Act 1959 also enables the Secretary of State for Scotland to make the Building Standards (Scotland) Regulations and their associated Technical Standards [Sco90], in which these issues or principles are translated into more detailed technical requirements. Figure 2.1 illustrates the overall structure of building regulations.

The Building Standards (Scotland) Regulations contain 24 regulations which state requirements for the sixteen subject areas to achieve the generic objectives quoted 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 the Building Standards (Scotland) Regulations should be interpreted. For example, the Standard J2.2 describes how the 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 attached to J2.2 which provides another way of satisfying the standard J2.2. Therefore there are at least four ways of complying with Regulation 22: by satisfying J2.2 or J2.3 or J2.5 or provisions to J2.2. For example, J2.3 describes requirements in terms of U-values [1] of building fabric and the areas of openings in walls and roofs (see Figure 2.2).

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 example shown in Figure 2.2, 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 factors, for example the material of building fabric or the shape and position of openings in walls and roofs, are not specified. U-values specified in J2.3 are a performance target for designers which can be met by

[1] Thermal transmittance coefficients.

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 2.2: The Standard J2.3

a variety of combinations of materials or arrangements of openings. It is up to the designer to decide what specification or layout should meet the required performance. Cases are also described in the same manner. They do not provide a design of a whole building but provide only the description of aspects or features of a 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. As a consequence, new legal rules are set out in unexpected circumstances or slightly different circumstances and existing legal rules may be relaxed in exceptional circumstances, provided that the legislative intentions can still be met. Therefore new ways of interpretation 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 rulings 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.

2.1.2 Relaxation

An application for relaxation of particular requirements is first submitted to the Local Building Control Authority (the LBCA). If there is disagreement about interpretation between the applicant and the LBCA, the applicant may appeal to the Secretary of the State by submitting an appeal application to the BD in The Scottish Office. The appeal application consists of a form, (usually) a set of drawings, a list of regulations against which relaxation is sought, and other administrative information (e.g., names, addresses, etc.). The application is processed in four stages [Tweed 90]. The first stage is to make sure that the application provides all essential information. The application is given a reference number and the application form is examined to make sure it has been completed fully and correctly, and is checked that the necessary drawings have been enclosed. Any missing information is requested from the applicant. The second stage is to consult all information available to the experts. After consulting all information available, the third stage is to produce a recommendation, i.e., the final decision on relaxation. The final stage is a filing/archiving process. We describe the consultation process in more detail in this section. Our description of this process is by no means complete but is to provide a simplified model which allows case-based reasoning to be employed to support experts working in the consultation process.

The consultation process can be seen as consisting of two processes: the fact-finding process and the argumentation process. In the fact-finding process, the experts will not only search for related information through legal documents or research papers from the building research institutes but also call upon their own knowledge built from experience of dealing with previous applications. Through this process, the experts need to identify the information about the proposed building, the regulations at issue and the issues behind those regulations.

First, the information about the proposed building is obtained from the application form and accompanying drawings. The purpose group of the building (or the building category) and the description of part of the building that requires relaxation are stated in the application.

Second, the experts need to consider the regulations. Although the statutory regulations at issue are listed in the application, experts still need to identify the issues behind specific regulations, government policies and principles taken for granted in the field of building regulations. There are two sources from which such information can be found. First, some information about issues can be found in legal documents including the statutory building regulations themselves and research papers. Second, some information about issues is gained through experience of processing previous applications. More context-sensitive issues can be identified by examining how decisions on previous applications were made on what grounds in what circumstances. Previous case files are the only tangible sources of such information.

In the argumentation process, the interested parties, i.e., the applicant, the LBCA or experts in related fields, are invited to discuss different proposals suggested. The decision is made through argumentation. The experts in the BD consider the positions and the arguments suggested by the players and decide which position is reasonably acceptable in the particular circumstances of the case. The argumentation surrounding each appeal decision is often complex, involving the balancing of a variety of assertions by the parties concerned. The decision process is often influenced by social rather than technical considerations.

It would be ideal to build a system which includes all information to which experts need refer in this process but that would be too ambitious. We restrict ourselves to a system which has a knowledge base of regulatory information, i.e., legal rules from statutory regulations and case histories, and therefore can support the consultation of these loosely associated legal rules without being forced to make explicit all the information needed to arrive at a decision. The system does not make any decision about relaxation but is intended to provide the experts with relevant information (i.e., relevant legal rules from statutory regulations and cases) so that they can arrive at a decision.

For each appeal case, the BD keeps a file of all information referred to, which usually includes letters and drawings and sometimes photographs, and correspondence between players who participate in the decision process. There are 187 recorded appeal cases from the 1980's and 1990's. 24 cases were randomly selected for the prototype system and summarised in one or two pages. The summaries of these 24 cases are listed in Appendix A and have been used in developing and evaluating the prototype system. [2]

2.2 Why Not a Mixed Paradigm Approach ?

In existing systems described in Section 1.2.3 which handle both legal rules and cases, the mixed paradigm approach has been regarded as natural, i.e., legal rules from statutory law and other legal sources are represented in If-Then rules and handled by a rule-based reasoning system and cases are handled by a separate case-based reasoning system. But the nature of legal rules described in Section 1.2 and the previous section gives difficulties in employing a rule-based approach for legal rules and has led us to use only a case-based approach for both legal rules and cases rather than the mixed paradigm approach. In this section we explain why.

2.2.1 Rule-Based Approach Revisited

In general, law is considered as a set of legal rules which define legal concepts. These naturally fit into the If-Then rule form which can be interpreted as:

I1: If certain conditions are known to hold
Then draw the stated conclusion

If-Then rules are sometimes interpreted procedurally as follows:

I2: If certain conditions are known to hold
Then take the stated action

Legal rules which provide a general framework of interpretation are definitional. There are of course legal rules which detail exemptions, classification of terms and legal proceedings but we concentrate on the representation of I1 type rules here.

[2] Summaries were provided by the BD. The work was actually done by Patrick Shaw, an architecture student, during June–August 1993, supervised by David Stone.

A traditional approach to the representation of legal rules is: legal rules are represented as If-Then rules in a knowledge base, and some form of rule interpreter is used to derive information from legal rules represented in the knowledge base. This approach has been taken in Waterman and Paterson's LDS [Waterman *et al.* 86], Sergot et al's representation of the British Nationality Act 1981 [Sergot 85, Sergot *et al.* 86] and Susskind's system [Susskind 87]. The Scottish Office's Building Directorate has also experimented with this approach in their in-house prototype systems [Stone & Wilcox 88].

Waterman and Paterson's LDS [Waterman *et al.* 86] represents both I1 and I2 type rules. How to calculate defendant liability, case worth, and an equitable settlement amount is represented as a "schema", which is a set of three types of "rulesets": procedural, predicate and generator rulesets. In a given condition, a procedural ruleset performs some activity such as updating the database or notifying the user some information. Predicate rulesets provide a way of computing the truth or falsity of a "proposition" (in legal rules). Generator rulesets produce instances of a class of items. A predicate ruleset shown in [Waterman *et al.* 86] is: [3]

> To decide a plaintiff does manifest radiographic-evidence of asbestosis:
> If (the plaintiff does exhibit parenchymal-findings of asbestosis
> and the plaintiff's parenchymal-findings are not explained by alternatives
> and the plaintiff's parenchymal-findings are not contested)
> or (the plaintiff does not show pleural-plaques in x-ray
> and the plaintiff's pleural-plaques are not explained by alternatives
> and the plaintiff's pleural-plaques are not contested),
> Conclude that
> the plaintiff does manifest radiographic-evidence of asbestosis.
> End.

Susskind's system [Susskind 87] and Sergot et al's representation of the British Nationality Act 1981 [Sergot 85, Sergot *et al.* 86] represent legal concepts in I1 type rules. They represent the structure of statutes using and/or decision trees, in which a proposition in the antecedent part of a rule may be defined by another rule(s) whose consequent part is the antecedent of this rule and whose antecedent part provides definition. Forward and backward chaining is used to find related rules. In [Sergot *et al.* 86], a legal rule which defines British Citizenship is represented as follows:

[3] For ease of understanding, antecedent and consequent in If-Then rules are described not in computer-executable form but in approximate English throughout this Chapter.

x is a British citizen
if x was born in the UK
    and x was born on date y
    and y is after or on commencement
    and z is a parent of x
    and z is a British citizen on date y

A proposition "z is a parent of x" is defined in the following two rules:

z is a parent of x if z is mother of x;
z is a parent of x if z is father of x

2.2.2 Representation of Regulatory Information in If-Then Rules

Legal rules from regulatory information can be seen as requirements which state 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

Stone and Wilcox' system [Stone & Wilcox 88] represents legal rules from the Scottish building regulations in this form. For example, the Standard J2.3 can be represented in this form as follows.

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

Legal rules used in Stone and Wilcox' system [Stone & Wilcox 88] and systems described in the previous section are formal and prescriptive like J2.3. However, this sort of legal rules is only part of regulatory information. Appeal cases are hard cases and, in relaxation, experts need refer to less formal and descriptive legal rules which state government policies and principles, for example, the generic objectives of the building regulations stated in The Building (Scotland) Act 1959 and Regulation 22. The text of statutory legal rules merely provides a general framework within which the interpretation of statutory legal rules is established through precedent. The nature of regulatory information described in Section 2.1 can be summarised as the following four points and we will continue our discussion under the headings of these four points.

1. Legal rules have interpretation hierarchies.
2. More general terms are used in legal rules than in cases.
3. The meaning of terms is often in debate.
4. New legal rules are constantly emerging.

1. Legal rules have interpretation hierarchies.

In addition to legal rules which are prescriptive and state requirements at a very detailed level such as J2.3, there are legal rules which are less prescriptive, and state the government policies and principles. These rules are often used to explain why these detailed requirements should be satisfied in the context of the legal system. The generic objectives of the building regulations stated in The Building (Scotland) Act 1959 and Regulation 22 are examples of these legal rules.

Legal rules are independently described at various levels of detail. Legal rules which state the generic objectives and legal rules from 24 regulations describe requirements using more abstract terms which have broader meaning. For example, the generic objectives stated in The Building (Scotland) Act 1959 can be encoded into four If-Then rules:

R1: If there is a building
Then the building must ensure the safety of people
R2: If there is a building
Then the building must ensure the convenience of people
R3: If there is a building
Then the building must ensure the conservation of energy
R4: If there is a building
Then the building must ensure the public health

Regulation 22 states that buildings should make reasonable provision for the conservation of fuel and power. Experts know from experience that reasonable provision for the conservation of fuel and power means reasonable provision to limit heat loss, and this can be represented in a If-Then rule as follows:

R5: If there is a building
Then the building must be built to limit heat loss

R5 does not describe very much detailed requirements than R3 but this rule is needed to represent intermediate requirements establishing a connection between R3 and R6 described below. At the levels of R1–R4 and R5, terms like "safety", "building", "energy conservation" and "reasonable provision" are used. One or more legal rules can be introduced to provide various ways of interpreting these abstract legal rules. For example, J2.3 describes requirements which should satisfy to comply with Regulation 22 in terms of "building fabric" and J3 describes requirements in terms of "building services". Therefore, first, these requirements can be represented as:

R6: If there is a building
Then the building must provide reasonable provision for building fabric
R7: If there is a building
Then the building must provide reasonable provision for building services

Second, R6 can be redescribed as more detailed rules which describe how building fabric should provide reasonable provision to ensure the conservation of energy in terms of "openings", "walls", "floors" and so forth. One example of rules at this level is R8 shown below:

R8: If the type of building is dwelling and
there are floors next to the ground
Then the U-values of the floor must not exceed 0.45

In summary, legal rules at lower levels are a refinement (or specialisation or decomposition) of legal rules at upper levels. Although it may be possible to overcome the problem of different levels of detail in legal rules by using appropriate rule structuring such as decision trees or and/or graphs in Sergot et al's representation [Sergot 85, Sergot *et al.* 86] and Susskind's system [Susskind 87], there would still remain other difficulties, as described below.

2. More general terms are used in legal rules than in cases.

Since more general terms are used to express legal rules than cases, there exists a terminological gap between legal rules and cases. The language used in legal rules is generally at a more general level of description than that used in describing a particular situation in cases. The description of legal rules is the description of requirements generalised from many particular circumstances. Required solutions in legal rules specify a range of solutions rather than a particular solution. A case arises in one of those circumstances and propose one solution which may or may not within the range of the required solution.

For example, legal rules at the same level as R8 is stated using terms such as "dwelling" and "window". In cases, the type of building may be given as "flat" or "house", which are of type "dwelling", or instead of "window" the term "rooflight" may be used if the window is located in the roof.

It would be possible to bridge these terminological gaps by introducing domain-specific bridging rules such as generator rulesets in LDS, but it is not always possible reliably to bridge this terminological gap in this way, as we demonstrate below.

3. The meaning of terms is often in debate.

It is sometimes a matter of debate as to what the legal rules mean by basic concepts. For example, the issue in case 85 is whether or not embankment is acceptable as "a place of safety", and the issue in case 90 is whether or not "timber with 1-hour fire-resistance" can be regarded as "non-combustible material". [4] 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.

It might be possible to encode in If-Then rules how these terms are interpreted in particular circumstances. But there would remain the serious practical problem of maintaining consistency across the vast collection of these rules, given that the interpretation of the regulations constantly changes over time and there is always the possibility of conflicts between these rules.

Another way of solving this problem is to build a separate case-based reasoning system which keeps such appeal cases and to refer to cases when open-textured and conflicting

[4] Case 85 and case 90 are shown in Appendix A.

rules are encountered. This approach was taken by CABARET [Rissland & Skalak 91], GREBE [Branting 91] and PROLEXS [Walker *et al.* 91].

4. New legal rules are constantly emerging.

Even if the above theoretical and practical objectives could be addressed by appropriate rule structuring, bridging rules and a case-based reasoning system, there would still remain the problem of accommodating the changes in the interpretation of legal rules within the system. New legal rules are set out in new, unexpected circumstances and existing legal rules may be relaxed in exceptional circumstances. These new legal rules should be added to the general framework of interpretation of legal rules.

Legal rules represented in If-Then rules in systems described above are formal and prescriptive (i.e., terms used are well-defined and therefore there is little or no difficulty in classifying given terms) and stable (i.e., rules are more or less fixed and used in the middle period of the development of law). Statutory legal rules cannot be complete by nature and precedents often set out new legal rules in unexpected or exceptional circumstances.

2.2.3 Use of CBR in This Domain

We take a view of legal rules as follows:

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

Legal rules in the statutes are the result of accumulation and generalisation of 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. However, precedent cases provide a way of interpreting a particular set of legal rules in the statutes under particular circumstances—in other words, they provide 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. As 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), it seems appropriate to represent legal rules from both statutes and cases using only a case-based model and to handle them by case-based reasoning method. A case-based reasoning system has been developed using this approach and is described in the following five chapters (Chapters 3–7). In this system, legal rules, i.e., the generalised cases, are represented as *models*, and cases are kept in a separate case library. Legal rules from statutory regulations are represented as "strong" models and legal rules generated from cases are represented as "weak" models. Interpretation hierarchies which exist among legal rules are represented as abstraction hierarchies of models. Each case is relevant to a model or models and links between cases and models can be established using the notion of relevance. There are various matching algorithms which can be used to access models at various levels in the abstraction hierarchies. Models in the abstraction hierarchies are loosely connected to each other so that new legal rules can be added as "weak" models to the existing abstraction hierarchies even when new "weak" models are somehow inconsistent with the existing models.

2.3 Overview of the System

This section presents an overview of the system. The detailed description of each component of the system shown in Figure 2.3 is provided in Chapters 3–7.

There are three knowledge bases: the Model Knowledge Base, the Case Library and the Domain Knowledge Base. Legal rules are represented in the Model Knowledge Base (see Chapter 3). In addition to legal rules, background domain knowledge must be used to bridge terminological gaps between models at different levels of abstraction hierarchies and between models and cases in the retrieval and maintenance of legal rules, and cases and argumentation surrounding each case must be provided to the user. Cases and argumentation surrounding each case are represented in the Case Library (see Chapter 4). Background domain knowledge is represented in the Domain Knowledge Base (see Chapter 5).

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

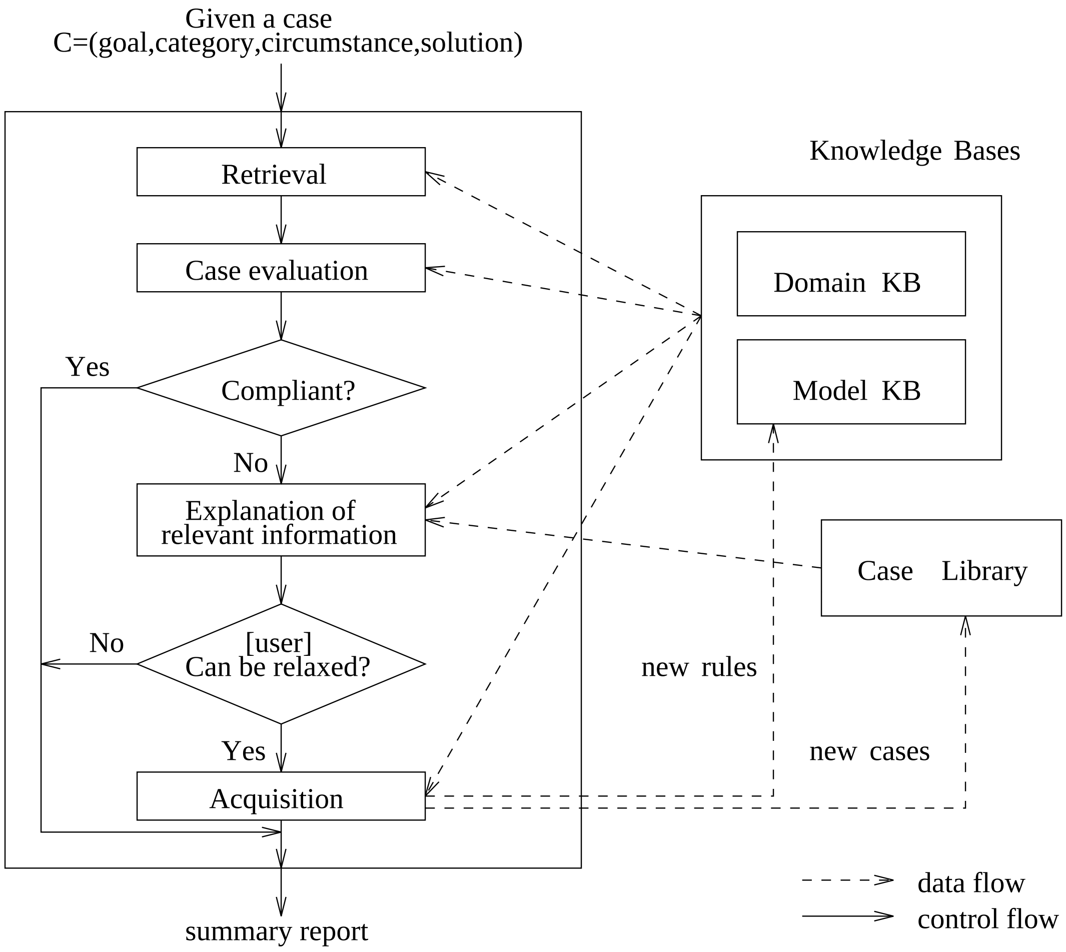

Figure 2.3: System overview

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 overall procedure of the system is as follows.

1. Retrieval: The system first retrieves relevant models by comparing goal, category and circumstance parts of the given case with models in the MKB. The system uses the notions of implied similarity as well as structural mapping (see Chapter 6).

2. Case evaluation: 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 (see Chapter 6).

3. Explanation of relevant information: If the case is not compliant, there is a possibility of relaxation and the system will explain relevant regulatory information and similar cases. An argumentation structure is used to explain key points in arguments surrounding the relevant cases (see Chapter 7).

4. Acquisition: If the expert decides to grant relaxation, new legal rules set out by the case will be integrated into the existing abstraction hierarchy by generalising new rulings and (if possible) combining them with relevant weak models. The case itself will also be stored into the Case Library (see Chapter 7).

Chapter 3

Representation of Regulatory Information and the Model Knowledge Base

Legal rules are represented as models in the Model Knowledge Base (the MKB). In the MKB, interpretation hierarchies of legal rules should be represented explicitly so that the system can retrieve legal rules which are more general and can be used to provide explanation. The MKB also needs to be able to support the changes in legal rules. Regulatory information has the characteristics of design knowledge as well as legal knowledge. Some design systems have addressed the issue of representing design knowledge in the context of design and we have taken one of these approaches to represent regulatory information in the MKB. This chapter includes the following sections.

- Section 3.1 recapitulates the characteristics of design information and regulatory information to see the similarity which exists in the development of these two sorts of information.
- Section 3.2 introduces some design systems which have addressed problems of structuring and maintaining design knowledge.
- Section 3.3 describes how legal rules are represented and organised in the MKB.
- Section 3.4 describes the implementation of the MKB in the prototype system.

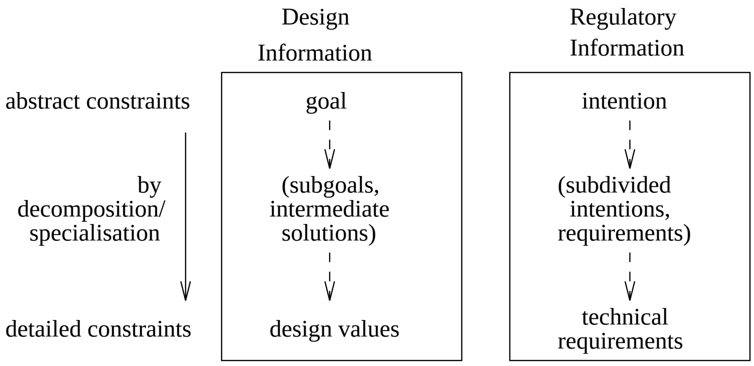

Figure 3.1: Development of design information and regulatory information

3.1 Design Information and Regulatory Information

Legislators create regulations which achieve certain intentions. These intentions are typically the government policy and the principles underlying the regulations. These are expressed initially in very general terms and then translated into more detailed technical requirements using more detailed and concrete terms. Detailed requirements provide various ways of achieving intentions or requirements expressed in more general terms, and these detailed requirements may be translated into a set of yet more detailed requirements if necessary.

Designers start designing an object with a set of goals or functional requirements in their minds. This set of goals or functional requirements represents the highest level design problem. In order to produce a description of an object which satisfies these goals or functional requirements, the highest level design problems may be decomposed and refined into several subproblems or specialised using more specific terms. These refined or specialised subproblems are intermediate solutions and they serve as subgoals to which the designer should attend.

Development of regulatory information is similar to designing an object. The similarity between these two processes is illustrated in Figure 3.1. A set of goals or functional requirements in design or a set of intentions in regulations provides a starting point in the development process, and also provides most abstract constraints which the detailed design of an object should satisfy or which in regulations the detailed technical requirement should satisfy. More abstract constraints may be decomposed into constituent constraints or specialised into more detailed constraints using more specific terms. [1]

3.2 Design Systems

Some design systems take the view of design described in the previous section and address problems of structuring and maintaining design problems and solutions. This section describes three design systems: Gero *et al*'s system, Logan's MTS (Multi-Theory System) and Ganeshan *et al*'s system.

3.2.1 Gero *et al*'s System

Gero *et al* [Gero & Rosenman 89] emphasise the importance of experience in design and propose "prototypes" as a framework for storing and processing separate design cases or generalised design experience. They argue that experience recorded in prototypes can be used in design in two ways: first, in routine design, by selecting and copying the design of the prototype which is applicable to the design problem at hand, and second, in innovative and creative design, by changing the contents of the selected prototypes. A prototype contains functional requirements represented as a set of variables, knowledge of how to map between functional requirements and design description, and a design context (i.e., particular areas of interest). Prototypes are implemented in a framelike structure in which part and partition relations are used to implement inheritance of properties.

Prototypes in this system make a clear distinction between design problems and design solutions and this system may be used to record the initial design problem and the final solution. But it is difficult to distinguish design problems and design solutions during the design process and therefore this system cannot be used to record design problems and solutions which emerge during the design process. This system also does not take into account interaction between design problems either.

[1] We are not the first who noted the affinity between two seemingly different domains, design and law. See Chapter 11 in [Collins 71] for instance.

3.2.2 Logan's MTS

In [Logan 89, Logan & Newton 90], Logan proposes a representation system, called the Multi-Theory System (MTS), in which the designer can conceptualise the structure of a design problem according to the designer's preference of structuring and processing design information. In MTS, the structure of a design problem is conceptualised as a set of theories. A theory represents a series of (weak) relationships between constraints. Constraints are represented as arbitrary statements or propositions describing the problem at any level of abstraction. Satisfaction of statements is dependent on the attainment of the low-level statements which form its definition. Dependencies between theories are defined in terms of subset relationships.

Introduction of theories makes it possible to partition a problem into a number of distinct models each of which conceptualises alternative design solutions. There are two drawbacks however. First, as described in Section 1.3.1, design problems are hierarchical and interactive, and the subset relationship is too restricted to represent such design problems. Second, it takes no account of the dynamics of the design process itself, and therefore it cannot accommodate possible changes in design problems and solutions during the design process.

3.2.3 Ganeshan *et al*'s System

Ganeshan *et al* propose a model for an interactive computer-based design environment in which the designer can build a model of intents (objectives or design goals) with some assistance of the system such as generating alternatives, evaluating and selecting alternatives [Ganeshan *et al.* 92]. The system records the design process by recording all states of each step of the design, starting from the highest-level design problems (or design objectives), including all their refinements, and relating these to the decision. These design records can be used to explain why a particular decision was made, to identify the ramification of changes in the design, and to synthesise a design from a previous design with a similar specification.

In this model, design intents are represented as a hierarchy of objectives which is developed during the design process. An objective is represented as a desired relationship over variables which represent a physical object or a property measured over these physical objects. Importance of objectives is expressed as ordinal ranking or weights and they are used to select an alternative which best satisfies the current design objective. Assumptions are weak requirements that are usually based on the experience and judgement of the designer. An objective is refined by generating alternatives to the objective. An objective can be decomposed or specialised into subobjectives or associated with a set of constraints that specify the conditions which the objective must satisfy to be valid.

Designers can reject the system's choice and correct the model of their intent by changing the importance of objectives or guiding the selection of alternatives using unstated objectives. Conflicts occur when no alternatives are generated for refinement or when none of the alternatives generated satisfies all the relevant objectives. The system tries to resolve conflicts by: changing an arbitrary selection, or changing a heuristic evaluation or selection, or changing the importance associated with some objective. The design process is led by the designer and the system provides only limited assistance.

We use Ganeshan *et al*'s system's approach for structuring legal rules in the Model Knowledge Base (the MKB). The next section describes how regulatory information is organised and represented in the MKB.

3.3 Representation of Regulatory Information

The MKB consists of models, which represent individual legal rules from statutory regulations and case histories. Legal rules state requirements which must be satisfied under particular circumstances to satisfy certain intentions or goals. Legal 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. Legal rules from case histories act as weak constraints in the sense that, as the interpretations of legal rules in the statutory regulations change over time, legal rules set out in case histories may or may not apply to new cases.

To build a knowledge base which contains legal rules at various abstract levels, we can start with a set of legal rules which state legislative intentions and decompose and specialise the initial set of legal rules into more detailed legal rules. In the Scottish building regulations, [2] we can find four legal rules (R1–R4) which state the most abstract intentions: all buildings should be built to ensure the safety (R1) and convenience (R2) of people in and around buildings and the conservation of energy (R3), and not to harm the health of the public (R4). 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 and the total areas of single-glazed openings in those walls and roofs do not exceed the prescribed areas. 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. For example, the regulations (J2.3) prescribe U-values only for floors next to the ground (R8), exposed floors (R9), semi-exposed floors (R10), exposed walls (R11), and semi-exposed walls (R12). The regulations (J2.3) also prescribe different U-values and areas of openings to those components of different categories of buildings. For example, in dwellings, the U-value of roof should not exceed 0.25 (R13), but in other categories of buildings the threshold is 0.45 (R14). In dwellings, the total area of single-glazed openings in walls should not exceed 15 percent of the total floor area (R15).

These legal rules should apply to buildings which fall into certain situations in which these rules are stipulated. However, new legal rules may be set out in circumstances not covered by legislation and existing statutory legal rules may be relaxed in exceptional circumstances. Here is an example. The regulations (J2.3) 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 example, 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 3.2 shows the structure of legal rules as four hierarchies.

[2] Examples of legal rules used in this section are extracted from the regulations shown in Section 2.1.1.

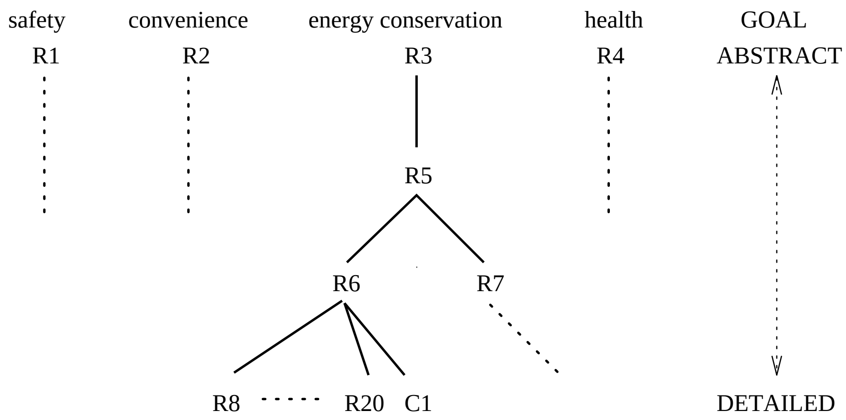

Figure 3.2: Interpretation hierarchies of legal rules

Note that, at the top and second levels of the hierarchies legal rules are applied to a building, at the third level the building fabric which is part of a building, and at 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 legal rules are to apply become more concrete and therefore the scope of application of legal rules becomes more restricted.

Each legal rule is represented as a model M which consists of five parts: $M = (M_{id}, M_c, M_{cd}, M_{sd}, M_s)$ where M_{id} is the id number of the model, 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 the statutory regulations, or as "weak" otherwise. M_{cd} and M_{sd} are described in terms of objects, their attributes and relations between objects.

For example, R3, R5–R6, R15 and C1 in Figure 3.2 can be represented as M3, M5–M6, M15 and M21 respectively as follows: [3]

M3:

category:	building
circumstance:	building(A)
solution:	ensure(A,energy_conservation)
strength:	strong

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

M5:	
category:	building
circumstance:	building(A)
solution:	limit(A,heat_loss)
strength:	strong
M6:	
category:	building
circumstance:	building_fabric(A)
solution:	provide(A,reasonable_provision)
strength:	strong
M15:	
category:	dwelling, industrial_building, storage, warehouse
circumstance:	window(A), attribute(A,single_glazed), wall(B), attribute(B,exposed), part_of(A,B)
solution:	percentage_area(A,C), C =< 15
strength:	strong
M21:	
category:	dwelling, industrial_building, storage, warehouse
circumstance:	window(A), attribute(A,double_glazed), wall(B), attribute(B,exposed), part_of(A,B)
solution:	percentage_area(A,C), C =< 17
strength:	weak

An abstraction hierarchy of models can be built from the interpretation hierarchy of legal rules. It is also possible to construct several hierarchies, each of which represents a hierarchy of one intention. For example, four abstraction hierarchies can be built from four different subjects (R1–R4). Figure 3.3 shows four model hierarchies constructed from interpretation hierarchies in Figure 3.2. It is assumed that if there are two models at different levels the model at the upper level has greater strength than the model at the lower level. Abstraction hierarchies in Figure 3.3 are also used to explain the retrieval of relevant models in Section 6.2 and the acquisition of new regulatory information in Section 7.2–7.3. In the implementation and evaluation of the prototype system, the MKB shown in Appendix B was used.

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 all descendant models of this model neither apply to the given case. The category description together with this hierarchical structure of models are used as effective indices in the retrieval of relevant models (described in Chapter 6).

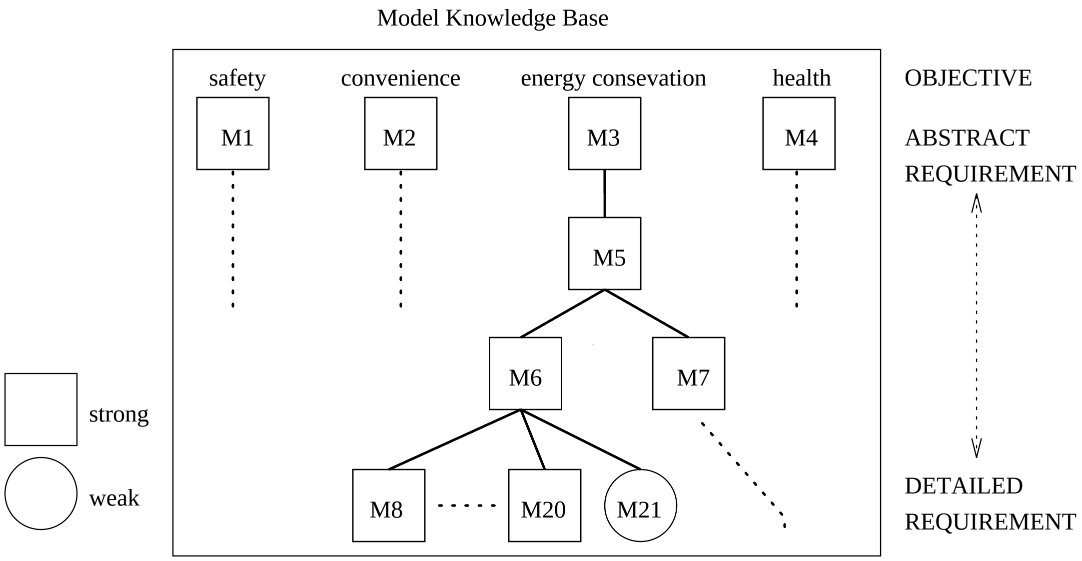

Figure 3.3: Abstraction hierarchies of models

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 indices to cases stored in the case library as well as the interpretation hierarchies of information about regulations.

3.4 Implementation

102 legal rules related to 24 cases shown in Appendix A were extracted from the 1981 version of the Scottish building regulations [Sco81] and formalised into 102 models. Four abstraction hierarchies of these 102 models were built into the MKB and used for the implementation and evaluation of the prototype system. The Prolog implementation of this MKB is shown in Appendix B.

Chapter 4

The Case Library and the Argumentation Structure

Cases and the argumentation surrounding each case are stored in the Case Library. Section 4.1 describes how summaries of appeal cases were formalised. Section 4.2 describes the argumentation process for relaxation. Section 4.3 introduces two existing models of argumentation from which our argumentation structure was developed, and describes the argumentation structure which is used to record key points in argumentation surrounding an appeal case. Finally, Section 4.4 describes the implementation of the Case Library in the prototype system.

4.1 Formalisation of Appeal Cases

Among 187 recorded appeal cases in the 1980's and 1990's, 24 cases were randomly selected for the prototype system. 9 cases (case file number 11, 41, 42, 77, 82, 85, 87, 90 and 93) are appeal cases against the 1971 version [Sco71], 13 cases (case file number 95, 97, 110, 116, 131, 132, 134, 141, 150, 154, 156, 165 and 172) are against the 1981 version [Sco81], and 2 cases (case file number 0004 and 0012) are against the 1990 version [Sco90]. The 1981 version of Scottish building regulations was chosen for the prototype system for the following two reasons.

1. The system does not provide a versioning facility. Therefore the system cannot accommodate different versions of regulations.

2. 11 cases are appeal cases against regulations in the 1971 and 1990 versions. The 1981 version contains the same set of regulations against which 10 of these 11 cases appealed.

Case 0004 appealed to relax requirements on the gradient of ramps, which the 1981 version does not include, and case 0004 was excluded in the list of test cases.

For each appeal case, the Building Directorate of The Scottish Office (the BD) keeps a file of all information referred to, which includes letters, drawings, photographs, and correspondence between players who participate in argumentation for relaxation of a particular regulation. The following 14 items were used to summarise information in the files of 24 selected appeal cases into the shorter text form shown in Appendix A.

CASE FILE NUMBER The number attached to the case file.

PURPOSE GROUP The purpose for which the building is built. For example, residential building, shop, office, theatre, gym, etc.

STATUS Whether the building is an existing building or a new building.

REG/VERSION The section of regulations and the year of legislation against which the applicant appealed. For example, K10/1981.

CLAUSE The clause(s) particularly relevant to the case.

SUBJECT This roughly corresponds to the headings of each part of regulations. For example, fire safety, convenience, health, etc.

ELEMENTS Building components related to the **ISSUE** described below.

RELATIONSHIP Whether the issue is about a relationship between two (or more) building elements.

DATE Approximate dates during which the case was processed.

ISSUE The description of proposed construction or alteration and issues.

ARGUMENTS Arguments from players.

DETERMINATION Decision of whether the BD granted or refused relaxation.

CONDITION Conditions attached to **DETERMINATION** if there are any.

REASONS The reason why relaxation was granted or refused.

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, which is the case file number, 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. Once a case has been processed within the system, this case will be stored in the Case Library.

We define the Case Library as a collection of cases processed by the system. A case C in the Case Library consists of eight parts: $(C_{id}, C_g, C_c, C_{cd}, C_{sd}, C_d, C_m, C_t)$ where $(C_{id}, C_g, C_c, C_{cd}, C_{sd})$ is the same as the corresponding parts in the case description provided as input, C_d is the decision about relaxation taken by the BD, i.e., "granted" or "refused", C_m contains the id number of models which were retrieved as relevant to this case, or "nil" if there was no relevant model, and other information not represented by $(C_g, C_c, C_{cd}, C_{sd}, C_d, C_m)$ but which is useful will be stored in C_t. C_t can be stored in texts or other form allowed in Prolog. In the current implementation, C_t contains the id number of a strong model (or strong models) which is retrieved as relevant and therefore the case was supposed to satisfy but it did not satisfy.

Table 4.1 shows the correspondence between items in the case summary and parts of the case representation within the system, and the proposed use of some items. **CASE FILE NUMBER**, **SUBJECT**, **PURPOSE GROUP** and **DETERMINATION** were translated into C_{id}, C_g, C_c and C_d respectively. **ISSUE**, **RELATIONSHIP** and **ELEMENTS** were translated into C_{cd} and C_{sd}. C_{cd} and C_{sd} were extracted mostly from **ISSUE**. But how to divide **ISSUE** into C_{cd} and C_{sd} was not clear in all cases and the following guidelines were used to divide **ISSUE** into C_{cd} and C_{sd}.

- All building components appearing in **ISSUE** are represented in C_{cd}. Properties of those components and relationships between those components which are necessary to describe C_{sd} are also represented in C_{cd}.

purpose	case summary	representation and use
As input and in the Case Library	CASE FILE NUMBER	C_{id}
	SUBJECT	C_g
	PURPOSE GROUP	C_c
	ELEMENTS RELATIONSHIP ISSUE	C_{cd} and C_{sd}
In the Case Library	DETERMINATION	C_d
	STATUS ISSUE ARGUMENTS CONDITION REASONS	to be recorded as argumentation
In evaluation	REG/VERSION CLAUSE	to be compared with the models retrieved by the system to check the correctness of the retrieval procedure
	DETERMINATION	to be used as an expert's decision on relaxation in the acquisition procedure
	DATE	to be used to determine the order of cases to be supplied to the system

Table 4.1: Translation and use of information from case summaries

- C_{sd} represents only properties or structural relationships of those building components in C_{cd} which are required to satisfy a particular regulation at issue.

STATUS, **ISSUE**, **ARGUMENTS**, **CONDITION** and **REASONS** explain why relaxation was granted or refused. The information about the status of the building needs to be provided because statutory building regulations apply to the construction of a new building or a new part of an existing building or the alteration, extension or demolition of an existing building or part of an existing building. The analysis of 24 appeal cases shows that, although the Local Building Control Authority (the LBCA) tends to insist on full compliance when the relaxation application is made for a construction or alteration of part of an existing building, relaxation is sometimes granted in such cases if it is unreasonable or impracticable for specific requirements to be applied. This argumentation surrounding each case is recorded in a network structure described in Section 4.3.4 and will be used to explain key points in cases, as described in Chapter 7.

4.2 The Argumentation Process for Relaxation

A pattern of argumentation was identified from the careful examination of 24 appeal cases. Main players in argumentation are the applicant, the LBCA and the BD. The argumentation process for the relaxation of specific part(s) of regulations starts with usually two positions, one suggested by the applicant and the other one or exceptionally two suggested by the LBCA. Since the applicant is the player who seeks relaxation on the regulations at issue, the applicant's proposal constitutes partial compliance with the regulations. Appeal cases are submitted to the BD because the LBCA refused relaxation and insisted on full compliance. Therefore, the position suggested by the LBCA is the solution required by the statutory building regulations in the given circumstances. Occasionally, the LBCA also suggests an alternative solution which the LBCA considers reasonable in the given circumstances. [1] There may be other players in the argumentation process, such as the BD itself or the firemaster, who does not propose any position but make arguments to support or oppose the suggested positions.

Each player may make arguments to support or oppose the suggested positions. Arguments are the description of facts relevant to the issue and are made with reasons which explain why this argument should be used to support or oppose the position. The role of the BD is to arbitrate between two main players—the applicant and the LBCA—and decide which position is reasonably acceptable in the context of the given case. When the relaxation is granted, the BD takes the view that the applicant's position—i.e., partial compliance—is acceptable in the given context. When the relaxation is refused, the BD takes the view that the LBCA's position—i.e., full compliance—is acceptable in the given context.

4.3 The Argumentation Structure

To record the key points in argumentation surrounding each appeal case, we use an argumentation structure which was developed from Toulmin's model and the IBIS model of argumentation. This section first introduces these two models and then we describe our argumentation structure.

[1] Case 11 shown in Appendix A is an example.

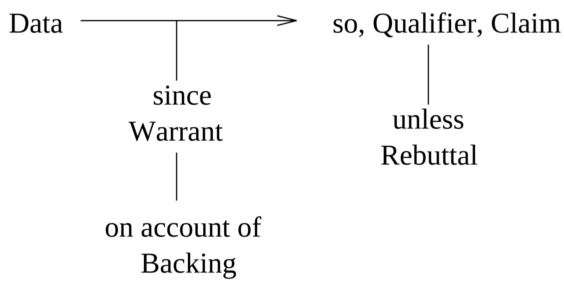

Figure 4.1: Toulmin's model of argumentation

4.3.1 Toulmin's Model

Toulmin [Toulmin 58] examines the progress of an argument from the initial statement of an unsettled problem to the final presentation of a conclusion, and formulates a form of argument illustrated in Figure 4.1. This form includes the following six elements involved in establishing a claim or conclusion by producing arguments.

- Claim: The claim or conclusion whose merits it is being sought to establish.
- Data: The facts proposed as a foundation for the claim.
- Warrant: The practical standards or canons of argument which explain the legitimacy of the step from Data to Claim.
- Qualifier: A modal qualifier which confers varying degrees of force on the conclusions or the claims.
- Rebuttal: Conditions of exception in which the general authority of the warrant would have to be set aside.
- Backing: The general or categorical statements of the facts which back the Warrant and therefore explain why in general the warrant should be accepted.

If we compare this model of argumentation to the argumentation process described in Section 4.2, Claims correspond to positions, Data correspond to arguments, and Warrants correspond to reasons which explain why an argument supports or opposes a position. But there is no part corresponding to the suppose/oppose relationships

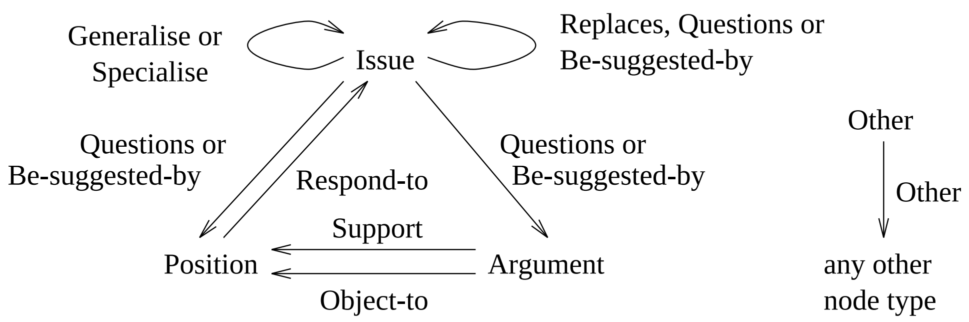

Figure 4.2: The IBIS model of argumentation

which exist between arguments and positions. In Toulmin's model, there exist links between elements but it does not allow the definition of various kinds of relationships between elements.

4.3.2 The IBIS Model

The IBIS (Issue-Based Information System) model was developed by Kunz and Rittel [Kunz & Rittel 70] to support coordination and planning of political decision processes by guiding the identification, structuring and settling of issues raised by problem-solving groups. IBIS recognises six elements involved in the argumentation process, "topics", "issues", "positions", "arguments", "questions of fact" and "model problems"; and relationships between issues and positions, such as "issue A is a direct successor of issue B", "issue A is a generalisation of issue B", "issue A is a relevant analogy to issue B", and "a position taken in response to issue A is compatible, consistent or incompatible with a position assumed in response to issue B". IBIS contains several subsystems: an issue bank, an evidence bank, a handbook (which is a collection of model problems), a topic list, an issue map and a documentation system. This model was further developed and refined in gIBIS (graphical IBIS).

gIBIS [Conklin & Begeman 88, Begeman & Conklin 88] is a hypertext system designed to support the collaborative construction of IBIS networks. Conklin and Begeman view the design process as the resolution of design issues through argumentation, and developed gIBIS to capture the rationale behind design, which includes design problems, alternative resolutions including those rejected, trade-off analyses among alternatives, and design decisions.

In gIBIS, there are four types of nodes: Issue, Position, Argument and Other. Participants in argumentation can bring their expertise and viewpoints to the resolution of design "Issues". Any problem, concern, or question can be an issue. Each Issue can have many "Positions". A Position is a statement or assertion which resolves the Issue. Each of an Issue's Positions may have one or more "Arguments" which either support that Position or object to it. "Other" nodes can have any information.

There are nine kinds of links: Respond-to, Support, Object-to, Generalise, Specialise, Replace, Question, Be-supported-by and Other. A Position "Responds-to" an Issue. Arguments must be linked to their Positions with either "Supports" or "Objects-to" links. Issues may "Generalise" or "Specialise" other Issues and may also "Question" or "Be-suggested-by" other Issues, Positions and Arguments. "Other" links may connect Other nodes to any other node type. Figure 4.2 shows all node types and the possible link types between the node types.

This model allows the definition of various types of relationships between elements arising in argumentation. But this model does not have elements to record the reasons why a particular argument supports or opposes a particular position, which play an important role in deciding relaxation.

4.3.3 Discussion

In the previous two sections, we described Toulmin's model and the IBIS model of argumentation. But neither of these models provides an entirely adequate set of elements and relationships between elements for representing the argumentation process in appeal cases. We take parts of these two models and combine them to define our argumentation structure as shown in the table below.

Toulmin	IBIS	KICS
–	Issue	C_{cd} and C_{sd}
Claim	Position	Position
Data	Argument	Argument
–	Support	Support
–	Object-to	Oppose
Warrant	–	Reason attached to Support/Oppose

The issue in a case is translated into C_{cd} and C_{sd} as described in Section 4.1. This corresponds to Issue in the IBIS model. There are two more elements we can identify in the argumentation process for relaxation: *positions* and *arguments*. Positions correspond to Claims in Toulmin's model and Positions in the IBIS model. There exist two sorts of relationships between positions and arguments: *support* and *oppose*. Support/oppose relationships correspond to Support/Object-to link types in the IBIS model and there is no corresponding part in Toulmin's model. There are *reasons* attached to these support/oppose relationships. Reasons correspond to Warrant in Toulmin's model and there is no corresponding part in the IBIS model.

4.3.4 Our Argumentation Structure

There are two node types: position and argument. Positions represent design proposals suggested by players. There are at least two positions. First, there is the position submitted by the applicant, i.e., a design proposal which constitutes only partial compliance to the regulations at issue. Second, there is the position suggested by the LBCA, i.e., a design proposal which ensures full compliance with the building regulations at issue. And there may be other design proposals suggested by the LBCA as alternatives to the solution required by the statutory building regulations. Arguments are the description of relevant circumstances or facts to support or oppose a position.

There are two types of links which connect positions and arguments: support and oppose. These two types of links represent the relations between positions and arguments: an argument supports/opposes a position. Reason(s) for supporting or opposing a position should be attached to support/oppose links. Reasons can be one of the following: "full compliance", "partial compliance", "non-compliance", "reasonable", "unreasonable", "practicable", "impracticable" or "predicates" which rep-

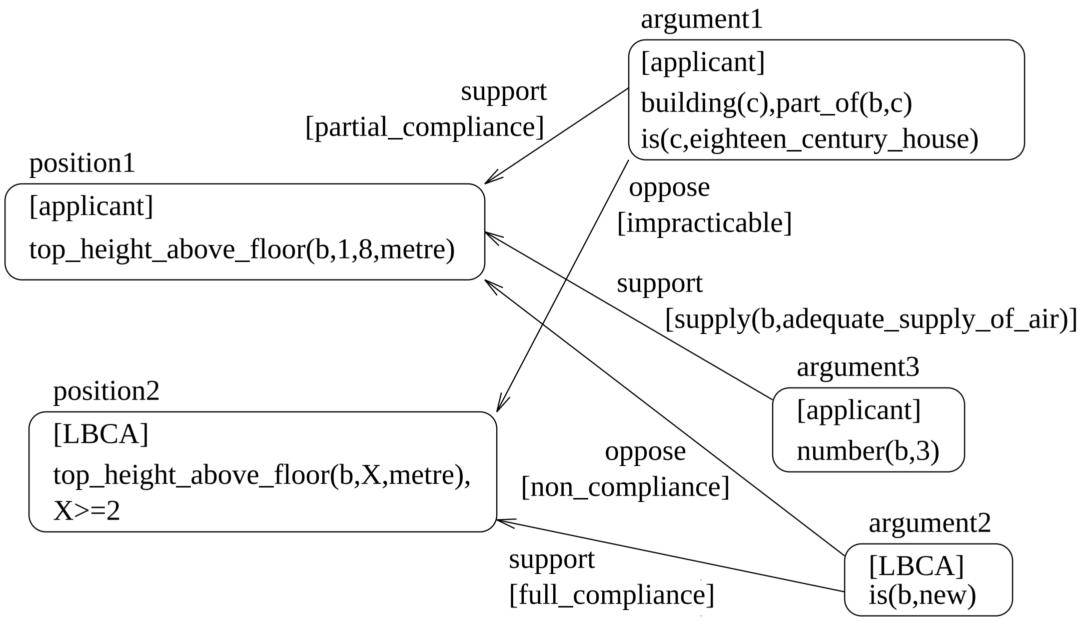

Figure 4.3: Argumentation surrounding the case 93

resent reason other than these predefined seven reasons (for instance "window(b), supply(b,adequate_supply_of_air)"). These attachments should be read as one of the following depending on the type of links and the reason attached to it.

- The argument supports/opposes the position because the proposal described as the position constitutes full compliance/partial compliance/non-compliance based on the facts described as the argument.

- The argument supports/opposes the position because the proposal described as the position is a reasonable/practicable/impracticable solution in the circumstances described as the argument.

- The argument supports/opposes the position because, under the circumstances described as the argument, the proposal described as the position has the consequence described as "predicates".

An argumentation structure extracted from case 93 is shown in Figure 4.3 as an example. Reasons to support/oppose a position are enclosed in brackets ([$\cdots$]) in this figure.

4.4 Implementation

Two cases, case 0004 and case 85 are excluded in the list of test cases. As explained in Section 4.1, Case 0004 appeals to relax requirements on the gradient of ramps, which the 1981 version does not include, and case 0004 is excluded. The reason why case 85 is excluded is explained in Section 5.3.

Case 90 is an appeal case for relaxation of regulations about materials used in exit stairways which consist of stairs, division walls and floors. Regulations specify requirements for each building component related to a particular subject, and therefore case 90 was divided into three sub-cases: c90_1 for stairs, c90_2 for division walls and c90_3 for floors. For the same reason, case 154 was also divided into two sub-cases: c154_1 for the floor in the second floor and c154_2 for the floor in the third floor. Appendix C shows the Case Library after all 25 test cases were processed by the system.

Chapter 5

Representation of Background Domain Knowledge and the Domain Knowledge Base

Background domain knowledge is represented in the Domain Knowledge Base (the DKB) and used in classifying a given case, i.e., in retrieving models whose M_c and M_{cd} are similar to C_c and C_{cd} respectively (see Chapter 6). We use semantic networks to represent background domain knowledge and define similarity between two concepts using the notion of semantic distance in the semantic networks. This chapter consists of three sections:

- Section 5.1 introduces the semantic network formalism and the notion of semantic distance.
- Section 5.2 describes how background domain knowledge is represented in the DKB and how similarity between two concepts is defined.
- Section 5.3 describes how the DKB is implemented.

5.1 Semantic Networks and Semantic Distance

Network representation of knowledge first appeared in Quillian's computational model of human long-term memory [Quillian 85]. The term "semantic networks" was initially used because the model was devised to model the meaning of natural language in a network formalism. However, the term "associative networks", which puts more emphasis on association between concepts, is also widely used.

In Quillian's model, a definition of a word (in dictionaries) is represented as a "plane", which is a mass of token nodes headed by a type node (a patriarchal type node) and interconnected by different kinds of associated links. Intraplane links are used as: a subclass-superclass pointer, a modification pointer, a disjunctive set, a conjunctive set or any other relationships. A word's full concept is defined as "all the nodes that can be reached by an exhaustive tracing process originating at its initial patriarchal type node, together with the total sum of relationships among these nodes specified by intraplane links". There is no restriction to prevent reentries or loops within a full concept. There is no predetermined hierarchy of superclasses and subclasses. Every word is the patriarch of its own hierarchy, and every word is defined in terms of the configuration of other words in memory.

Comparing and contrasting two concepts is done by spreading activation intersection search (or "spreading activation" for short, or "marker passing"). First, the program searches for points in the memory at which two full concepts intersect. Tags are used to mark where the search originated. Then, possible relations between them can be found in two paths from the intersection node to the two patriarch nodes where search begins.

Collins and Quillian [Collins & Quillian 72] also first described the notion of semantic distance. By semantic distance they mean "accessibility" from one concept to the other, and they gave a more formal description of semantic distance in terms of semantic memory described in [Quillian 85]. Numbers can be assigned to accessibilities such that lower numbers reflect more accessible properties, and then semantic distance is the sum of these numbers along the path between two concepts. Under this definition, we can say that one concept is closer to a given concept than another if the sum of accessibilities to the first is less than the sum of accessibilities to the second. The path between two concepts can be found by spreading activation intersection search or marker passing, as described above. Collins and Quillian also identified semantic properties of concepts which permit certain kinds of inferences to be made: superset or subordinate relations, subset relations, similarity relations, part relations, proximity or adjacency relations, consequence or causality relations and precedence relations.

Since Quillian's model, there have been many systems which use semantic networks as a knowledge representation formalism, such as KR systems in the KL-ONE family [Brachmann 79, MacGregor 91], conceptual dependency [Schank 72], propositional semantic networks [Maida & Shapiro 85, Shapiro 91] and conceptual graphs [Sowa 84, Sowa 91]. Every system has its own version of a semantic network formalism but, as Brachmann points out in [Brachmann 79], one thing that we can find in common among all is their connectivity, i.e., nodes and links. Therefore, the most important issue in the semantic network representation lies in the structural decomposition of concepts to find a semantic network notation suitable for given tasks. When one devises a semantic network notation, it is necessary to specify the following [Brachmann 79, Woods 85]:

- The types of nodes and links that can be used and the rules of their possible combinations, i.e., the syntax of the network notation.
- What is meant by them, i.e., the semantics of the network notation.
- An algorithm or procedure for translating knowledge expressed in sentences into this representation.
- Algorithms or procedures which can make use of this representation for the subsequent inferences and deductions that the human or machine must perform on them.

5.2 Representation of Background Domain Knowledge

The Domain Knowledge Base contains background domain knowledge to which the system needs to refer when classifying cases. There are two kinds of information needed to be represented in the DKB:

- Aggregation relationships between building components.
- Specialisation relationships between two concepts, i.e., building categories, building component types and attributes which building components may have.

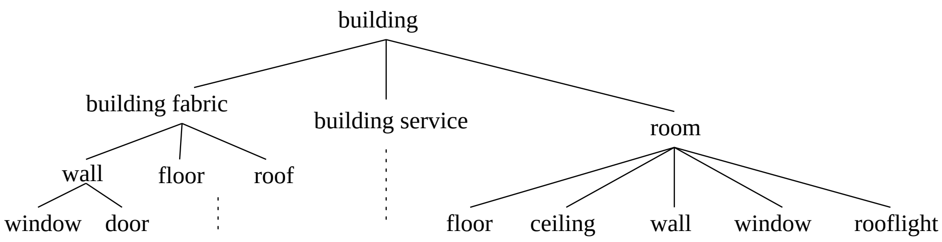

Figure 5.1: An aggregation hierarchy of building components

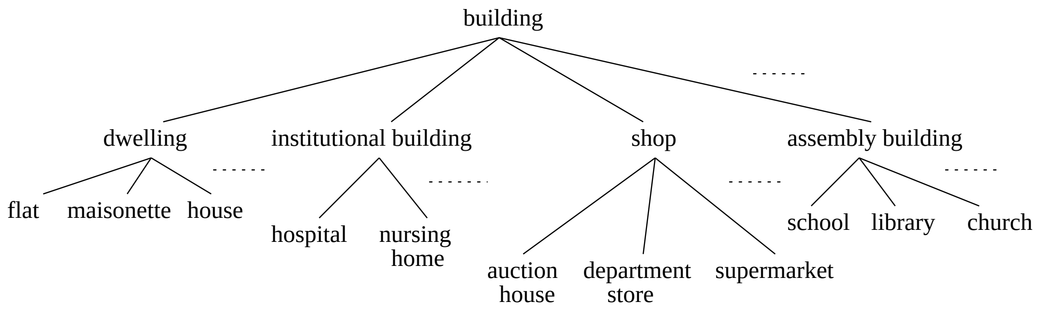

Figure 5.2: A specialisation hierarchy of building categories

Information about aggregation relationships between building components are used to select models at the right abstraction level (see Section 6.2.2). Examples of aggregation relationships between building components are shown in Figure 5.1.

Information about specialisation relationships between two concepts is used to identify similarity between two concepts. There are three kinds of specialisation hierarchies used to represent specialisation relationships.

First, there is a specialisation hierarchy of building types, which are used to select models whose building categories listed in M_c subsumes the category given in C_c (see Section 6.2.1). Part of this specialisation hierarchy is shown in Figure 5.2.

There are two other kinds of specialisation hierarchies: specialisation hierarchies of building component types and attributes, which are used to identify similarity between two situations, i.e., similarity between M_{cd} and C_{cd} (see Section 6.2.3). Examples are shown in Figure 5.3.

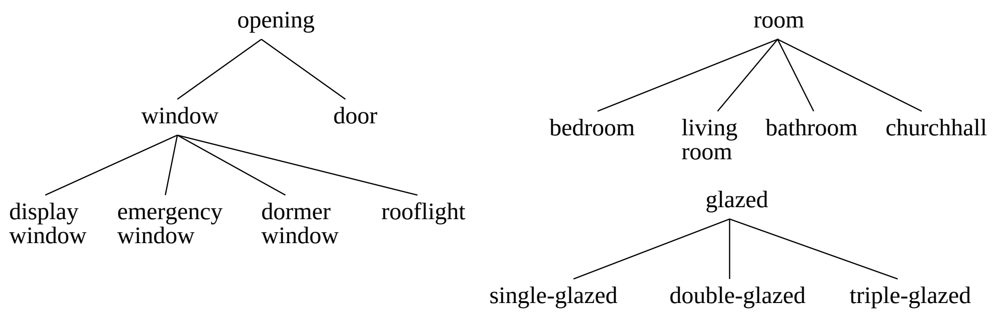

Figure 5.3: Specialisation hierarchies of object types and attributes

Note that the specialisation hierarchy of building categories are strictly hierarchical (i.e., a building category appears only once in the hierarchy) but in other aggregation/specialisation hierarchies a concept may appear more than once in a hierarchy or may appear in more than one hierarchy.

Two levels of similarity must be distinguished in classifying a case, i.e., in retrieving models whose M_{cd} are similar to C_{cd}. First, similarity between two concepts needs to be defined, for example, similarity between two building component types in M_{cd} and C_{cd}. Once specialisation hierarchies are defined, similarity between two concepts can be defined using semantic distance between two nodes in a specialisation hierarchy. Second, similarity between two situations, i.e., similarity between M_{cd} and C_{cd}, needs to be defined. Similarity between two situations is in turn defined using similarity between individual concepts and is described in the next chapter.

We define the similarity between two concepts in terms of the number of links between two nodes in a specialisation hierarchy, and how we decided the magic number, i.e., the maximum semantic distance between two concepts which ought to be regarded as similar, is described in the next section.

5.3 Implementation and Discussion

There are 26 test cases. The MKB contains 102 models. A DKB was built to cover all these test cases and models and is shown in Appendix D. To build the DKB, first all the vocabulary appearing in category and circumstance descriptions of cases and models

case:model	object types in Ccd	object types in Mcd
c42:m116	gable bedroom	room
	dormer window	opening
c77:m65	tread	stair
c82:m23	loadbearing partition wall	separation wall
c85:m52	embankment	place of safety
	corridor	passageway
c87:m108	church hall	room
	door	opening
	window	opening
c90_2:m26	division wall	compartment wall
c93:m116	bedroom	room
	window	opening
c95:m39	games room	room
c110:m18	room	place
c132:m201	gent toilet facility	sanitary convenience
c134:m115	storage room	room
c141:m106	window	opening
c165:m56	automatic sliding door	door

Table 5.1: Comparison of objects in cases and models

(Cc/Ccd and Mc/Mcd) was collected, and then aggregation/specialisation hierarchies were built. We define the similarity between two concepts in terms of the number of links between two concepts in specialisation hierarchies. The magic number, i.e., the maximum semantic distance between two concepts which ought to be regarded as similar, was decided after comparing concepts used to describe the circumstances of each appeal case, i.e., C_{cd} of each test case, and concepts used to describe the circumstances of statutory regulations relevant to each appeal case, i.e., the M_{cd} of relevant model(s) in the MKB.

The object types used to describe the C_{cd} of the 26 test cases and the object types used to describe M_{cd} of the models relevant to these 26 test cases are listed in Table 5.1. Object types in C_{cd} are one of the subclasses of corresponding object types in M_{cd} and the semantic distance between them is 1 except in the following three cases.

- In c42, “dormer window” is a subclass of “window” and “window” is a subclass of “opening”. Therefore semantic distance between “dormer window” and “opening” is 2.

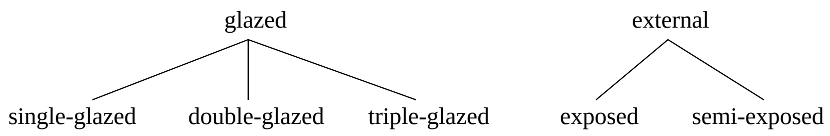

Figure 5.4: The specialisation hierarchy of attributes

- In c85, whether "embankment" is to be regarded as a "place of safety" is the issue in c85, and therefore the subclass/superclass relationship between "embankment" and "place of safety" cannot be defined.
- In c90_2, the object "division wall" in C_{cd} is a superclass of "compartment wall" in M_{cd} and the semantic distance between them is 1.

Since the subclass/superclass relationship between "embankment" and a "place of safety" cannot be defined, c85 was excluded in the list of test cases. We define an object type O_C used in C_{cd} to be similar to an object type O_M used in M_{cd} if:

1. O_C is a subclass of O_M and the semantic distance between them is not greater than 2; or
2. O_C is a superclass of O_M (i.e., O_M is a subclass of O_C) and the semantic distance between them is not greater than 1.

The following attributes are used to describe cases and models:

- Attributes of walls: exposed, external, semi_exposed, enclosed, unenclosed.
- Attributes of windows: glazed, single_glazed, double_glazed, triple_glazed.

We can build two specialisation hierarchies as shown in Figure 5.4, one for the attributes of walls and the other for the attributes of windows, and add them to the DKB. Then we define an attribute A_C used in C_{cd} to be similar to an attribute A_M used in M_{cd} if A_C is a subclass or a superclass of A_M and the semantic distance between them is not greater than 1.

The following list shows the vocabulary used to represent relations between objects:

- Structural relations between objects: part_of, attached, on, separate_from, connected, connected_by, direct_to.
- Other relations between objects: ensure, provide, inhibit, reduce, flout, block, remove, conform, is, require, used_as, can, covered_with, give_access_to.

No sensible specialisation hierarchy can be found for these relations. Therefore similarity between two concepts is defined only for object types and attributes.

Note that since specialisation hierarchies of object types are represented in Prolog predicates, we use unification instead of spreading activation to count links between two object types.

Chapter 6

Retrieval of Relevant Regulatory Information

The previous three chapters described the three knowledge bases used by the system: the Model Knowledge Base, which consists of abstraction hierarchies of models which represent interpretation hierarchies of legal rules; the Case Library, which contains cases and the argumentation surrounding each case; and the Domain Knowledge Base, which contains background domain knowledge. This chapter and the next chapter describe how the system makes use of three knowledge bases to help experts make decisions on relaxation when a new case arises. First, this chapter explains how relevant legal rules stored in the Model Knowledge Base are retrieved and how the case is evaluated against requirements from retrieved legal rules. A new case will be represented and provided to the system 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. A case needs to satisfy only those legal rules which apply to the case. Whether or not a legal rule should apply to the case is determined by assessing similarity between the circumstances in which the legal rule is stipulated and the circumstances in which the case occurs. The content of this chapter is as follows:

- Section 6.1 introduces the notions of similarity between two situations (represented as M_{cd} and C_{cd}) used in the retrieval of relevant legal rules (represented as models).

- Section 6.2 describes how relevant models are retrieved.
- Section 6.3 describes how the conformance of the case is evaluated against the retrieved models.
- Section 6.4 describes how these are implemented and discusses issues raised during implementation.

6.1 Similarity between Two Situations

M_{cd} and C_{cd} describe key features of circumstances and M_{cd} is said to describe a situation similar to that of C_{cd} if M_{cd} and C_{cd} satisfy one of the following similarity criteria:

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

The first criterion "simple similarity" is adapted from two theories in analogical mapping: Structural Mapping Theory [Gentner & Landers 89, Gentner 83] and Constraint-Satisfaction Theory [Holyoak & Thagard 89, Thagard & Holyoak 89]. The other two criteria, "embedded similarity" and "implied similarity" are defined as extensions of the "simple similarity" criterion. This section first introduces Gentner's and Holyoak and Thagard's theories in analogical mapping and then describes the similarity criteria used in our system.

6.1.1 Analogical Mapping

Analogical mapping is referred to a mapping of knowledge from one domain (the base) into another domain (the target), which indicates that a system of relations that holds among base objects also holds among the target objects. The base and the target are represented as a set of predicates, which denote features such as objects, object attributes and relations. The process of mapping can be seen as the construction of *correspondences* or *mappings* between the features of the base and those of the target. This section introduces two theories that were adapted for our system: Gentner's Structural Mapping Theory and Thagard and Holyoak's Constraint-Satisfaction Theory.

Gentner's Structural Mapping Theory (SMT)

In Gentner's Structural Mapping Theory [Gentner & Landers 89, Gentner 83] finding correspondences of features is purely syntactic matter. This theory assumes that the base and the target are represented using the following four representational constructs: entities, attributes, relations and functions.

- Entities represent individual objects and constants.
  For example, ball, table.

- Attributes are predicates taking one argument.
  For example, RED(ball), SQUARE(table).

- Relations are predicates taking one or more arguments. First-order relations are predicates which take objects as arguments.
  For example, HIT(ball,table).
  Second- and higher-order relations such as IMPLIES and CAUSE are predicates which take propositions as their arguments.
  For example, CAUSE[HIT(cue stick, ball), ENTER(ball, pocket)].

- Functions map one or more entities into another entity or constant.
  For example, SPEED(ball) maps the physical object ball into the quantity that describes its speed.

Assuming the representation above, Gentner decomposes similarity into different kinds of similarity and describes them in terms of mapping predicates from the base to the target.

- literal similarity: Both (first-order) relational predicates and object attributes are mapped.
- analogy: Only (first-order) relational predicates are mapped.
- mere appearance: Only object attributes are mapped.
- relational abstraction: (First-order) relational structures in the base are mapped into more abstract relational structures in the target.
- systematicity: Higher-order relational predicates are mapped.

Systematicity takes precedence when a mapping is constructed between the base and the target. The object and function correspondences are determined by their roles in the systematic relational structure.

For example, in water/heat analogy described in [Gentner & Landers 89], first, water-flow situation such as "the pressure of water in a large beaker is greater than the pressure of water in a small vial and water flows from the beaker to the vial through the pipe which connects the beaker and the vial" is represented as:

CAUSE{GREATER[PRESSURE(beaker), PRESSURE(vial)],
[FLOW(water, pipe, beaker, vial)]}
GREATER[DIAMETER(beaker), DIAMETER(vial)]
liquid(water)
flat-top(water)
cylindrical(beaker)

Second, heat-flow situation such as "a silver bar is standing in a mug with warm coffee, an ice cube is stuck at the end of this bar, and heat flows from coffee to ice through the bar" is represented as:

CAUSE{GREATER[TEMP(coffee), TEMP(ice)],
[FLOW(heat, bar, coffee, ice)]}
liquid(coffee)
flat-top(water)

Then according to Gentner's theory similarity between these two situations can be explained as follows:

- There are object correspondences:

  water and heat;
  pipe and bar;
  beaker and coffee; and
  vial and ice.

- There is a functional correspondence:

  PRESSURE and TEMP.

- The pressure difference structure in the water domain is mapped into the temperature difference structure in the heat domain, i.e., there is systematic similarity:

  CAUSE{GREATER[PRESSURE(beaker), PRESSURE(vial)],
  [FLOW(water, pipe, beaker, vial)]} and
  CAUSE{GREATER[TEMP(coffee), TEMP(ice)],
  [FLOW(heat, bar, coffee, ice)]}.

- Object attributes and isolated relations are disregarded.

Thagard and Holyoak's Constraint-Satisfaction Theory (CST)

In Constraint-Satisfaction Theory [Holyoak & Thagard 89, Thagard & Holyoak 89], Thagard and Holyoak proposed three constraints to be used as guidance in analogical mapping: structural consistency, semantic similarity and pragmatic centrality. In this theory, the base and the possible targets are assumed to be represented as a set of objects and predicates. The structural consistency constraint favours correspondences that are isomorphic to the base representation, i.e., the target whose objects and predicates are mapped one-to-one into objects and predicates in the base. The semantic similarity constraint supports correspondences between elements which have similar meanings. Similarity between elements depends on semantic relations such as being represented by the same predicate, or being represented by predicates similar in meaning by superordinate/subordinate (kind-of) relations, part/whole (part-of) relations, synonymy or other semantic relations. How similarity between two elements are measured and how two elements are determined to be similar are not described in [Holyoak & Thagard 89, Thagard & Holyoak 89]. The pragmatic centrality constraint favours correspondences that are pragmatically important to the analogist, that is, relevant to the purposes of the mapping.

Gentner's SMT	Holyoak & Thagard's CST	objects	attributes	(first-order) relations
literal similarity	structural consistency	C	C	C
analogy	-	C	F	C
mere appearance	-	C	C	F
-	semantic similarity	S	S	S
abstraction	-	(relations are mapped into more abstract relations)		
systematicity	-	(higher-order relations are mapped)		
-	pragmatic centrality	(the purpose of analogy is relevant)		

Note: C means *C*orrespondence, F means *F*ew or no correspondence, and S means *S*imilar.

Table 6.1: Comparison of SMT and CST

Discussion

The previous two sections described Gentner's structural mapping theory and Holyoak and Thagard's constraint-satisfaction theory and Table 6.1 summarises these two theories. Literal similarity in Gentner's theory and structural consistency in Holyoak and Thagard's theory are the same kind of similarity. In abstraction in Gentner's theory, relations in the base are mapped into more abstract relations in the target. Abstraction relationships between relations can be represented by superordinate/subordinate (kind-of) relations and more abstract relations can be regarded as similar relations according to the superordinate/subordinate relations. Therefore, semantic similarity in Holyoak and Thagard's theory includes the notion of abstraction in Gentner's theory.

Their theories deal with cross-domain similarity, i.e., similarity between the base in one domain and the target in the other domain. We are dealing with intra-domain similarity, i.e., similarity between the descriptions of two situations that describe a particular aspect or feature of a building. The notion of analogical mapping described in this section is adopted to define intra-domain similarity in our domain. First, in Gentner's and Holyoak and Thagard's theories, objects are represented as constants such as "water", "heat" and "beaker". In our system, objects are represented using object names and object types. Each object (i.e., building component) is given a unique name such as "a" and "b", and the type of an object such as "window" and "wall" is represented as a predicate with an object name as an argument. Second, in Gentner's theory, second-order (or higher-order) relations are used to explain similarity. To define the notion of similarity in our domain, only first-order predicates are used. Therefore, three kinds of first-order predicates are used to represent key features of a building in our system:

- Object types are represented as predicates which take an object name as an argument.
  For example, window(a), wall(b).
- Object attributes are represented as predicates which take two arguments, an object name and an attribute value.
  For example, attribute(a,external).
- Relations are represented as predicates which take two or more object names as arguments.
  For example, part_of(a,b).
  Properties of objects such as height and width of window are also represented as a relation between an object and their values.
  For example, height(a,1,metre), width(a,0.6,metre).

Third, correspondence between two objects is recognised only when the predicates which represent the types of these two objects are exactly the same or similar. Similarity between two concepts, i.e., two object types, two attributes or two relations, is defined in terms of semantic distance in specialisation (i.e., superordinate/subordinate) hierarchies as explained in Section 5.2. Once the correspondences between objects are established, attributes of corresponding objects and relations between them are compared to see whether there is similarity. To avoid any confusion, we use the term "match" instead of "correspondence" when comparing object types, attributes and relations. By "A matches B" we mean "A" is exactly the same as "B". "Correspondence" will be used only between two objects whose object types are the same or similar.

6.1.2 Simple Similarity

We first define three kinds of simple similarity in this section: literal similarity, semantic similarity and analogy. The definition of simple similarity is derived from Gentner's

C1

C_g:	energy_conservation
C_c:	house
C_{cd}:	window(a), attribute(a,double_glazed), wall(b), attribute(b,exposed), part_of(a,b)
C_{sd}:	percentage_area(a,17)

Figure 6.1: Case C1

SMT and Holyoak and Thagard's CST: literal similarity from the notion of literal similarity in SMT and structural consistency in CST; semantic similarity from the notion of semantic similarity in CST; and analogy from the notion of analogy in SMT. Embedded similarity and implied similarity which are defined using the notion of simple similarity are described in the subsequent sections. Examples used to describe these similarity criteria are variants of the case C1 (or M21) explained in Section 3.3 and C1 is shown in Figure 6.1. They are given in the context of "energy conservation". Their C_c is "house". Their C_{cd} is described in terms of "wall", "window", "roof", "bedroom", their attributes and relations between them. They should comply with the requirement that the total area of single-glazed windows in walls should not exceed 15 percent of the total floor area. This requirement is represented as model M15 shown in Figure 6.2.

Literal similarity

All object types match, in other words, there is one-to-one correspondence between objects in M_{cd} and objects in C_{cd} and their object types are exactly the same. All attributes of corresponding objects are matched and all relations between corresponding objects are also matched. For example, suppose that the C_{cd} of case C2 is given as follows.

C2

C_{cd}:	window(a), attribute(a,single_glazed), wall(b), attribute(b,exposed), part_of(a,b)

M15
M_c: dwelling, industrial_building, storage, warehouse
M_{cd}: window(A), attribute(A,single_glazed),
wall(B), attribute(B,exposed),
part_of(A,B)
M_{sd}: percentage_area(A,C), C =< 15
M_s: strong

Figure 6.2: Model M15

There is literal similarity between the M_{cd} of M15 shown in Figure 6.2 and the C_{cd} of C2. There are two objects in the M_{cd} of M15, “A” and “B”, and there are also two objects in the C_{cd} of C2, “a” and “b”. The type of “A” in the M_{cd} of M15 is “window” and the type of “a” in the C_{cd} of C2 is “window”. They are exactly the same type and therefore “A” matches “a”. The type of “B” in the M_{cd} of M15 is “wall” and the type of “b” in the C_{cd} of C2 is “wall”. They are exactly the same type and therefore “B” matches “b”. All other features in the M_{cd} of M15 are exactly the same as all other features in the C_{cd} of C2 and this is illustrated in the table below.

M_{cd} of M15	C_{cd} of C2	mapping
window(A)	window(a)	same type and A matches a
wall(B)	wall(b)	same type and B matches b
attribute(A,single_glazed)	attribute(a,single_glazed)	A/a have the same attribute
attribute(B,exposed)	attribute(b,exposed)	B/b have the same attribute
part_of(A,B)	part_of(a,b)	same relation between A/a and B/b

The attribute of “A” in the M_{cd} of M15 is given as “single_glazed” and the attribute of “a” in the C_{cd} of C2, which corresponds to “A”, is also given as “single_glazed”. Therefore their attributes match. The attribute of “B” in the M_{cd} of M15 is given as “exposed” and the attribute of “b” in the C_{cd} of C2, which corresponds to “B”, is also given as “exposed”. Therefore their attributes match. The relation between “A” and “B” in the M_{cd} of M15 is exactly the same as the relation between “a” and “b” in the C_{cd} of C2.

Semantic similarity

There is one-to-one correspondence between objects in M_{cd} and objects in C_{cd}, and all features of corresponding objects are the same or similar. However there should be at least one feature in C_{cd} which is not the same as the feature of the corresponding object in M_{cd}. Otherwise, i.e., all features of corresponding objects are the same, we say that there exists literal similarity but not semantic similarity. Similarity between two features can be defined in terms of the semantic distance between these two features in specialisation hierarchies as explained in section 5.2. Semantic similarity allows non-matching features if the semantic distance between these non-matching features is defined as small as to be regarded as similar. For example, the C_{cd} of case C3 shown below and the M_{cd} of M15 are similar by the semantic similarity criterion.

C3
C_{cd}: window(a), attribute(a,single_glazed),
wall(b), attribute(b,external),
part_of(a,b)

The C_{cd} of C2 is similar to the M_{cd} of M15 by literal similarity as shown in the previous section, and the only difference between the C_{cd} of C2 and the C_{cd} of C3 is that the attribute of "b" is described as "external" instead of "exposed". Suppose that, as described in Section 5.3, a specialisation hierarchy is built from attributes of walls and the semantic distance between "exposed" and "external" is defined as close as to be regarded as similar. Then the attributes of "B" and "b" are similar and the M_{cd} of M15 and the C_{cd} of C3 are similar by semantic similarity. Mapping between all features in the M_{cd} of M15 and the C_{cd} of C3 is illustrated in the table below.

M_{cd} of M15	C_{cd} of C4	mapping
window(a)	window(a)	same type and a matches a
wall(b)	wall(b)	same type and b matches b
attribute(a,single_glazed)	attribute(a,single_glazed)	a/a have the same attribute
attribute(b,exposed)	attribute(b,external)	attributes of b/b are similar
part_of(a,b)	part_of(a,b)	same relation between a/a and b/b

If the object type of "a" is described as "opening" instead of "window" or the attribute of "b" is described as "triple_glazed" instead of "single_glazed", the C_{cd} of C3 and the M_{cd} of M15 are still similar by semantic similarity provided that "opening"/"window" and "triple_glazed"/"single_glazed" are defined as similar features. However, if the attribute of "a", "single_glazed", or the attribute of "b", "external", is not specified, the C_{cd} of C3 is similar to the M_{cd} of M15 not by literal similarity or semantic similarity but by analogy as we describe below.

Analogy

All object types match, in other words, there is one-to-one correspondence between objects in M_{cd} and objects in C_{cd} and their object types are exactly the same, and all relations between corresponding objects are also matched. Attributes of some objects in C_{cd} may be missing in M_{cd} or different from the attributes of corresponding objects in M_{cd}, but other attributes of corresponding objects are matched. A difference between two attribute features is recognised when the semantic distance between two attribute features in specialisation hierarchies (explained in Section 5.2) are too great to be regarded as similar. For example, the C_{cd} of case C4 shown below and the M_{cd} of M15 are similar by analogy.

C4
C_{cd}: window(a), attribute(a,single_glazed),
wall(b), part_of(a,b)

The C_{cd} of C2 is similar to the M_{cd} of M15 by literal similarity as shown earlier, and the only difference between the C_{cd} of C2 and the C_{cd} of C4 is that the attribute of "B" is described as "exposed" in the M_{cd} of M15 but the attribute of "b", which corresponds to "B", is missing in the C_{cd} of C4. Mapping of features between the M_{cd} of M15 and the C_{cd} of C4 is illustrated in the table below.

M_{cd} of M15	C_{cd} of C3	mapping
window(A)	window(a)	same type and A matches a
wall(B)	wall(b)	same type and B matches b
attribute(A,single_glazed)	attribute(a,single_glazed)	A/a have the same attribute
attribute(B,exposed)	-	the attribute of b is missing
part_of(A,B)	part_of(a,b)	same relation between A/a and B/b

object types	attributes	relations	similarity
M	M	M	literal similarity
M/S	M	M	semantic similarity
M	M/S	M	semantic similarity
M	M	M/S	semantic similarity
M/S	M/S	M	semantic similarity
M	M/S	M/S	semantic similarity
M/S	M	M/S	semantic similarity
M/S	M/S	M/S	semantic similarity
M	M/O	M	analogy
M/S	M/O	M	analogy + semantic similarity
M	M/S/O	M	analogy + semantic similarity
M	M/O	M/S	analogy + semantic similarity
M/S	M/S/O	M	analogy + semantic similarity
M	M/S/O	M/S	analogy + semantic similarity
M/S	M/O	M/S	analogy + semantic similarity
M/S	M/S/O	M/S	analogy + semantic similarity

Note: M means *M*atch, O means *O*mit or different, and S means *S*imilar.

Table 6.2: Simple similarity: Mapping of M_{cd} and C_{cd}

Summary

Since M_{cd} and C_{cd} are described using predicates which describe situations in terms of object types, object attributes and relations between objects, the notion of simple similarity can be defined in terms of these entities as shown in Table 6.2. In this table, three symbols are used: "M", "O" and "S".

- "M" means that entities in M_{cd} and C_{cd} match.
- "O" used only in the column of "attributes" means that there are object attributes in M_{cd} but the attributes of corresponding objects are missing or different in C_{cd}.
- "S" means that entities in M_{cd} and C_{cd} are similar.
- "M/O" means that some attributes match and other attributes are different or missing.
- "M/S" means that some entities match and other entities are similar.

- "M/S/O" means that some attributes match, some attributes are similar, and other attributes are different or missing.

For example, the first row of the table means that, if all entities in M_{cd} match entities in C_{cd}, in other words, there is 1:1 correspondence between all entities in M_{cd} and all entities in C_{cd}, then there is literal similarity between M_{cd} and C_{cd}. It is also possible that two kinds of similarity exist between M_{cd} and C_{cd}. The tenth row of the table means that M_{cd} and C_{cd} are similar by the criteria of analogy and semantic similarity as follows.

- "M/S" in the first column means that there is one-to-one correspondence between objects in M_{cd} and objects in C_{cd} and their object types are matched or similar.
- "M/O" in the second column means that some attributes of corresponding objects are the same and other attributes of corresponding objects are different or missing.
- "M" in the third column means that all relations between corresponding objects match.

6.1.3 Embedded Similarity

Embedded similarity is defined using the notion of simple similarity described in the previous section. M_{cd} and C_{cd} are said to be similar by embedded similarity if M_{cd} is similar to only part of C_{cd} according to the criteria of simple similarity. In other words, C_{cd} contains extraneous objects which do not correspond to any object in M_{cd}, their attributes, and some relations which do not match any relation in M_{cd}. Such extraneous information may be provided by the applicant in the following circumstances:

- to compensate missing information such as "percentage_area" in case C5 shown below; or
- to strengthen the applicant's argument for relaxation by providing additional information which might be relevant to the issues in debate. [1]

[1] See cases c11, c42, c77, c87, and c95 in Appendix A.

It is assumed that the M_{cd} of each model contains all features relevant to legal rules represented. Therefore, when comparing the M_{cd} of a model and the C_{cd} of a given case, if there are extraneous features in the C_{cd} which do not match any features in the M_{cd}, these extraneous features are regarded as irrelevant in determining the uniqueness of the particular circumstances to which a model should apply, and are not taken into account in the similarity test.

An example case which is similar to M15 by embedded similarity is shown below.

C5

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

In the C_{cd} of C5, 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 needed only to calculate the percentage area of the window "a". If we put aside the description about the bedroom and the floor area of the bedroom, i.e., "bedroom(r), part_of(a,r), floor_area(r,6,square_metre)", from C_{cd} then the C_{cd} of C5 is literally similar to the M_{cd} of M15 as illustrated in the table below.

M_{cd} of M15	C_{cd} of C5	mapping
window(A) wall(B) attribute(A,single_glazed) attribute(B,exposed) part_of(A,B)	window(a) wall(b) attribute(a,single_glazed) attribute(b,exposed) part_of(a,b)	same type and A matches a same type and B matches b A/a have the same attribute B/b have the same attribute same relation between A/a and B/b
-	bedroom(r) part_of(a,r) floor_area(r,6,square_metre)	extraneous features

6.1.4 Implied Similarity

Sometimes there may be objects which do not match but relations between objects imply that the situations are similar. This criterion of similarity needs to be added because of the nature of legal rules described in Section 1.2.1 and Section 2.1. Legal rules can be seen as representatives of many possible cases but it is impossible to frame in legal rules all possible situations in which a case may arise. Therefore there may be cases in which similarity between the input case and any models cannot be found by using the criteria of literal similarity, analogy and semantic similarity described in the previous two sections. As an example consider case C6.

C6
C_c: house
C_{cd}: window(a), attribute(a,double_glazed),
bedroom(b), part_of(a,b)
C_{sd}: percentage_area(a,16)

The M_{cd} of M15 is described in terms of "window" and "wall" and the C_{cd} of C6 is described in terms of "window" and "bedroom". Similarity between the C_{cd} of C6 and the M_{cd} of M15 cannot be found by the criteria of similarity described in the previous two sections. However, from the fact that the window "a" is part of the bedroom "b", it can be hypothesised that there is a wall "x" in the bedroom "b" and the window "a" is in the wall "x". Therefore, the C_{cd} of C6 can be redescribed as follows:

C6
C_{cd}: window(a), attribute(a,double_glazed),
wall(x), part_of(a,x),
bedroom(b), part_of(a,b)

Then the M_{cd} of M15 is similar to this description by the criterion of embedded analogy as shown in the table below.

M_{cd} of M15	C_{cd} of C6	mapping
window(A) wall(B) attribute(A,single_glazed) attribute(B,exposed) part_of(A,B)	window(a) wall(x) attribute(a,double_glazed) - part_of(a,x)	same type and A matches a same type and B matches x attributes of A/a are different the attribute of x is missing same relation between A/a and B/x
-	bedroom(b) part_of(a,b)	extra information

This kind of similarity requires the addition of domain knowledge to derive hypothetical information from the given description. How such domain-specific knowledge is represented and used in the current implementation is described in Section 6.4.2.

6.2 Retrieval of Relevant Models

Given a new case $C = (C_{id}, C_g, C_c, C_{cd}, C_{sd})$, the system first selects an abstraction hierarchy H_G of the given goal C_g. Then among models in H_G the system retrieves models relevant to the case in the following three steps:

1. Select models whose M_c subsumes C_c (Section 6.2.1).

2. Select models whose M_{cd} contains objects at the right abstraction level so that they are comparable to objects in C_{cd} (Section 6.2.2).

3. Select models whose M_{cd} is similar to C_{cd} (Section 6.2.3).

6.2.1 Retrieval by Category

The system compares the C_c of C and the M_c of models in H_G and selects models whose M_c subsumes the C_c of C using the specialisation hierarchies of building categories in the DKB (see Figure 5.2). M_c is said to *subsume* C_c when:

- a building category in M_c is the same as the building category in C_c; or

- a building category in M_c is at a higher level in the specialisation hierarchy than the building category in C_c.

The system starts by comparing C_c with the root model of H_G. The system goes down until the lowest level L at which M_c subsumes C_c is found. Once the system finds a model whose M_c does not subsume C_c, the system stops going down to descendant models of the current model. Only models in the path from the root model to the parent model of the current model will be marked "relevant". The system continues comparing the remaining models in H_G until there is no model to compare. The system then retrieves models marked "relevant". Note that, since the M_c of a parent model always subsumes the M_c of all child models, retrieved models again constitute a hierarchy, which is a subhierarchy of H_G. We will call this hierarchy of retrieved models H_R.

For example, suppose that the complete description of C2 is given as follows.

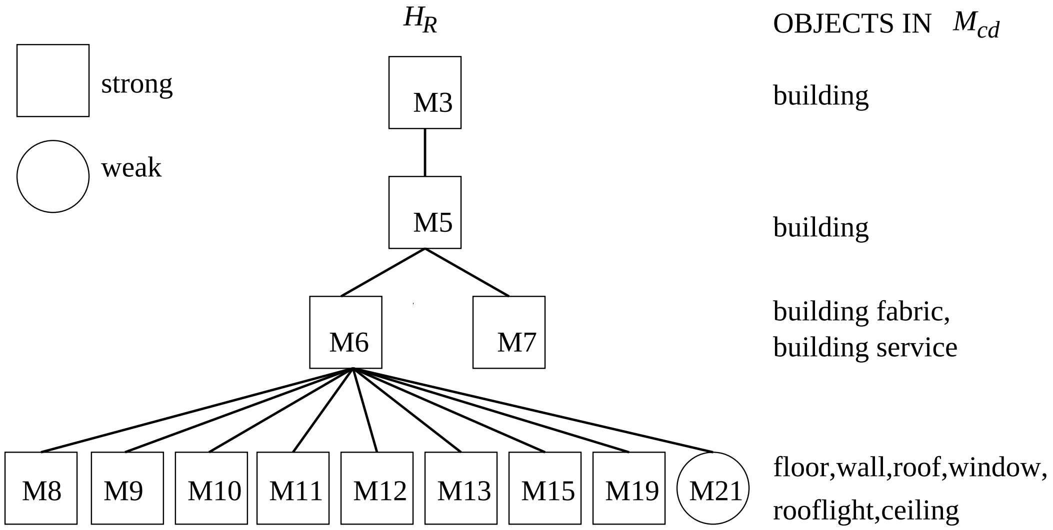

Figure 6.3: The hierarchy of retrieved models H_R

C2

C_g:	energy_conservation
C_c:	house
C_{cd}:	window(a), attribute(a,single_glazed), wall(b), attribute(b,exposed), part_of(a,b)
C_{sd}:	percentage_area(a,14.5)

C_g is given as "energy conservation" and thus the abstraction hierarchy of "energy conservation" is selected as H_G first. The C_c of C2 is given as "house". Then, based on the specialisation relations shown in Figure 5.2, models whose M_c includes "building", "dwelling" and "house" will be retrieved. If the system has the MKB shown in Figure 3.3, the system selects models H_R = {M3, M5, M6, M7, M8, M9, M10, M11, M12, M13, M15, M19, M21}, which is a subhierarchy of H_G as shown in Figure 6.3.

6.2.2 Selection of Models at the Right Abstraction Level

Before testing similarity between C_{cd} and M_{cd} of models retrieved in the previous step, the system looks for models whose M_{cd} contains objects at the *right abstraction level.* The right abstraction level means the level in the hierarchy of retrieved models H_R at which objects in M_{cd} are at the same level of the aggregation hierarchies as objects in C_{cd} so that they are comparable to each other. As described in Section 3.3, as levels go down in the abstraction hierarchy, objects in M_{cd} become more concrete and therefore the scope of application becomes more restricted. To be able to compare the details

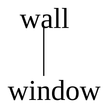

Figure 6.4: The object hierarchy of case C2

of M_{cd} and C_{cd}, objects in M_{cd} and objects in C_{cd} must be at the same abstraction level. First, the system constructs an object hierarchy of objects included in C_{cd} and gathers all leaf objects O_t in this object hierarchy.

Second, the system finds a level in H_R at which models are comparable with C_{cd}. Suppose that O_l is a set of all objects which have the same parent object with objects in the M_{cd} (s) at the level l of H_R according to the aggregation hierarchy (see Figure 5.1). For example, in Figure 6.3, there is only one object in the M_{cd} of the root model M3, "building", and "building" is the root of the aggregation hierarchy. Therefore O_1 is {building}. At the level 2, there is only one object in the M_{cd} of the model M5, "building", and "building" is the root of the aggregation hierarchy. Therefore O_2 is {building}. At the level 3, there are two objects in the M_{cd} of models M6 and M7, "building fabric" and "building service". "building" is the parent object of "building fabric" and "building service" in the aggregation hierarchy and there is another object "room" whose parent object is "building". Therefore, O_3 is {building fabric, building service, room}. Similarly, O_4 is {floor, wall, roof, window, rooflight, ceiling}. The system first finds O_1 and see whether O_1 includes O_t. The system repeats this test for each level of H_R from the top level to the bottom level until O_l which includes O_t is found.

For example, in C2 described in the previous section, C_{cd} is described in terms of "window" and "wall". According to the aggregation hierarchy, "window" is part of "wall" and the aggregation hierarchy for those objects is illustrated in Figure 6.4. There is only one leaf object "window" in this case, therefore O_t is {window}. Once O_t is found, the system first compares O_t = {window} with objects in the M_{cd} of the root model in H_R shown in Figure 6.3. O_1 is {building}, and it does not include O_t. The system goes down until the system reaches the fourth level at which M_{cd}s are described in terms of "floor", "wall", "roof", "window" and "rooflight". From the aggregation hierarchy shown in Figure 5.1, the system obtains O_4 = {floor, wall, roof, window, rooflight, ceiling}, and it includes O_t. Therefore the models at the fourth level, H_{R_4} = {M8, M9, M10, M11, M12, M13, M15, M19, M21} are retrieved for the next step.

6.2.3 Selection of Models by Similarity of Circumstances

Once the right abstraction level is found, the system compares the details of C_{cd} and each M_{cd} of models retrieved so far and selects models whose M_{cd} is *similar* to C_{cd} of the given case according to similarity criteria described in Section 6.1. First, the M_{cd} of each retrieved model is compared to C_{cd} to see whether there is a model (or models) whose M_{cd} is similar to C_{cd} by simple similarity and embedded similarity. If a model (or models) whose M_{cd} is similar to C_{cd} by either simple similarity or embedded similarity is not found, the system then tries to find a model (or models) whose M_{cd} is similar to C_{cd} of the given case by implied similarity.

Simple similarity test

Among retrieved models, only models whose M_{cd} have exactly the same number of object type predicates and relation predicates as C_{cd} are tested for simple similarity and this is carried out as follows.

1. All objects in M_{cd} and all objects in C_{cd} are compared to see whether there is object similarity, i.e., whether all objects in M_{cd} match or are similar to all objects in C_{cd}.

2. If there is object similarity and therefore there is a correspondence between objects in M_{cd} and C_{cd}, then attribute predicates and relation predicates of corresponding objects are compared to see whether there is attribute similarity and relation similarity.

3. If there is no object similarity, there is no simple similarity and attribute predicates and relations predicates are not compared.

Details of M_{cd} and C_{cd} are compared in terms of object type predicates (object similarity), attribute predicates (attribute similarity) and relation predicates (relation similarity). The similarity between two object types, attributes and relations can be measured in terms of semantic distance in specialisation hierarchies of these concepts described in Section 5.2. We use three matching indicators to indicate three kinds of similarity: an object matching indicator (I_O) for object similarity, an attribute matching indicator (I_A) for attribute similarity and a relation matching indicator (I_R) for relation similarity. Symbols used in Table 6.2 are used as values to be assigned to the matching indicators. If there is no match, the indicator will be set as "F", which means "Fail to find similarity". Given three indicators as the result of object similarity, attribute similarity and relation similarity, what kind of similarity exists can be found out by looking up the table shown in Table 6.2. Note that object similarity takes precedence in the simple similarity test over attribute and relation similarity.

Embedded similarity test

If the system fails to find a model (or models) whose M_{cd} is similar to C_{cd} by applying the criteria of simple similarity, embedded similarity is tested for the M_{cd} of each retrieved model. Embedded similarity is tested only for models:

- whose M_{cd} have the same number of object type predicates as or less object type predicates than C_{cd}; and
- whose M_{cd} have less relation predicates than C_{cd}.

If the M_{cd} of a model satisfies the conditions described above, embedded similarity is tested as follows.

1. Find all possible subsets of C_{cd} which are comparable with M_{cd}, i.e., subsets of C_{cd} in which the number of object predicates is equal to that of object predicates in M_{cd} and all attribute predicates and relation predicates of those objects are included.
2. For each subset of C_{cd}, do the simple similarity test.

Implied similarity test

If a model (or models) whose M_{cd} is similar to C_{cd} by embedded similarity is not found, the system then tries to find model(s) whose M_{cd} is similar to C_{cd} of the given case by implied similarity. The system transforms C_{cd} into alternative description using knowledge provided in the DKB and repeats the simple and embedded similarity test with this alternative description. The transformation of C_{cd} is carried out in three steps:

1. Introduction of new objects and part_of relations:
   Objects which exist in M_{cd} but not in C_{cd} are added and "part_of" relations between existing and new objects are also added. For example, in C6 shown in 6.1.4 to explain implied similarity, A wall "x" which exist in the M_{cd} of M15 but not in the C_{cd} of C6 is added to the C_{cd} and a new relation between the existing window "a" and the new wall "x", "a is part of x", is added as well.

2. Default property generation:
   Default properties are added 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 the number of storeys in the building is not specified, then "1" is added as the number of storeys.

3. Relation inference:
   Relations between objects in C_{cd} are inferred and added 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}.

Each step uses domain-specific heuristic rules, examples of which are given in Section 6.4.2.

6.3 Case Evaluation

To determine the conformance of the given case, the system compares the proposed solution C_{sd} of the given case C with the solution M_{sd} of each retrieved model M. If the solution described in C_{sd} is within the range of the solution M_{sd}, C is said to satisfy M. For example, in the previous section, M15 is selected as relevant to C2. C_{sd} of C2 and M_{sd} of M15 are as follows.

C2
C_{sd}: percentage_area(a,14.5)
M15
M_{sd}: percentage_area(A,C), C =< 15

"a" in C_{sd} matches "A" in M_{sd} and "14.5" in C_{sd} matches "C" in M_{sd}. "C" must be equal to or less than "15" and "14.5" satisfies this requirement. Therefore, C2 is said to satisfy M15.

Here is another example C5 from the previous section. M15 is retrieved as relevant to this case.

C5
C_{cd}: window(a), attribute(a,single_glazed),
wall(b), attribute(b,exposed), part_of(a,b),
bedroom(r), part_of(a,r),
floor_area(r,6,square_metre)
C_{sd}: height(a,1,metre), width(a,0.6,metre)

However, it is impossible to compare C_{sd} of C5 with M_{sd} of M15 directly as in C2. "a" in C_{sd} matches "A" in M_{sd}. M15 specifies the requirement in terms of "percentage_area" of windows and a value to be compared to "C" must be calculated from information available in C5. Knowledge about how to calculate the percentage area of window will be stored in the DKB as heuristic rules (described in detail in the next section). The system use these rules to calculate the value required for the evaluation of the case. The percentage area of the window is defined as 100×(the size of the window)/(the area of the floor). Therefore, the percentage area of the window "a" will be 100×(1×0.6)/6 = 10. "10" is compared with "C" in M_{sd} and C5 satisfies M15.

object types	attributes	relations	similarity
M	M	M	literal similarity
M	M/O	M	analogy
M/S	M	M	semantic similarity
M	M/S	M	semantic similarity
M/S	M/S	M	semantic similarity
M/S	M/O	M	analogy + semantic similarity
M	M/S/O	M	analogy + semantic similarity
M/S	M/S/O	M	analogy + semantic similarity

Note: M means *M*atch, O means *O*mit or different, and S means *S*imilar.

Table 6.3: Simple similarity in the prototype system: Mapping of M_{cd} and C_{cd}

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. Cases attached to these retrieved models are cases similar to the current case and there is no need to compare all cases in the Case Library with the current case to find similar cases.

6.4 Implementation and Discussion

SICStus Prolog Version 2.1.9 running on Sun Sparc workstations was used as an implementation language. The prototype system was developed in two phases. First, the system was designed with the six imaginary cases (C2–C7) mentioned throughout this thesis. Then the prototype system was implemented and evaluated with 22 appeal cases (supplied by The Scottish Office) shown in Appendix A. This section describes limitations in the current implementation of the retrieval procedure and discusses issues raised during the implementation.

6.4.1 Implementation of Semantic Similarity

In the current implementation, similarity only between object types and attributes is defined as described in Section 5.3. No specialisation hierarchy was built from the vocabulary used to represent relationships between objects. Therefore relation similarity was not taken into account in the current implementation. Table 6.2 was modified to accommodate this situation and Table 6.3 is used in the prototype system.

The other aspect worth discussing is that the maximum semantic distance allowed in order for two concepts to be considered as similar is fixed as described in Section 5.3. However, if the maximum semantic distance is increased more models would be retrieved as relevant, and if the maximum semantic distance is decreased less models would be retrieved as relevant. Therefore it might be possible (and useful from the user's point of view) to allow the user to play with the semantic distance during the retrieval so that the user can have control over which models to be retrieved as relevant, in other words, the scope of the legal rules to which a case should apply.

6.4.2 Implementation of the Implied Similarity Test

When the system fails to find a model (or models) whose M_{cd} is similar to the C_{cd} of the given case by applying the criterion of simple and embedded similarity, C_{cd} is transformed into an alternative description in three steps: first introduce new objects and part_of relations, second generate default properties, and third infer new relations. In the implementation of the first step, it is supposed that each case has a local design database which contains additional features about the building in question. Then the following information in the local design database are extracted and added to C_{cd}:

- Predicates which represent object types which exist in M_{cd} but not in C_{cd}; and
- Predicates which represent "part_of" relations between objects already in C_{cd} and objects extracted from the local design database.

Although, local design databases are represented as Prolog predicates in the current implementation, [2] it is imaginable to link a CAD system to the system and extract information from the design database in this design system. The integration with CAD systems is further discussed in Chapter 9 as a possible extension of the system.

In the second and third steps, domain specific heuristic rules are used to transform the C_{cd} of a given case. The following three rules are used to add default properties if properties of objects in C_{cd} are missing:

[2] Local design databases are represented using a Prolog Predicate additional_fact/2. The specification of additional_fact/2 and examples can be found in Appendix C.

D1: If the number of a building component is not given
Then add the number of that building component as 1
D2: If the number of storeys in the building is not given
Then add the number of storeys as 1
D3: If the occupancy capacity of a storage room is not given
Then add the default occupancy capacity as 0

The following ten rules are used to infer new relations between objects in C_{cd}:

H1: If a window is in the bedroom and
the height of the window is H metres and
the width of the window is W metres and
the area of the floor is X square metres
Then the percentage area of the window in the bedroom
is 100×H×W/F
H2: If the area of the window is X square metres and
the area of the floor is Y square metres
Then the percentage area of the window is 100×X/Y
H3: If an object A is part of an object B and
B is part of another object C
Then add a new relation "A is part of C"
H4: If an object A is connected to an object B and
B is connected to another object C
Then add a new relation "A is connected to C"
H5: If an object A is connected to an object B
Then an object B is connected to an object A
H6: If there is an access to the building and
there is a storey at the ground level
Then the access to the building is part of that storey
H7: If a storey is part of a building
Then the area of that storey is the same as the area of the building
H8: If there is a storey at the first level
Then the height above the ground of that storey is
not less than 2.5 metres
H9: If there is a storey at the second level
Then the height above the ground of that storey is
not less than 5 metres
H10: If there is a storey at the third level
Then the height above the ground of that storey is
not less than 7.5 metres

The system uses a forward chaining method to derive hypothetical relations, i.e., if all information described in the antecedent part of a rule (If part) is known, then it is assumed that relations described in the consequent part of the rule (Then part) hold. Ten rules used in "Relation inference" are also used in the case evaluation to derive knowledge which is not given explicitly in C_{sd}. These rules are not always intended to be true (for example, H6–H10 cannot be true in all cases) but are used to bring out possible similarities under plausible default assumptions in the building regulations. Heuristic rules used to transform C_{cd} and C_{sd} must be devised by employing domain-dependent knowledge, which limits the generality of the system. More heuristic rules might be needed as the number of cases grow. The use of domain-specific knowledge will be discussed further in Section 8.6.

6.4.3 Implementation of the Case Evaluation

The conformance of a case with a retrieved model is decided by checking whether features given in the case satisfy all requirements specified in the M_{sd} of the retrieved model. In order to make use of heuristic rules described in the previous section, the prototype system uses a backward chaining method. First, all features in the C_{cd} and C_{sd} are saved as known facts. Second, all objects represented as variables in the M_{sd} are replaced with corresponding object names in the C_{cd} or C_{sd} and these instantiated predicates are set as goal predicates. With predicates saved as facts and heuristic rules backward chaining is performed for each goal predicate. Although variables in goal predicates which represent objects were instantiated, there may still be variables which represent property values. If there is such a variable in goal predicates, a simple unification algorithm is used to find a unifying constant from the predicates saved as known facts. [3]

[3] We use a very simple unification algorithm used in Prolog. See p. 71 in [Sterling & Shapiro 86] for further details.

Chapter 7

Acquisition of New Regulatory Information

The previous chapter described how relevant regulatory information, i.e., relevant models, is retrieved and how the proposed solution C_{sd} of the case is evaluated against the retrieved models. When the proposed solution does not satisfy any "strong" models retrieved as relevant, there is a possibility of relaxation. The decision of whether relaxation is to be granted will be made by a human expert (i.e., the user of the system). The system provides relevant information stored within the system which might be useful in making such a decision. Once relaxation is granted, new legal rules are acquired by integrating rulings made in the relaxed case into the existing MKB. The content of this chapter is as follows:

- Section 7.1 explains what regulatory information is to be acquired and what relevant information to be explained to the user.
- Section 7.2 describes how relevant information is explained.
- Section 7.3 describes how legal rules set out in relaxed cases are acquired.
- Section 7.4 describes how these are implemented and discusses issues raised during implementation.

7.1 Acquiring New Regulatory Information

For a case to be valid for being integrated into the MKB, there are two conditions to be satisfied.

1. The relaxation of regulations should to be granted.

Only cases in which relaxation is granted are integrated into the MKB. The decision on relaxation is often made under the influence of social and political rather than technical considerations. It might be possible to build a computational model of the decision process but that is beyond the scope of this thesis. Therefore, the decision on whether relaxation of regulations relevant to the case will be made by the user. The expert makes a decision on relaxation after consulting all information available as described in Section 2.1.2. The system helps the user to make the decision by retrieving and explaining relevant information stored in the knowledge bases. Information which the system can provide to the user is as follows.

- Principles or issues behind relevant regulations:

  Models in the MKB represent legal rules from statutory regulations and legal rules established in relaxed cases at various abstraction levels. Legal rules represented at the higher levels are more descriptive and explain the general principles or issues behind more prescriptive legal rules represented at the lower levels. Suppose that M_n is a model at level n. Then models in the path from the root model M_1 to the model M_{n-1} provide the general principles or issues behind requirements represented as M_n and can be used to explain reasons why M_n should be satisfied. The general principles or issues behind regulations relevant to the case are represented as models in the path from the root model to the retrieved strong model(s), and the system explains them to the user.

- Previous similar cases:

  More context-sensitive issues behind particular regulations can be identified by examining how decisions were made: on what grounds and in what circumstances the same regulations were applied in previous cases. Such points are made explicit during the argumentation process and recorded in the Case Library using the argumentation structure described in Section 4.3. Cases attached to models retrieved as relevant are previous cases similar to the given case and the system explains them to the user.

2. All requirements described using the same criterion with respect to other subjects or objectives must be checked to ensure that there is no conflict.

This test is needed because building regulations deal with various issues and it is very rare for any part of a building to serve only one purpose. It is possible to give different standards to one building component to achieve different objectives. An example is the requirement on the size of windows. The building regulations specify the maximum size of the window for the conservation of energy and the minimum size of the window for ventilation. When the applicant seeks the relaxation of the maximum size of the window under the objective of the conservation of energy and a window which is smaller than the required maximum size is allowed to be constructed, the window should not be smaller than the minimum size of the window required for ventilation. Therefore, when a criterion is evaluated for requirements in one goal, all requirements described using the same criterion with respect to other objectives must be checked to ensure that there is no conflict. [1]

7.2 Explanation of Relevant Information

Relevant information to be explained includes:

- principles or issues behind relevant regulations, i.e., models at the upper levels of retrieved strong models which provide the reason why requirements represented as the retrieved strong models should be satisfied; and
- previous similar cases, i.e., cases attached to retrieved models.

[1] From the law's point of view, this condition is to maintain among legal rules represented in the MKB some degree of "coherence" in McCormick's terminology [McCormick 78].

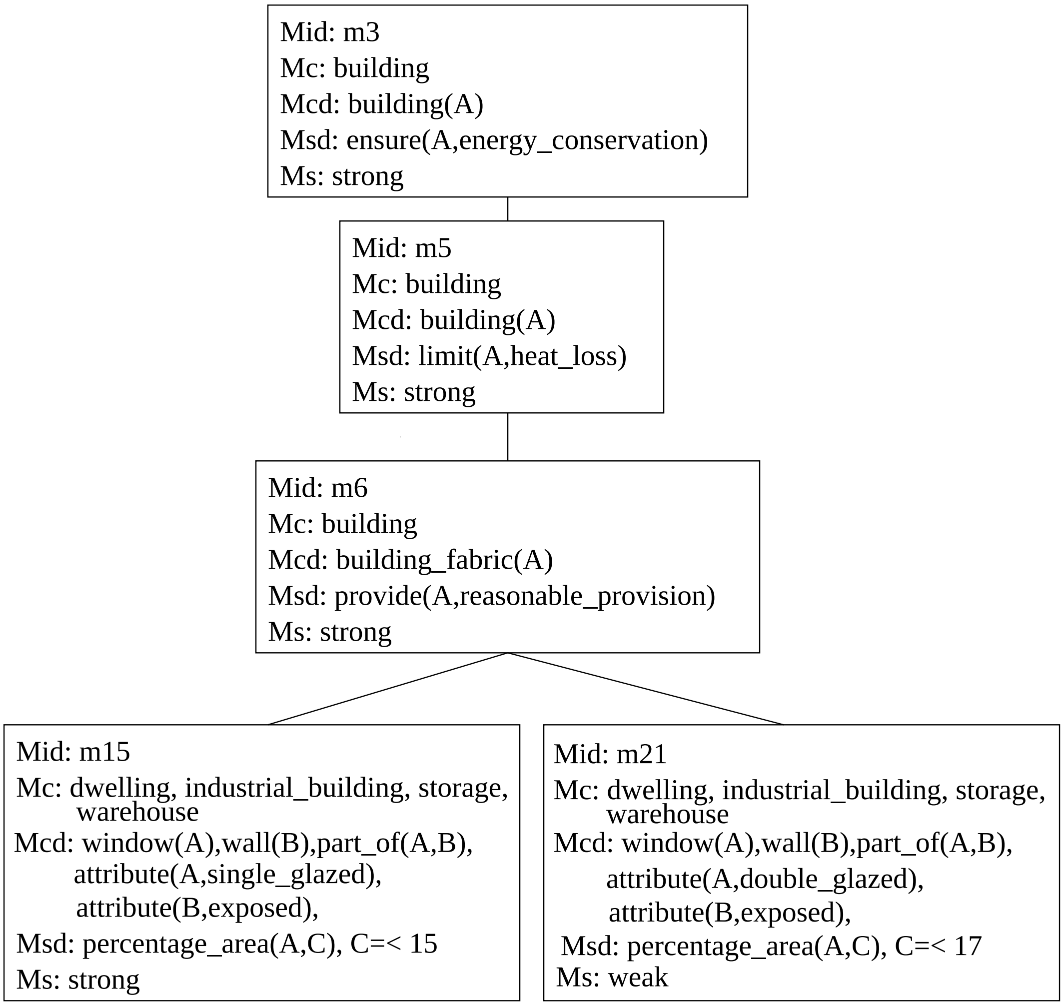

Figure 7.1: Models related to C7

7.2.1 Explanation of Principles

Models at the upper levels of retrieved strong models are explained to the user as principles behind requirements retrieved as relevant. For example, suppose that a case C7 is given as follows.

C7
C_g: energy_conservation
C_c: house
C_{cd}: window(a), attribute(a,double_glazed),
wall(b), attribute(b,exposed), part_of(a,b),
C_{sd}: percentage_area(a,18)

Then two models are retrieved as relevant to C7: a strong model M15 and a weak model M21 (see Figure 7.1). M15 requires that the percentage area of the window "a", which corresponds to "A" in M15, be less than or equal to 15. In C7 the percentage area of the window "a" is proposed as 18, which is greater than the 15 required by M15. Thus C7 does not satisfy M15 and models at the upper levels of M15 are explained to the user. As shown in Figure 7.1, M15 is at the fourth level of the abstraction hierarchy of "energy conservation", and there are three models at the upper levels of M15: M3, M5 and M6. M3 represents a principle that all buildings should be built to ensure the conservation of energy. This principle is translated into the more concrete principle represented in M5: all buildings should be built to limit heat loss. And this principle represented in M5 is again translated into the more concrete principle represented in M6: reasonable provision should be made for the building fabric. M15 provides one of many ways to achieve this principle in terms of the percentage area of windows. Predicates used to represent these principles (or general requirements) are converted into English-like texts and displayed to the user as follows:

the building fabric provides reasonable provision
therefore
the building limits heat loss
therefore
the building ensures energy conservation

7.2.2 Explanation of Previous Similar Cases

Cases attached to retrieved models are previous cases similar to the given case. Cases stored in the Case Library are represented as: $C = (C_{id}, C_g, C_c, C_{cd}, C_{sd}, C_d, C_m, C_t)$ where $(C_{id}, C_g, C_c, C_{cd}, C_{sd})$ is the same as the corresponding parts in the case description provided as input, C_d is the decision about relaxation taken by the BD, i.e., "granted" or "refused", C_m contains the id numbers of models which were retrieved as relevant to this case, or "nil" if there was no relevant model, and (in the current implementation) C_t contains the id number of a strong model (or strong models) which the case was supposed to satisfy but did not satisfy. Cases attached to retrieved models can be found by retrieving all cases from the Case Library whose C_m is one of models retrieved as relevant. Predicates in the C_{cd} and C_{sd} of such cases are translated into English-like sentences. The id number of models in C_t is replaced with the article number of the regulations. For example, suppose that, after processed by the system, case C7 shown in the previous section is stored in the Case Library apart from the argumentation as follows:

C7
C_g: energy_conservation
C_c: house
C_{cd}: window(a), attribute(a,double_glazed),
wall(b), attribute(b,exposed), part_of(a,b),
C_{sd}: percentage_area(a,18)
C_d: granted
C_m: m15, m21
C_t: m15

Then C7 is explained as follows:

CASE: c7
SUBJECT: energy conservation
BUILDING CATEGORY: house
IN THE FOLLOWING CIRCUMSTANCES:
the window is double-glazed
the wall is exposed
the window is part of the wall
PROPOSED SOLUTION:
the percentage area of the window is 18
DECISION MADE:
The case was not compliant with J2.3.
Relaxation was granted.

Context-sensitive issues debated in a case are recorded in the Case Library using an argumentation network described in Section 4.3.4. The argumentation network described in Section 4.3.4 can be easily converted into English-like sentences by following the links in the argumentation network. For each position, first, the position itself is explained to the user. This is done by translating predicates which represent the position into English-like sentences. Second, supporting arguments and opposing arguments are explained by following "support" and "oppose" links between the position and the arguments linked to this position. Reasons for supporting or opposing the position are attached to support/oppose links. According to the sort of attachements, three canned phrases are used to explain each argument.

- If the reason is "full compliance", "partial compliance" or "non-compliance", the following canned phrase is used:

[player] argued that:
the proposal constitutes [reason] based on the facts that
[argument]

- If the reason is "reasonable", "unreasonable", "practical" or "impracticable", the following canned phrase is used:

  [player] argued that:
  the proposal is a [reason] solution when
  [argument]

- If the reason is given as predicates, the following canned phrase is used:

  [player] argued that:
  under the circumstances in which
  [argument]
  the proposal has the following consequences
  [reason]

Predicates which represent arguments and reasons are translated into English-like sentences, and inserted into the phrase. For example, the argumentation surrounding the case c93 shown in Figure 4.3 is explained as follows.

The following proposal suggested by the applicant was granted:
the top height above the floor of the window is 1.8 metres

The proposal was supported by the following arguments:
the applicant argued that:
the proposal constitutes partial compliance based on the facts that
the window is part of the building
the building is an eighteenth century house
the applicant argued that:
under the circumstances in which
there are 3 windows
the proposal has the following consequences
the window provides adequate supply of air

The proposal was opposed by the following arguments:
the Local Building Control Authority argued that:
the proposal constitutes non-compliance based on the facts that
the window is new

The other proposal suggested was:
the Local Building Control Authority suggested that:
the top height above the floor of the window is not less than 2 metres

The proposal was supported by the following arguments:
the Local Building Control Authority argued that:

the proposal constitutes full compliance based on the facts that
the window is new

The proposal was opposed by the following arguments:
the applicant argued that:
the proposal is an impracticable solution when
the window is part of the building
the building is an eighteenth century house

7.3 Acquisition

After relevant information—principles or issues behind the relevant regulations and similar cases—is explained, the system checks whether or not there is conflict between the proposed solution C_{sd} in a given case C and existing models in the abstraction hierarchies whose goal is different from C_g, and informs the expert. Then the human expert is asked to make a decision on relaxation. If it is decided to grant relaxation, the new rulings established in the relaxed case C are acquired by integrating the relaxed case C into the abstraction hierarchy of C_g. This section first describes the conflict checking and then the integration of a relaxed case into the MKB.

7.3.1 Conflict Checking

To identify conflicts, among models in the abstraction hierarchies whose goal is different from C_g, the system looks for strong models whose:

- M_c subsumes the C_c of the relaxed case; and
- M_{sd} is provided in terms of the same criteria as the C_{sd} of the relaxed case.

If there exist such models, C_{sd} is evaluated against the M_{sd} of those models. For example, when the case C7 shown in the previous section is given, M15 is retrieved as relevant by the analogy criterion but the solution proposed in this case does not satisfy M15. In order for C7 to be added to the MKB, there must be no conflict with existing strong models. In C7 the C_{sd} is described in terms of "the percentage area" of the window "a". There are two other requirements specified in terms of "the percentage area" of windows other than the requirements represented in M15 [Sco90]:

- Under the subject of "convenience", every apartment must have a window of a glazed area equal to at least 1/15th of the floor area of the apartment and situated in an external wall.

- Under the subject of "health", in order to provide an adequate supply of air for users of the building, a room, other than a kitchen, must have provision for natural ventilation which provides a ventilator with an opening area of not less than 1/30th of the floor area.

Suppose that these two models are represented as M31 and M43 as part of the MKB shown Figure 3.3: M31 in the abstraction hierarchy of "convenience" and M43 in the abstraction hierarchy of "health".

M31

M_c:	dwelling
M_{cd}:	apartment(A), wall(B), type(external), window(C), part_of(C,A), part_of(C,B)
M_{sd}:	percentage_area(C,D), D >= 6.67
M_s:	strong

M43

M_c:	dwelling
M_{cd}:	room(A), not kitchen(A), floor(B), part_of(B,A), area(B,C,square_metre), C >= 4, ventilator(D), part_of(D,A)
M_{sd}:	percentage_area(D,E), E >= 3.33
M_s:	strong

The M_c of these two models subsume the C_c of C7 and they describe their required solution M_{sd} in terms of the same criterion as the C_{sd} of C7:

- M31 describes the requirement in terms of the "percentage_area" of windows; and

- M43 describes the requirement in terms of the "percentage_area" of ventilators, which are similar to windows because window is a kind of ventilator.

C7 does not conflict with the requirements from M31 and M43. Therefore, C7 is valid for acquisition.

7.3.2 Integration of Relaxed Cases into the MKB

Given all relevant information, i.e., principles or issues behind the relevant regulations, similar cases and whether or not there is conflict with existing legal rules, the human expert is asked to make a decision on relaxation. If the expert decides that the relaxation of a specific regulation relevant to the case should be granted, the case C is integrated into the existing abstraction hierarchy of the given goal C_g in the MKB. The M_c, M_{cd} and M_{sd} of the relevant weak models and the C_c, C_{cd} and C_{sd} of the relaxed case are combined to create a weak model which represents a more general requirement. Then this new weak model replaces the weak models which are combined with the relaxed case. If there is no "weak" model relevant to the relaxed case or the relaxed case cannot be combined with existing relevant weak models, a new "weak" model will be created from the case. The M_c, M_{cd} and M_{sd} of the relevant weak models and the C_c, C_{cd} and C_{sd} of the relaxed case are combined as follows.

1. Guess a new M_c.

The system first guesses the category of buildings to which the requirements set out in the relaxed case should apply. The system guesses the category by looking at "strong" models which are at the same level as the relevant strong models and have the same parent as the relevant strong models. Among those models, the system looks for strong models:

- whose M_c subsumes the C_c of the relaxed case;
- whose M_{cd} is similar to the C_{cd} of the relaxed case; and
- whose M_{sd} is given in terms of the same criteria as the C_{sd} of the relaxed case.

If there are such models, then building categories in the M_cs of selected models are generalised to obtain a new M_c which subsumes all the M_cs of selected models. If there is no such model, then the C_c of the relaxed case will be the new M_c.

2. Generalise the M_{cd}s of similar "weak" sibling models and the C_{cd} of the relaxed case to get a new M_{cd}.

The system looks at "weak" models which are at the same level as the relevant strong models and have the same parent as the relevant strong models. Among those models, the system looks for weak models which represent requirements in the same context as the relaxed case. In other words, the system selects models:

- whose M_c subsumes the C_c of the relaxed case;
- whose M_{cd} is *simply similar* to the C_{cd} of the relaxed case; and
- whose M_{sd} is given in terms of the same criteria as the C_{sd} of the relaxed case.

If there are such models, the M_{cd}s of the selected models and the C_{cd} of the relaxed case are generalised into a new M_{cd}. If there is no such model, the C_{cd} of the relaxed case will be the new M_{cd}.

3. Generalise the M_{sd}s of similar "weak" sibling models and the C_{sd} of the relaxed case to get a new M_{sd}.

The M_{sd}s of weak models selected in the previous step and the C_{sd} of the relaxed case are generalised. If there is no such model, the system looks at "strong" models which are at the same level as the relevant strong models and have the same parent as the relevant strong models. Among those models, the system looks for "strong" models which represent requirements in the same context as the relaxed case. In other words, the system selects strong models:

- whose M_c subsumes the C_c of the relaxed case;
- whose M_{cd} is similar to the C_{cd} of the relaxed case; and
- whose M_{sd} is given in terms of the same criteria as the C_{sd} of the relaxed case.

If there are such models, the M_{sd}s of the selected models and the C_{sd} of the relaxed case are generalised into a new M_{sd}. If there is no such model, a new M_{sd} will be created from the pattern of a criterion used to describe the M_{sd}s of selected strong models.

Note that, in step 2 and 3, only strong models whose M_{cd} is *simply similar* to the C_{cd} of the relaxed case are selected for generalisation. This is due to the limitations in the implementation of generalisation discussed in the next section.

A new weak model which consists of these new M_c, M_{cd} and M_{sd} replaces the weak models which are combined with the relaxed case. If there is no "weak" model relevant to the relaxed case or the relaxed case cannot be combined with the existing weak models, a "weak" model is added.

For example, in Section 7.3.1, C7 was decided to be valid for acquisition. M15 was the strong model that C7 was supposed to satisfy. Among strong models which are at the same level as M15 and have the same parent as M15, M15 is the only strong model which represents requirements in the same context as C7, i.e., the M_c of M15 ("dwelling", "industrial building", "storage" and "warehouse") subsumes the C_c of C7 ("house"), the M_{cd} is similar to the C_{cd} of C7 (by the analogy criterion), and the M_{sd} is described in terms of the same criteria as the C_{sd} of C7 ("percentage area"). Therefore, the M_c of M15 will be the M_c of the new model.

Then, among "weak" models which are at the same level as M15 and have the same parent as M15, the system looks for "weak" models which represent requirements in the same context as the relaxed case. From the MKB shown in Figure 3.3, there is only one "weak" model which meets those conditions: M21.

M21

M_c:	dwelling, industrial_building, storage, warehouse
M_{cd}:	window(A), attribute(A,double_glazed), wall(B), attribute(B,exposed), part_of(A,B)
M_{sd}:	percentage_area(A,C), C =< 17
M_s:	weak

The M_{cd} of M21 is literally similar as the C_{cd} of C7 and thus the M_{cd} of M21 will be the new M_{cd}. The solution in M21 is given in terms of the *maximum* percentage area of window. In C7, "18" percent of the window area, which is greater than the maximum percentage area of windows prescribed in M21, is permitted.

M21
M_{sd}: percentage_area(A,C), C =< 17
C7
C_{sd}: percentage_area(a,18)

Therefore the M_{sd}s from M21 and C7 are generalised as "the percentage area of double-glazed windows in walls should not exceed 18 percent of the total floor area". The new model M22 is as follows and M21 replaces M22.

M22
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 =< 18
M_s: weak

7.4 Implementation and Discussion

7.4.1 Implementation of the Explanation Generation

Section 7.2 described how principles behind relevant regulations and similar cases are explained. Models which represent relevant principles and the formal descriptions of similar cases are transformed into English-like sentences using transformation rules which map each predicate into an English-like sentence. English-like sentences produced from predicates in this way look different from the original text which is formalised into the predicates, and might be read differently from the original text. The only way to remove such uncertainty is to show the source materials directly to the user. A hypertext (or hypermedia) system [Conklin 88, Nielsen 90] can be linked to the system to provide the user with source materials as in [Dayal *et al.* 93, Soper & Bench-Capon 94]. A hypertext system is a collection of documents, each of which can be an article in statute, a letter, a drawing, a research paper, a case report, etc. There are two ways of integrating a legal knowledge-based system and a hypertext system, as Soper and Bench-Capon suggested in [Soper & Bench-Capon 94]. One way is to loosely couple two systems. A hypertext system contains its own hypertexted version of documents. A knowledge-based system contains its own logical representation of information from the documents. It assumes that the hypertext manager and the knowledge-based system communicate by message passing, and two systems can be switched to each other whenever necessary. ASSESS [Dayal *et al.* 93] is one such system. ASSESS is a rule-based decision support system for the New Zealand Accident Rehabilitation and Compensation Insurance Corporation and uses a hypertext component as an explanation facility. The other way is to couple two systems more tightly. The knowledge-based system provides a conceptual model of legal information and this model is also used to structure documents in the hypertext document base. Link structures provided by the knowledge-based system support more directed and systematic traversal and retrieval of information stored in the hypertext document base. Sober and Bench-Capon also proposed to use hypertext as a top-level interface to a knowledge-based system. Animated hypertext [Soper & Bench-Capon 94] and PLINTH [Casson 93] are of this type. Both provide a semantic network-based model of documents and use If-Then type rules to support the traversal and retrieval of information. Our model of regulatory information, i.e., abstraction hierarchies of models and cases attached to relevant models, provides a model for structuring documents, and the case-based reasoning method described in this and previous chapters can be employed beneath a hypertext system of regulatory documents to support the retrieval of relevant information and the guided browsing of information.

7.4.2 Implementation of the Integration of a Relaxed Case

In the generalisation of circumstances and solution descriptions described in Section 7.3.2, only models whose M_{cd} is *simply similar* to the C_{cd} of the relaxed case are selected for generalisation and this is due to the limitation in the generalisation procedure. In the current implementation, two methods are used in generalisation: parameter adjustment and type generalisation.

Parameter adjustment is carried out when properties are given as numbers. One example of parameter generalisation is shown in Section 7.3.2. The M_{sd} of M21 and the C_{sd} of C7 are specified in terms of the percentage area of windows. In the M_{sd} of M21, the percentage area of windows is given as a range of numbers. In the C_{sd} of C7, the percentage area of windows is given as a specific number.

M21
M_{sd}: percentage_area(A,C), C =< 17
C7
C_{sd}: percentage_area(a,18)

Parameter adjustment is always guided by the strong model retrieved as relevant. In the M_{sd} of M21, the values of the percentage area of window is specified as the *maximum* percentage area of window, and adjustment is done by following this pattern. "18" percent of the window area is permitted and the maximum percentage area of window is extended to include this value as follows.

New M_{sd}: percentage_area(A,C), C =< 18

Type generalisation is carried out when object types or object attributes are given non-numerically and they are similar. When object types or object attributes are the same, there is no need for generalisation. When they are similar, generalisation is done by climbing a specialisation hierarchy of given types. Here is an example. Suppose that the attribute of the wall in C7 is given as "external".

M21
In M_{cd}: wall(B), attribute(B,exposed)
C7
In C_{cd}: wall(b), attribute(b,external)

The attribute "exposed" is a subclass of the attribute "external" as shown in Figure 5.4. Therefore the attribute of a wall "B" will become "external" in the new M_{cd}.

In new M_{cd}: wall(B), attribute(B,external)

Chapter 8

Evaluation: Experimental Results

We described a unified case-based model of statutory regulations and cases (see Chapter 2–3) and a case-based reasoning system which employs this model to access and maintain statutory regulations and cases (see Chapters 3–7). The system was designed with 6 imaginary cases (C2–C7), and then a prototype system was implemented and evaluated using real-world cases, i.e., 22 appeal cases provided by The Scottish Office. This chapter presents the results observed during the evaluation of the prototype system. The contents of this chapter is as follows:

- Section 8.1 describes the environments built for the evaluation of the prototype system.
- Section 8.2 shows how the prototype system works using a case.
- Section 8.3–8.5 describe and discuss the evaluation results in detail. Particularly in Section 8.4, we describe a series of experiments carried out to see how flexible the similarity assessment (described in Section 6.1–6.2) is.
- Section 8.6 provides the summary of this chapter and discusses issues related to scale-up problems.

8.1 The Environments for the Evaluation

Test cases

22 appeal cases provided by The Scottish Office were used in the evaluation of the prototype system: case 11, 41, 42, 77, 82, 87, 90, 93, 95, 97, 110, 116, 131, 132, 134, 141, 150, 154, 156, 165, 172 and 0012. The summaries of these 22 appeal cases are listed in Appendix A. There are 13 cases in the subject area "safety": case 11, 41, 77, 82, 90, 95, 97, 110, 131, 154, 156, 165 and 0012; 5 cases in the subject area "health": case 42, 87, 93, 134 and 141; and 4 cases in the subject area "convenience": case 116, 132, 150 and 172. There are three building components in question in case 90 and case 90 was divided into three sub-cases: case 90_1 for the stair in the stairway, case 90_2 for the division wall in the stairway and case 90_3 for the floor in the stairway. There are two floors in question in case 154 and case 154 was also divided into two sub-cases: case 154_1 for the floor in the second floor and case 154_2 for the floor in the third floor. As a result, 22 appeal cases were formalised into 25 test cases in the way described in Chapter 4. **CASE FILE NUMBER**, **SUBJECT** and **PURPOSE GROUP** in the case summaries were translated into C_{id}, C_g and C_c respectively. **ELEMENTS**, **RELATIONSHIP** and **ISSUE** were translated into C_{cd} and C_{sd}. Information given in **STATUS**, **ISSUE**, **ARGUMENTS**, **CONDITION** and **REASONS** were formalised as argumentation networks as illustrated in Figure 4.3. The $(C_{id}, C_g, C_c, C_{cd}, C_{sd})$ part of cases is used as the input to the prototype system and shown in Appendix C.

The Model Knowledge Base

The regulations related to 25 test cases were extracted from the 1981 version of the Scottish building regulations [Sco81] and formalised into 102 models. Four abstraction hierarchies were built from 102 models in the four subject areas: "safety", "health", "convenience" and "energy conservation". There are 79 models in the abstraction hierarchy of "safety", 18 models in the abstraction hierarchy of "health", 4 models in the abstraction hierarchy of "convenience", and only one model in the abstraction hierarchy of "energy conservation". No case was provided from The Scottish Office under the subject area "energy conservation", and no abstraction hierarchy was built in this subject area. Appendix B shows the MKB which contains four abstraction hierarchies built from these 102 models.

The Domain Knowledge Base

All vocabulary used in 102 models in the MKB and 25 test cases was collected and specialisation and aggregation hierarchies were built. 13 domain-specific heuristic rules described in Section 6.4.2 are also added to the DKB. The DKB used in the evaluation is shown in Appendix D.

Overall observation

Given the environments described in the previous section, 25 test cases were tested in the implemented prototype system. In order to simulate the changes of legal rules, 25 test cases were supplied to the prototype system in order of **DATES** shown in the case summaries and the results were compared to the results by the human expert recorded in the case files. In 24 cases, the system retrieved models which represent the requirements against which the case appealed. In case 87, the applicant appealed for requirements of mechanical ventilation (represented as m112). Instead of providing some means of mechanical ventilation, the applicant insisted on using windows and doors as a means of ventilation and the system retrieved m108, m109 and m116, which represents the requirements of natural ventilation. Relaxation was granted in 17 test cases and 17 weak models were added to the MKB shown in Appendix B. There was no case among those 17 relaxed cases in which rulings could be combined with existing weak models (i.e., rulings from previously relaxed cases), since issues involved in the selected cases were very diverse and there were no cases which were similar to existing weak models. Appendix E shows 17 weak models which were added the MKB. 25 test cases were also added to the CL and Appendix C shows the CL which includes 25 test cases.

Before we discuss the details of the evaluation results in Section 8.3–8.5, the next section shows an example of how the prototype system processes cases using case 93.

8.2 An Example of Running a Case

Case 93 was supplied to the prototype system after 9 test cases—case 11, 41, 42, 77, 82, 87, 90_1, 90_2 and 90_3—were processed by the prototype system. Relaxation was granted in 7 cases—case 11, 41, 42, 82, 90_1, 90_2 and 90_3—and the following 7 weak models were added to the MKB:

- 6 models in the abstraction hierarchy of "safety": m300 from case 11, m301 from case 41, m303 from case 82, m304 from case 90_1, m305 from case 90_2 and m306 from case 90_3; and
- 1 model in the abstraction hierarchy of "health": m302 from case 42.

Therefore there are 85 models in the abstraction hierarchy of "safety", 19 models in the abstraction hierarchy of "health", 4 models in the abstraction hierarchy of "convenience", and 1 model in the abstraction hierarchy of "energy conservation". All 9 cases were also added to the CL. We first explains case 93 and then show how case 93 is processed. Appendix F shows the interaction between the prototype system and the user (i.e., the expert) when the prototype system runs case 93.

8.2.1 Case 93

Figure 8.1 shows the summarised version of appeal case 93. **CASE FILE NUMBER**, i.e., the number attached to this case, is "93". The **PURPOSE GROUP** of the building in question in this case is "dwelling". The **STATUS** of the building is an existing building. The applicant in case 93 appealed for relaxation of K15 in the 1971 version of the regulations [Sco71], i.e., **REG/VERSION** is K15/1971. K15 in the 1971 version specifies requirements of "ventilation" (i.e., **SUBJECT**) in terms of the position of windows, ventilators and ventilation openings. Regulation K15 in the 1971 version is exactly the same as Regulation K14 in the 1981 version which we used to build the MKB and we use the 1981 version of the regulations to explain case 93. The following is a clause in K14 which is at issue in case 93.

CASE FILE NUMBER: 93
PURPOSE GROUP: dwellings
STATUS: existing
REG/VERSION: K15/1971
SUBJECT: ventilation; window ventilation
ELEMENTS: windows
RELATIONSHIP:
DATES: 23:08:81 to 09:12:81
ISSUE: Allow tops of the openings, of 3 bedroom windows, to be 1.8 metres above floor instead of at least 2 metres.
ARGUMENTS:
[applicant] Proposal for 3 new bedroom windows to be installed into existing eighteenth century house. The tops of the windows are at a height of 1.8 metres above the floor and the opening areas of the windows are sufficient to meet the requirements of the regulations.
[local building control authority] No good reason has been advanced to justify relaxation of the requirements of this regulation.
[local building control authority-COMMENT] BD receive all relevant drawings and correspondence according to the Building Regulations as complying with regulation 33 of the Building (Procedure) (Scotland) Regulations 1975.
[BD] BD notify building control authority that they are to relax regulation K15. BD allow building control authority 2 weeks to comment on proposed decision.
[local building control authority] The local building control authority consider that it is possible and reasonable to meet the standards and to relax the general requirement of reg. K15, particularly in sleeping rooms, is contrary to the principal of providing suitable conditions in which to live which in this case would be exacerbated by the deficiency in the area of opening provided by the existing windows.
DETERMINATION: Regulation is relaxed.
REASONS:
It would not be reasonably practicable to raise the height of the top of the window opening part.

Figure 8.1: An example of the summary of appeal cases: case 93

> K14. Every window, ventilator, permanent ventilator and permanent ventilation opening provided so as to comply with this Part shall be so positioned that the top of the opening part or of the permanent ventilation opening is not less than 2 metres above the floor: $\cdots$

K14/1981 is included in Part K, which specifies requirements of ventilation of buildings, i.e., requirements to ensure that buildings shall be so constructed as to provide adequate supply of air. The principle behind these requirements of ventilation is that all buildings should be built to ensure not to harm the health of people in and around buildings. Building components in question in case 93 are three bedroom windows (i.e., **ELEMENTS**) and the height of the top of these windows is at **ISSUE**. The applicant appealed for relaxation of K14/1981, which requires the top of the openings to be at least 2 metres. Arguments put forward in case 93 are summarised in the **ARGUMENTS** part. There were two main players, the applicant and the Local Building Control Authority (the LBCA), and arguments from these two players were as follows.

The applicant proposed to build three new bedroom windows in an existing eighteenth century house and argued that the opening areas of these three windows are sufficient to meet the requirements of the regulations, i.e., the adequate supply of air to ensure the health of people living in the building. The LBCA had refused to grant relaxation on the ground that there was no good reason advanced to justify relaxation and an appeal was put forward. The BD supported the applicant's position and granted relaxation (i.e., **DETERMINATION**). The reason for granting relaxation given by the BD (i.e., **REASONS**) was that it would not be reasonably practicable to raise the height of the top of the opening parts of the window.

8.2.2 Running Case 93

CASE FILE NUMBER, **SUBJECT**, **PURPOSE GROUP**, **ELEMENTS** and **ISSUE** are formalised into C_{id}, C_g, C_c, C_{cd} and C_{sd} as follows and supplied to the system:

C_{id}: c93
C_g: health
C_c: house
C_{cd}: bedroom(a), window(b), part_of(b,a)
C_{sd}: number(b,3), top_height_above_floor(b,1.8,metre)

CASE FILE NUMBER is 93 and C_{id} is given as "c93". **SUBJECT** in this case is "health" and C_g is given as "health". The category of the building in question is specified as "dwelling" in the **PURPOSE GROUP** part. But the applicant provides more detailed information about the building category in the **ARGUMENTS** part, i.e., "house", and C_{cd} is given as "house". Information given in the **ELEMENTS** and **ISSUE** parts—the building components at issue are three windows in a bedroom and it is proposed to build these windows to be 1.8 metres above the floor—is represented as C_{cd} and C_{sd}.

Given c93 shown above, the system first selects the abstraction hierarchy of "health", which contains 18 strong models (m4 and m100–m116) and 1 weak model (m302). Then after comparing the C_c and the M_c of these 19 models, the system selects 10 models, m4, m100, m101–106, m116 and m302, whose M_c includes a building category that subsumes the category given in the C_c, "house". The M_c of m4, m100, m116 and

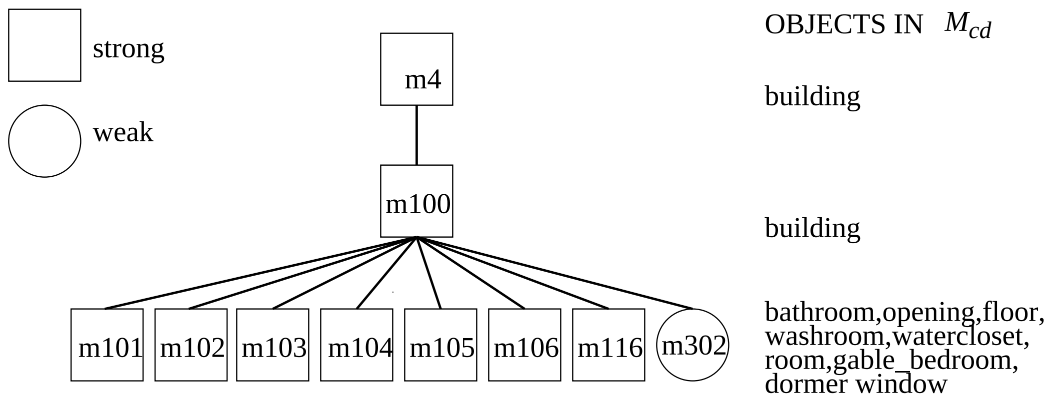

Figure 8.2: The hierarchy of models retrieved in c93

m302 is "building", which is a subclass of "house" and m4, m100, m116 and m302 are selected. The M_c of m101–m106 is "house", which is the same category as given in the C_c, and m101–m106 are selected as well. But the M_c of m107–m113 is "assembly building, recreational building", the M_c of m114–m115 is "storage, warehouse", and m107–m113 and m114–m115 are not selected. The hierarchy of models retrieved so far is illustrated in Figure 8.2.

The next step is to select models whose M_{cd} contains building components in the right abstraction level so that building components in the M_{cd} are comparable to the building components in the C_{cd}, "bedroom" and "window". m4 is at the first level of the abstraction hierarchy of "health" and there is only one building component in the M_{cd} of m4, "building". m100 is at the second level and there is only one building component in the M_{cd} of m4, "building". "building" is at the higher level in the aggregation hierarchies than "bedroom" and "window" and therefore m4 at the first level and m110 at the second level are not selected. m101–m106, m116 and m302 are at the third level and building components in the M_{cd} of these models are "bathroom", "opening", "floor", "washroom", "watercloset", "room", "gable bedroom" and "dormer window", which are at the same level in the aggregation hierarchies as "bedroom" and "window". Therefore 8 models at the third level, m101–m106, m116 and m302, are selected for the similarity assessment between M_{cd} and C_{cd}. The C_{cd} of c93 and the M_{cd}s of the 8 selected models are shown below.

C_{cd} of c93:	bedroom(a), window(b), part_of(b,a)
M_{cd} of m101:	bathroom(A), opening(B), part_of(B,A), floor(C), part_of(C,A)
M_{cd} of m102:	bathroom(A), opening(B), part_of(B,A)
M_{cd} of m103:	washroom(A), opening(B), part_of(B,A), floor(C), part_of(C,A)
M_{cd} of m104:	washroom(A), opening(B), part_of(B,A)
M_{cd} of m105:	watercloset(A), opening(B), part_of(B,A), floor(C), part_of(C,A)
M_{cd} of m106:	watercloset(A), opening(B), part_of(B,A)
M_{cd} of m116:	room(A), opening(B), part_of(B,A)
M_{cd} of m302:	gable_bedroom(A), dormer_window(B), part_of(B,A), height_above_floor(A,2070,millimetre)

First, the M_{cd} s of m102, m104, m106 and m116 have the exactly same number of object type predicates and relation predicates as the C_{cd} of c93 and the system compares each M_{cd} of m102, m104, m106 and m116 to the C_{cd} to see whether there is simple similarity.

- m102 : An opening "B" in the M_{cd} is a superclass of a window "b" in the C_{cd} and "B" and "b" are of similar object types. But a bathroom "A" in the M_{cd} is neither a subclass nor a superclass of a bedroom "a" in the C_{cd}. Therefore there is no simple similarity.

- m104: An opening "B" in the M_{cd} is a superclass of a window "b" in the C_{cd} and "B" and "b" are of similar object types. But a washroom "A" in the M_{cd} is neither a subclass nor a superclass of a bedroom "a" in the C_{cd}. Therefore there is no simple similarity.

- m106: An opening "B" in the M_{cd} is a superclass of a window "b" in the C_{cd} and "B" and "b" are of similar object types. But a watercloset "A" in the M_{cd} is neither a subclass nor a superclass of a bedroom "a" in the C_{cd}. Therefore there is no simple similarity.

- m116: A room "A" in the M_{cd} is a superclass of a bedroom "a" in the C_{cd} and "A" and "B" are of similar object types. An opening "B" in the M_{cd} is a superclass of a window "b" in the C_{cd} and "B" and "b" are of similar object types. The relation between "A" and "B" exactly matches the relation between "a" and "b", which correspond to "A" and "B" respectively. Therefore there is semantic similarity between the M_{cd} of m116 and the C_{cd} of c93.

The M_{cd} of m116 is found to have simple semantic similarity with the C_{cd} of c93. Therefore embedded similarity and implied similarity are not tested and m116 is retrieved as relevant to c93. m116 represents requirements stated in K14/1981 which requires the top of the openings to be at least 2 metres as shown below.

C_{cd} of c93:	number(b,3), top_height_above_floor(b,1.8,metre)
M_{cd} of m116:	top_height_above_floor(B,X,metre), X >= 2

The top of the windows in question is proposed to be "1.8 metres", which does not satisfy the requirement specified in the M_{sd} of m116. Therefore there is a need to consider relaxation and relevant information stored within the system is explained to the expert.

There are two kinds of information to be explained: the principles behind non-compliant requirements represented as m116, which provide the reason why requirements represented as m116 should be satisfied, and similar cases stored in the CL. m116 represents requirements included in K14/1981, which specifies requirements of ventilation of buildings, i.e., requirements set out to ensure that buildings shall be so constructed as to provide adequate supply of air. These principles behind m116 are represented as ancestor models of m116, i.e., m4 and m100, in the abstraction hierarchy of "health" as shown in Figure 8.2 and explained to the expert as demonstrated in Appendix F. Previous similar cases are cases attached to the retrieved model m116. There are two cases attached to m116, c42 and c87, and these two cases are also explained to the expert as demonstrated in Appendix F.

The system also checks whether the C_{sd} of c93 conflicts with any M_{sd} of models in abstraction hierarchies other than the abstraction hierarchy of "health". The C_{sd} of c93 is given in terms of the "number" of windows and the "top_height_above_floor" of windows. There is no model whose M_{sd} is specified in those terms in the abstraction hierarchies of "safety", "convenience" and "energy conservation". This is informed to the expert and the system asks whether relaxation to be granted or not.

In this case, as shown in the previous section, relaxation was granted. The expert answers "y", and c93 is generalised and added to the MKB as follows. First, to guess the new M_c, the system looks for strong models whose parent model is m116 and which specify requirements in the same context as c93, i.e., whose M_c subsumes the

C_c, whose M_{cd} is similar to the C_{cd}, and whose M_{sd} is given in terms of the "number" of windows and the "top_height_above_floor" of windows. There is one model, m116, whose M_c subsumes the C_c and whose M_{cd} is similar to the C_{cd}. But the M_{sd} of m116 is specified only in terms of the "top_height_above_floor" of windows. Therefore the C_c of c93 becomes the new M_c, "house".

Second, to create the new M_{cd} and M_{sd}, the system looks for weak models whose parent model is m116 and which specify requirements in the same context as c93, i.e., whose M_c subsumes the C_c, whose M_{cd} is simply similar to the C_{cd}, and whose M_{sd} is given in terms of the "number" of windows and the "top_height_above_floor" of windows. There is no such model and the new M_{cd} and M_{sd} are created from the C_{cd} and C_{sd} of c93. The new M_{cd} is created by replacing constants, "a" and "b", in the C_{cd} of c93 with variables, "A" and "B", i.e., "bedroom(A), window(B), part_of(B,A)". A new M_{sd} is created by copying the pattern of the "number" property of windows and the "top_height_above_floor" property of windows used in the M_{sd} of m116. The M_{sd} of m116 is specified only in terms of the *minimum* "top_height_above_floor" of windows, the predicate "top_height_above_floor(b,1.8,metre)" is generalised by following this pattern. A new predicate "top_height_above_floor(B,D,metre), D>=1.8" is created by replacing constants, "b" and "1.8", with variables, "B" and "D", and introducing a new expression "D>=1.8". The predicate "number(b,3)" is generalised as "number(B,C), C=3".

Finally, the M_s is set as "weak" and a new M_{id}, m307, is given to this newly created model. A complete model m307 is as follows and m307 is added to the abstraction hierarchy of "health" as a child model of m100.

M_{id}:	m307
M_c:	house
M_{cd}:	bedroom(A), window(B), part_of(B,A)
M_{sd}:	number(B,C), C=3, top_height_above_floor(B,D,metre), D>=1.8
M_s:	weak

8.3 The Evaluation of the Retrieval Procedure

The functionality of the system is divided into two distinct procedures: the retrieval of relevant regulatory information (see Chapter 6) and the acquisition of new regulatory information (see Chapter 7). We present and discuss the evaluation results in respect of the retrieval procedure in this section and the next section. The details of the evaluation results on the acquisition procedure is described in Section 8.5.

8.3.1 Overall Results

The input to the retrieval procedure was the C_g, C_c and C_{cd} of 25 test cases. Table 8.1, 8.2 and 8.3 show the test results for each subject area: Table 8.1 in safety, Table 8.2 in health, and Table 8.3 in convenience. There are six columns labelled as A–F in those tables and each column represents the followings:

- A: The id number of the model which represents the requirements against which the case appealed.
- B: The number of models in the abstraction hierarchy of the given goal C_g of the case (this is constant for each table but included for easy comparison with C and D).
- C: The number of models retrieved after the retrieval by category.
- D: The number of models retrieved after the selection of models at the right abstraction level.
- E: The number of models retrieved after the similarity assessment.
- F: The id number of models retrieved by the similarity assessment and the kind of similarity that exists between the M_{cd} of each retrieved model and the C_{cd} of the case.

case	A	B	C	D	E	F
c11	m23	79	65	42	2	m23 (embedded literal similarity)
						m24 (embedded literal similarity)
c41	m48	79	63	58	1	m48 (literal similarity)
c77	m65	79	63	21	1	m65 (embedded semantic similarity)
c82	m25	79	63	42	2	m25 (semantic similarity)
						m30 (analogy+semantic similarity)
c90_1	m23	79	63	42	1	m23 (literal similarity)
c90_2	m26	79	63	42	1	m26 (semantic similarity)
c90_3	m27	79	63	58	1	m27 (literal similarity)
c95	m39	79	65	60	1	m39 (embedded semantic similarity)
c97	m84	79	57	52	1	m84 (implied embedded literal similarity)
c110	m18	79	63	58	1	m18 (implied embedded semantic similarity)
c131	m67	79	63	42	3	m67 (literal similarity)
	m70					m70 (literal similarity)
	m73					m73 (literal similarity)
c154_1	m45	79	65	60	1	m45 (implied embedded literal similarity)
c154_2	m45	79	65	60	1	m45 (implied embedded literal similarity)
c156	m65	79	63	21	1	m65 (literal similarity)
c165	m56	79	65	39	1	m56 (semantic similarity)
c0012	m14	79	66	61	4	m14 (implied embedded literal similarity)
						m41 (implied embedded literal similarity)
						m42 (implied embedded literal similarity)
						m43 (implied embedded literal similarity)

Table 8.1: The test result of the retrieval procedure: safety

case	A	B	C	D	E	F
c42	m116	18	3	1	1	m116 (embedded semantic similarity)
c87	m112	18	10	8	3	m108 (embedded semantic similarity) m109 (embedded semantic similarity) m116 (embedded semantic similarity)
c93	m116	18	9	7	1	m116 (semantic similarity)
c134	m115	18	5	3	1	m115 (implied embedded semantic similarity)
c141	m106	18	9	7	2	m106 (semantic similarity) m116 (semantic similarity)

Table 8.2: The test result of the retrieval procedure: health

case	A	B	C	D	E	F
c116	m201	4	4	2	1	m201 (implied embedded literal similarity)
c132	m201	4	4	2	1	m201 (implied embedded semantic similarity)
c150	m202	4	4	2	1	m202 (implied embedded literal similarity)
c172	m202	4	4	2	1	m202 (implied embedded literal similarity)

Table 8.3: The test result of the retrieval procedure: convenience

We can observe from Table 8.1, 8.2 and 8.3 that abstraction hierarchies in the MKB serve as efficient indices. First, abstraction hierarchies enable the system to focus on one particular group of models by pruning models unlikely to be relevant to the input case before the similarity assessment begins. Second, there is no need to compare the input case with all cases in the CL but only models in the abstraction hierarchies need to be compared with the input case. The number of strong models are fixed, and the number of weak models in the MKB will increase slower compared to the increase in the number of cases in the CL since the rulings only from cases in which the relaxation was granted are added to the MKB by integrating the rulings from those cases into the existing abstraction hierarchies in the MKB. As the number of cases in the CL increases well over the number of models in the MKB, the input case needs to be compared only to models in the MKB to find relevant cases.

In the similarity assessment, relevant models are found by using the notion of structural mapping (i.e., simple similarity and embedded similarity) in 15 cases. But the system fails to find relevant models in 10 cases by using the notion of structural mapping and uses the notion of implied similarity. In all 25 test cases except c87, the retrieval procedure retrieves a model or models which represent the regulations against which the case appealed. In 5 cases among these 24 cases, c11, c82, c131, c141 and c0012, in addition to a model (or models) which represents the requirements against which the input case appealed, other models are retrieved. In c87, the applicant appealed against requirements of mechanical ventilation represented as m112. But, instead of providing some means of mechanical ventilation, which the LBCA proposed to provide, the applicant insisted on natural ventilation, i.e., using windows and doors as a means of ventilation, and the retrieval procedure does not retrieve m112. We first explain why we have obtained such results in these 6 cases—c11, c82, c87, c131, c141 and c0012—in detail and then discuss the implications of the observations in these cases.

8.3.2 Case 11

c11 appealed for the relaxation of the requirements of stairs stated in Regulation D7 and represented as m23. In this case, two models, m23 and m24 are retrieved by the embedded similarity criterion. The C_{cd} of c11 and the M_{cd} s of m23 and m24 are shown below.

C_{cd} of c11: stair(a), stairway(c), used_as(c,exit),
attribute(c,enclosed),
part_of(a,c),
building(b), is(b,small), part_of(c,b)

M_{cd} of m23: stair(A), stairway(B), used_as(B,exit),
attribute(B,enclosed),
part_of(A,B)

M_{cd} of m24: stair(A), stairway(B),
attribute(B,unenclosed), attribute(B,external),
part_of(A,B)

There is extraneous information provided in c11 about the building in which the stair and the stairway in question are located, i.e., "building(b), is(b,small), part_of(c,b)", and the remaining description of the C_{cd} of c11 is literally similar to the M_{cd} of m23. The stairway "B" in the M_{cd} of m24 is an unenclosed external stairway. The stairway "c" in the C_{cd} of c11, which corresponds to "B" in the M_{cd} of m24, is an enclosed stairway and is used as an exit. The fact that the stairway "c" is used as an exit (i.e., "used_as(c,exit)") does not match any predicate in the M_{cd} of m24, but m24 is still retrieved by the the embedded similarity criterion.

The observation in this case suggests the following. First, in m24, it needs to be clearer about how much difference in object attributes should be allowed in order to be recognised by the analogy criterion. For example, the stairway "B" in m24 is unenclosed and external, and the stairway "c" in c11 is enclosed. These two stairways would normally be distinguished in the context of case 11 ("fire safety") but they are regarded as relevant by the analogy criterion. Second, when a model is retrieved by embedded similarity, it needs to be sure that the extraneous information included in C_{cd}, such as "used_as(c,exit)" in c11, is not relevant to the issue of the input case.

8.3.3 Case 82

In c82, in addition to m25 which represents the requirements of separating walls which are stated in Regulation D7 and against which c82 appealed, m30 is retrieved by the analogy and semantic similarity criteria. The C_{cd} of c82 and the M_{cd} s of m25 and m30 are shown below.

C_{cd} of c82: loadbearing_partition_wall(a)
M_{cd} of m25: separating_wall(A)
M_{cd} of m30: wall(A), attribute(A, external)

m30 specifies the requirement of external walls, and this is also stated in Regulation D7. But whether the loadbearing wall "a" is internal or external is not specified in the C_{cd} of c82, and m30 is retrieved by the analogy criterion. In order to discriminate m25, which is more relevant to c82, from m30, we need to connect the descriptions of attributes more closely to the issues underlying the requirements.

8.3.4 Case 87

In c87, the LBCA proposed to install fans in the church hall to compensate the loss of ventilation windows (stated in K10 and represented as a model m112), but the applicant insisted that the church hall met the regulations because five doors in the church hall provide enough natural ventilation (stated in K10 and represented as m108 and 109). The case description does not include fans and therefore m112 is not retrieved. Instead, m108, m109 and m116, which specify the requirements of natural ventilation are retrieved. The C_{cd} of c87 and the M_{cd}s of m108, m109 and m116 are as follows.

C_{cd} of c87: churchhall(a), not_used_as(a,storage),
not_require(a,temperature_control),
cubic_space_per_occupant(a,12,cubic_metre),
window(b), part_of(b,a),
floor(c), part_of(c,a),
door(d), part_of(d,a)
M_{cd} of m108: room(A), not_used_as(A,storage),
not_require(A,temperature_control),
cubic_space_per_occupant(A,X,cubic_metre),
X > 2.8, X =< 21,
floor(B), part_of(B,A),
opening(C), part_of(C,A)
M_{cd} of m109: room(A), not_used_as(A,storage),
not_require(A,temperature_control),
cubic_space_per_occupant(A,X,cubic_metre),
X > 2.8, X =< 21,
opening(C), part_of(C,A)
M_{cd} of m116: room(A), opening(B), part_of(B,A)

m116 specifies the general requirements of windows and ventilators. The M_{cd} of m116 describes simpler circumstances than the M_{cd}s of m108 and m109, and larger part of the C_{cd} of c87 matches the M_{cd}s of m108 and m109 than the M_{cd} of m116. This suggests that, when more than one model is retrieved by embedded similarity, the percentages of predicates in the C_{cd} which match predicates in the M_{cd} of retrieved models could be used as a criterion to rank the relevance among retrieved models.

8.3.5 Case 131

c131 sought the relaxation on the guarding of stairs stated in Regulation S3. In the given circumstances in c131, 3 models, m67, m70 and m73, are retrieved. The M_{cd}s of m67, m70 and m73 are exactly the same as shown below.

C_{cd} of c131: access_stairway(a), is(a,external),
stair(b), part_of(b,a),
height_above_landing(b,620,millimetre),
landing(c), connected(b,c)

M_{cd} of m67, m70 and m73: access_stairway(A), is(A,external),
stair(B), part_of(B,A),
height_above_landing(B,X,millimetre),
X > 600,
landing(C), connected(B,C),

In the given circumstances described as the M_{cd} of m67, m70 and m73, Regulation S3 provide three different ways of satisfying the requirements of the guarding of stairs: by providing a guarding wall as specified in m67; by providing a balustrade as specified in m70; or by providing a railing as specified in m73. The retrieval procedure does not take into account the criteria specified in the C_{sd} of the input case, and therefore these three models are retrieved. In order to discriminate one model as relevant, the system needs to use the information given in the C_{sd} of the input case.

8.3.6 Case 141

In c141, in addition to m106 which represents the requirements which are stated in Regulation K6 and against which c141 appealed, m116, which specifies the general requirements of windows stated in the regulation K14, is retrieved. As shown below, the M_{cd} of m116 is similar to the C_{cd} of c141 by the semantic similarity criterion.

C_{cd} of c141: watercloset(a), window(b), part_of(b,a)
M_{cd} of m106: watercloset(A), opening(B), part_of(B,A)
M_{cd} of m116: room(A), opening(B), part_of(B,A)

However, the M_{sd} of m116 is described in terms of the top height of the opening above the floor, which is not in question in c141. As in c131, to identify m116 as an irrelevant model, the information given in the C_{sd} of the input case needs be used.

8.3.7 Case 0012

In c0012, in addition to m14 which represents the requirements which are stated in Regulation D3 and against which c0012 appealed, m41, m42 and m43 which represent the requirements stated in Regulation E6 are retrieved. Regulation E6 specifies the number of escape routes from a storey to ensure adequate means of escape in case of fire, which is not the issue in this case. But the C_{cd} of c0012 contains information about a compartment which can be thought of as part of a storey in the building, and m41, m42 and m43 are retrieved by the implied similarity criterion. The C_{cd} of c0012 and the M_{cd}s of retrieved models are shown below.

C_{cd} of c0012: building(a), area(a, 1200, square_metre),
compartment(b), part_of(b, a),
has(a, full_automatic_fire_control_system)
M_{cd} of m14: building(A), number_of_storey(A,N), N = 1,
storey(C), part_of(C,A),
area(C,X,square_metre), X >= 1000,
compartment(B), part_of(B,A)
M_{cd} of m41: storey(A),
occupancy_capacity(A,X), X >= 1, X =< 60
M_{cd} of m42: storey(A),
occupancy_capacity(A,X), X >= 61, X =< 600
M_{cd} of m43: storey(A),
occupancy_capacity(A,X), X > 600

To eliminate m41–m43 as irrelevant to c0012, the system needs to know more detailed subjects (or subgoals) in which the input case is brought out (or aims to achieve) and what building components and what properties of building components are related to what issues (or goals).

8.3.8 Discussion

The observations described in six previous sections suggest possible improvements on the similarity assessment.

First, the analogy criterion needs to be refined. As illustrated in c11, when attributes of corresponding objects are different or when an attribute of an object is specified in M_{cd} but the attribute of a corresponding object is missing in the C_{cd} of the input case, these objects may be absolutely unrelated objects. Therefore, we need to understand more about how much difference should be allowed in the analogy criterion.

Second, it would be possible to devise heuristics to rank models when more than one model is retrieved. For example, it would be possible to rank retrieved models by the kind of similarity which the model has. The system implicitly ranks the relevant models in the order of simple similarity, embedded similarity and implied similarity. It would also be possible to rank models within one of those similarity criteria. The following are plausible ways of doing it.

- In the simple similarity criterion, a model which is similar by the literal similarity criterion could be favoured more than a model which is similar by the analogy criterion or the semantic criterion.
- In the embedded similarity criterion, as illustrated in c87, if there are models which are similar by the embedded similarity criterion, a model whose M_{cd} has more matches than other models could be favoured.
- In the embedded similarity criterion, as Keane points out in [Keane 88], instead of simply considering identical matches and non-identical matches, the notion of "the salience of primitives relative to other", i.e., the salience of features to other features in a situation, could be introduced and used.

Third, the C_{sd} of the input case and the M_{sd} s of the retrieved models could be taken into consideration after the similarity assessment. The regulations sometimes prescribe the requirements in terms of different criteria, for example, K10 in c87 and S3 in c131, and only one of those requirements needs to be satisfied. It could be possible to compare the C_{sd} of the input case and the M_{sd} of retrieved models, and select only models which specify the M_{sd} in similar terms to the C_{sd} of the input case. This may be more complicated than simply comparing the attributes and the relations. The system may need to know what attributes and properties of building components are relevant to what issues, as we discuss below.

Fourth, subgoals of the input case could be taken into consideration during the retrieval procedure. The importance of contexts or contextual goals in analogical reasoning was pointed out by Keane in [Keane 88], and Thagard and Holyoak's Constraint-Satisfaction Theory makes use of goals as part of their pragmatic centrality constraint. The retrieval procedure in our system also makes use of goals given in the input case, i.e., C_g. However, a goal given in C_g is very abstract so that irrelevant models are retrieved in some cases. For example, the issue in c0012 is the maximum area of the compartment to ensure the structural stability in case of fire. Only m14 is relevant to this issue. But m41, m42 and m43 whose requirements are to ensure adequate means of escape from fire are also retrieved. The parent model of m14 is m6, which states the issue of the structural stability in case of fire, and the parent model of m41, m42 and m43 is m7, which states the issue of adequate means of escape from fire. m6 and its descendant models constitutes a subhierarchy of models which represents an abstraction hierarchy of requirements to ensure the structural stability in case of fire. m7 and its descendant models constitutes a subhierarchy of models which represents an abstraction hierarchy of requirements to ensure adequate means of escape from fire. If the subgoal of the input case could be identified, one of two subhierarchies could be eliminated. But identifying the subgoal of the input case might be tricky unless it is explicitly provided by the user. Without the assistance of the user, the system needs to know more about the domain, for example, what subgoals are related to what goals, what building components are related to what issue, what criteria or properties are related to what issue, etc.

8.4 Experiments on the Similarity Assessment

We have conducted a series of experiments with 3 test cases, c87, c90_1 and c141, to test the flexibility of the similarity assessment described in Section 6.1–6.2. In the first experiment, we vary the amount of information provided to the system in c87 to show the significance of the priorities given to various kinds of similarity. In the second experiment with c141, the tolerance of the matching algorithm is tested. In the third experiment, we demonstrate the importance of representational choices in the matching algorithm using c90_1.

8.4.1 Experiment-1: Priorities in the Similarity Criteria

In c87, the applicant appealed for the relaxation of the requirements of natural ventilation represented as m108, m109 and m116. The C_g, C_c and C_{cd} of c87 are given as follows:

c87
C_g: health
C_c: church
C_{cd}: churchhall(a),
not_used_as(a,storage), not_require(a,temperature_control),
cubic_space_per_occupant(a,12,cubic_metre),
window(b), part_of(b,a),
floor(c), part_of(c,a),
door(d), part_of(d,a)

8 models are selected for the similarity assessment:m107–m113 and m116. Among these 8 models, three models, m108, m109 and m116, specify the requirements of natural ventilation in terms of openings and the circumstances around the openings in question. The C_{cd} of c87 includes two kinds of openings, a window "b" and a door "d". The window "b" which appears first in the C_{cd}, matches the openings in the M_{cd}s of m108, m109 and m116, and m108, m109 and m116 are retrieved. In this experiment, we created two new cases from c87. In c87_1, the information about the window "b", i.e., "window(b), part_of(b,a)", is removed. In c87_2, the information about the door "d", i.e., "door(d), part_of(d,a)", is removed. The C_{cd}s of these new cases are shown below.

C_{cd} of c87_1: churchhall(a), not_used_as(a,storage),
not_require(a,temperature_control),
cubic_space_per_occupant(a,12,cubic_metre),
floor(c), part_of(c,a),
door(d), part_of(d,a)

C_{cd} of c87_2: churchhall(a), not_used_as(a,storage),
not_require(a,temperature_control),
cubic_space_per_occupant(a,12,cubic_metre),
window(b), part_of(b,a),
floor(c), part_of(c,a),

In these two new cases, only m108 is retrieved as relevant by the semantic similarity criterion as shown in Table 8.4. m109 and m116 are similar by the embedded semantic criterion, but simple similarity has priority in the similarity assessment over embedded similarity, and m109 and m116 are not retrieved.

case	D	E	F
c87	8	3	m108 (embedded semantic similarity) m109 (embedded semantic similarity) m116 (embedded semantic similarity)
c87_1	8	1	m108 (semantic similarity)
c87_2	8	1	m108 (semantic similarity)

Table 8.4: The results of the experiment-1 with c87

Discussion

The system give priorities in order of simple similarity, embedded similarity and implied similarity. However, as this experiment demonstrates, there may be cases in which relevant models are missed out because of the priorities among similarity criteria. Instead of assigning a fixed order of priorities to the kinds of similarity, it is plausible to retrieve all models which have any similarity and then to introduce an evaluator which compares the input case and the retrieved models and judges the soundness of relevance as suggested by Gentner [Gentner & Landers 89].

8.4.2 Experiment-2: The Tolerance of the Matching Algorithm

Given the description of c141 as below, 7 models are selected for the similarity assessment: m101–m106 and m116.

c141
C_g: health
C_c: house
C_{cd}: watercloset(a), window(b), part_of(b,a)

The M_{cd}s of m101–m106 and m116 are specified in terms of openings or windows in various kinds of a room. Given the C_{cd} of c141, m106 and m116 are retrieved as relevant. In this experiment, we created three new cases, c141_1, c141_2 and c141_3, by changing objects in the C_{cd} of c141 to objects of slightly different but still similar object types. In c141_1, a window "b" is changed to an opening "b". In c141_2, a watercloset "a" is changed to a room "a". In c141_3, a window "b" is changed to a rooflight "b". The C_{cd}s of these three cases are listed below.

C_{cd} of c141_1: watercloset(a), opening(b), part_of(b,a)
C_{cd} of c141_2: room(a), window(b), part_of(b,a)
C_{cd} of c141_3: watercloset(a), rooflight(b), part_of(b,a)

The results in three new cases are shown in Table 8.5. In c141_1 and c141_3, m106 and m116 are retrieved as relevant as in c141. In c141_2, object types given are too general, i.e., a room which could match any kind of a room, and too many models, i.e., four models, are retrieved. But m106 and m116, which are relevant to c141_2, are still retrieved.

case	D	E	F
c141	7	2	m106, m116 (semantic similarity)
c141_1	7	2	m106 (literal similarity) m116 (semantic similarity)
c141_2	7	4	m102 (semantic similarity) m104 (semantic similarity) m106 (semantic similarity) m116 (semantic similarity)
c141_3	7	2	m106, m116 (semantic similarity)

Table 8.5: The results of the experiment-2 with c141

Discussion

The results of this experiment can be seen as an example of the Keane's claim in [Keane 88] that "the development of analogical mechanisms will ease the problem of 'brittleness' in expert systems." The matching mechanism used to select models relevant to the input case not only bridges the inevitable terminological gaps which exist between models and cases in our domain but also tolerates small changes in the domain description. However, as demonstrated in c141_2, the tolerance of the matching algorithm has to be limited to some degree, since irrelevant models would be retrieved if not.

8.4.3 Experiment-3: Representational Choices

The issue in c90_1 is the material of the enclosed stairway used as an exit. Given the description of C_g, C_c and C_{cd} shown below, 42 models, m17–37 and m65–m85, are retrieved for the similarity assessment.

c90_1
C_g: safety
C_c: house
C_{cd}: stair(a),
stairway(b), used_as(b,exit), attribute(b,enclosed),
part_of(a,b)

m23 specifies the requirements of the material of the enclosed stairway used as an exit, and is retrieved by the literal similarity criterion. Only one attributes of the stairway "b", "enclosedness", is provided in c90_1. There are other attributes which the stairway "b" might have. In this experiment, we changed the attribute of the stairway "b" in three ways. In c90_1_1, no attribute is specified. In c90_1_2, more information about the attribute of the stairway "b" is given, such as "externalness". In c90_1_3, the least information about the stairway "b" is given, i.e., no attribute and no usage of the stairway is specified at all.

C_{cd} of c90_1_1: stair(a),
stairway(b), used_as(b,exit),
part_of(a,b)

C_{cd} of c90_1_2: stair(a),
stairway(b), used_as(b,exit),
attribute(b,enclosed), attribute(b,external),
part_of(a,b)

C_{cd} of c90_1_3: stair(a),
stairway(b),
part_of(a,b)

Table 8.6 shows the results in this experiment. In c90_1_1, the information that the stairway "b" is enclosed is missing, but m23 is still retrieved by the analogy criterion. In c90_1_2, instead of m23, m24 is retrieved by the embedded analogy criterion. Regardless of whether a stairway is external or enclosed, a stairway used as an exit needs to satisfy only the requirements represented as m23 (D7). In this case, however, too much information about the stairway "b" is given, i.e., "enclosed" and "external", and m24, whose M_{cd} includes the description of these two attributes, is retrieved. In c90_1_3, any information about the stairway "b" is not given. Too little information about the stairway "b" is given, and m24 is retrieved by the analogy criterion.

case	D	E	F
c90_1	42	1	m23 (literal similarity)
c90_1_1	42	1	m23 (analogy)
c90_1_2	42	1	m24 (embedded analogy)
c90_1_3	42	1	m24 (analogy)

Table 8.6: The results of the experiment-3 with c90_1

Discussion

The experiment-3 demonstrates the importance of the amount of information provided to the similarity assessment. There are two other representational issues which can be explained using c90_1 in this experiment. The first issue is how to choose the predicate type in representing a piece of information. The second and more fundamental one is (as Palmer pointed out in [Palmer 89] and referred to as "relation-sorting problem") how to decide which predicate type needs to match and which predicate type does not need to match. We illustrate these issues using c90_1.

In c90_1, the information that the stairway "b" is used as an exit is represented using an object type predicate "stairway(b)" and a relation predicate "used_as(b,exit)". There are three predicate types used to represent the domain—object type predicates, attribute predicates and relation predicates—and three predicate types are treated differently in the similarity assessment. In c90_1, the information that an object "b" is used as an exit is represented as a relation predicate "used_as(b,exit)". This means that only models whose M_{cd} contains exactly the same relation predicate, such as "used_as(X,exit)" in which "X" corresponds to the object "b", will be retrieved and no model whose M_{cd} does not contain a relation predicate "used_as(X,exit)" will be retrieved. The same piece of information can be represented using different predicate types as shown below, and be treated differently.

In the C_{cd} of c90_1 :	stairway(b), used_as(b,exit)
First variation:	exit_stairway(b)
Second variation:	stairway(b), attribute(b,used_as_exit)

In the first variation, an object type predicate "exit_stairway(b)" is used to represent the information that the object "b" is used as an exit. In the second variation, an attribute predicate "attribute(b,used_as_exit)" is used to represent the same piece of information. In the first variation, "exit stairway" matches any object which is similar, for instance, stairway, by the semantic criterion, and therefore models whose M_{cd} contains a stairway which is not used as an exit may be retrieved. In the second variation, any stairway matches a stairway "b" by the analogy criterion, and models whose M_{cd} contains a stairway which is not used as an exit may be retrieved. In other words, differences in the way in which a piece of information is represented may produce different results and, in worst cases, irrelevant models may be retrieved. In summary, in order to devise a matching algorithm, first, we have to make choices about what representational primitives to be used to represent information (or knowledge), second, we have to decide what kinds of similarity need to be defined, and, finally, we have to decide what representational primitives need to match to identify what kinds of similarity.

8.5 The Evaluation of the Acquisition Procedure

The input to the system was the C_g, C_c, C_{cd} and C_{sd} of 25 test cases. The 25 test cases are all appeal cases. Therefore none of the C_{sd} of the 25 test cases satisfies the M_{sd} of retrieved models as relevant and there is a need to consider relaxation in all 25 test cases. When the system asks if relaxation is to be granted or refused, the **DETERMINATION** by the BD in the case summaries is given to the system as the answer. When the answer is yes, rulings, i.e., C_c, C_{cd} and C_{sd}, are generalised and integrated into the existing abstraction hierarchies. Table 8.7, 8.8 and 8.9 show the results for each subject area: Table 8.7 in safety, Table 8.8 in health, and Table 8.9 in convenience. There are four columns labelled as A, G–I in those tables and each column represents the following:

- A: The id number of the model which represents the requirements against which the case appealed.
- G: Whether relaxation was grated or refused.
- H: The id number of newly added weak models which represent legal rules generalised from the input case when relaxation was granted.
- I: The id number of models to which the input case became attached, i.e., the id number of models which were retrieved as relevant to the input case.

Relaxation was granted in 17 test cases—case 11, 41, 42, 82, 90_1, 90_2, 90_3, 93, 95, 97, 110, 116, 131, 132, 134, 154_1 and 154_2—and 17 weak models (see Appendix E) were added to the MKB. Since issues involved in the selected cases were very diverse, there are no cases among those 17 cases in which C_{cd} could be combined with the M_{cd} of weak models that were added in previous cases.

In 3 test cases, c11, c42 and c82, a new M_c was created by generalising the M_cs of strong models which have the same parent model as the relevant strong models and which specify requirements in the same context as the input case. For example, in c11, the category of the building in question, i.e., C_c, is “office”, and m23 and m24 are retrieved as relevant. m23 and m24 apply to all categories of buildings, and relaxation

case	A	G	H	I
c11	m23	granted	m300	m23,m24
c41	m48	granted	m301	m48
c77	m65	refused	-	m65
c82	m25	granted	m303	m25,m30
c90_1	m23	granted	m304	m23
c90_2	m26	granted	m305	m26
c90_3	m27	granted	m306	m27
c95	m39	granted	m308	m39
c97	m84	granted	m309	m84
c110	m18	granted	m310	m18
c131	m67,m70,m73	granted	m312	m67,m70,m73
c154_1	m45	granted	m315	m45
c154_2	m45	granted	m316	m45
c156	m65	refused	-	m65
c165	m56	refused	-	m56
c0012	m14	refused	-	m14,m41,m42,m43

Table 8.7: The test result of the acquisition procedure: safety

case	A	G	H	I
c42	m116	granted	m302	m116
c87	m112	refused	-	m108,m109,m116
c93	m116	granted	m307	m116
c134	m115	granted	m314	m115
c141	m106	refused	-	m106,m116

Table 8.8: The test result of the acquisition procedure: health

case	A	G	H	I
c116	m201	granted	m311	m201
c132	m201	granted	m313	m201
c150	m202	refused	-	m202
c172	m202	refused	-	m202

Table 8.9: The test result of the acquisition procedure: convenience

of requirements represented as m23 and m24 is granted in the context of c11. Therefore rulings made in c11 may also apply to other purpose groups of buildings in cases which arise in the similar context as c11 in the future. The M_c of m300, "building", reflects this generalisation of the rulings made in c11.

In 5 test cases, c42, c93, c95, c154_1 and c154_2, C_{sd} was generalised by looking up the M_{sd} of strong models which have the same parent model as the relevant strong models and which specify requirements in the same context as the input case. In those cases, the properties of objects are quantified numerically, and the generalisation of descriptions was possible by adjusting property values (the parameter adjustment described Section 7.4.2). For example, in c42 and c93, by copying the patterns given in the M_{sd} of m116, "top_height_above_floor(B,X,metre), X>=2", the M_{sd} s of m302 and m307 are created as follows:

In the M_{sd} of m302 : top_height_above_floor(B,D,metre), D>=1.75
In the M_{sd} of m307 : top_height_above_floor(B,D,metre), D>=1.8

The system also uses the type generalisation technique (see also Section 7.4.2) in generalising C_{sd}. The type generalisation technique necessitates the existence of specialisation relations of concepts used in C_{sd}. However, there are 7 test cases, c11, c90_1, c90_1, c116, c132, c150 and c172, in which the issue was how to define a legal concept and some vocabulary used to describe the C_{sd} s of these 7 test cases cannot be included in the specialisation hierarchies in the DKB. As a consequence, it was impossible to generalise the C_{sd} s of these 7 cases, and the C_{sd} s had to become the M_{sd} of newly created weak models.

For example, in c11 and c90_1, the applicant appealed for relaxation of Regulation D7, which requires that stairs in the enclosed stairway be constructed of non-combustible materials and is represented as m23. The debate in c11 and c90_1 was about a particular kind of materials used to build stairs in the enclosed stairway: "hardwood" in c11 and "timber with 1-hour fire resistance" in c90_1. Relaxation was granted in c11 and c90_1, i.e., the interpretation of "non-combustible materials" was extended to include "hardwood" and "timber with 1-hour fire resistance". But "hardwood" and "timber with 1-hour fire resistance" cannot be represented as a subclass of "non-combustible materials" in the DKB since it is supposed that the DKB contains only the background domain knowledge which is well-established among experts, that is, not controversial and not debatable. The issue in c11 and c90_1 was how to interpret the concept "non-combustible materials" in the given context of c11 and c90_1. The rulings in these cases indicate that "hardwood" and "timber with 1-hour fire resistance" may be regarded as "non-combustible materials" mentioned in Regulation D7 only in the circumstances similar to those of c11 or c90_1. How to interpret "non-combustible materials" in a given context is reflected in the argumentation in those cases. In the current implementation, the system explains the arguments from players involved in debate and lets the user use their discretion in applying the interpretation of these concepts to specific situations. It would be possible to apply the interpretation reflected in the argument structure of previous similar cases to the input case automatically and we discuss a couple of plausible ways in Section 9.2.

8.6 Summary and Discussion

The system was first developed and designed with 6 imaginary cases. Then the system was implemented and tested with real-world data, i.e., 22 appeal cases provided by The Scottish Office. This chapter presented the results of the evaluation carried out with these 22 appeal cases. The system employs a case-based model of regulatory information and uses a case-based reasoning method to access and maintain regulatory information and cases. This case-based approach has been demonstrated to have the following advantages over a traditional rule-based approach in handling regulatory information.

First, the interpretation hierarchies which exist in regulatory information are more explicitly and conveniently represented as abstraction hierarchies of models. Furthermore, since models are loosely connected in the abstraction hierarchies and inconsistency among models is tolerated (to a limited extent), new regulatory information can be introduced more easily.

Second, although there is a scope of improvements in the matching algorithm (see Section 8.3.8), the representational framework, i.e., the abstraction hierarchies together with the matching algorithm, has the following advantages:

- It allows the accessing mechanism to be more tolerant towards the representation of the input case in two ways. First, terminological gaps between models and cases are bridged by using specialisation relations between terms used to describe models and cases. Second, in addition to the notion of structural mapping which has been widely used for similarity assessment [Gentner 83, Gentner & Landers 89, Holyoak & Thagard 89, Thagard & Holyoak 89, Branting 91], it makes use of domain-specific information to identify similarity between models and cases (implied similarity).
- It provides an efficient indexing mechanism by focusing on a particular group of models and pruning models unlikely to be relevant using background domain knowledge before the similarity assessment begins. However, the use of background domain knowledge could be a drawback because it limits the generality of the system and the system needs to provide extra facilities to acquire new background domain knowledge and to maintain consistency on the domain knowledge base.

The system was tested with 22 appeal cases but there are still over 160 appeal cases not tested and many more to come in the future. As the system moves towards the real-world environment, we need to ask the following questions: How can we make the system work with over hundreds or thousands of cases? In particular, how can we make the retrieval algorithm cover this size of case library?

The representational framework used in our system, i.e., the abstraction hierarchies and the matching algorithm, provides a way of coping with scale-up problems. Abstraction hierarchies allow regulatory information to be represented at different abstraction levels and the retrieval algorithm identifies the level of abstraction at which the input case is described. Only rulings made in cases in which relaxation is granted are integrated into the existing abstraction hierarchies by (if possible) combining rulings made in such cases with existing models. In other words, the abstraction hierarchies contain generalised cases (or model cases) each of which subsume one or more particular cases. Similar cases are found by comparing models at the right abstraction level of the abstraction hierarchies. This method reduces the number of models to be compared so that the retrieval in this method will be faster than retrieving relevant cases by comparing the input case to all cases in the Case Library when the number of cases in the Case Library increases well over the number of models in the Model Knowledge Base.

However, the retrieval algorithm in our system makes use of background domain knowledge to identify similarity. The experimental evaluation carried out with 22 appeal cases has exposed several problems which we have to face in scaling up the system. First, as pointed out in Section 8.3.8, we need more domain-specific contextual knowledge as the number of cases grows. The question is how much domain-specific contextual knowledge we need to give to the system. Second, as pointed out in Section 8.4.2, the tolerance of the matching algorithm has to be limited, and we have to decide how much tolerance should be allowed.

Chapter 9

Summary and Conclusion

9.1 Summary and Contributions

In this thesis, we proposed a unified case-based model of regulatory information (see Chapters 2–3) and presented a case-based reasoning system which uses this approach to access and maintain regulatory information and cases (see Chapters 3–7).

In this system, legal rules, i.e., the generalised cases, are represented as models; and interpretation hierarchies which exist among legal rules are represented as abstraction hierarchies of models in the Model Knowledge Base. Cases including argumentation surrounding each case are kept in a separate Case Library. An argumentation network which was developed from Toulmin's model and the IBIS model of argumentation is used to formulate and explain arguments among parties involved in debate. Background domain knowledge used in classifying input cases is represented as semantic networks and heuristic rules in the Domain Knowledge Base.

Given a case, the system first retrieves relevant regulatory information, i.e., models which are relevant to the input case and therefore with which the input case should be compliant. Once relevant models are found, the proposed design of the input case is evaluated against the requirements of the retrieved model(s). If the case is not compliant, there is a possibility of relaxation. The system does not make any decision about relaxation but only provides the users with relevant information—principles or issues behind the retrieved regulatory information and previous similar cases—so that they can arrive at an informed decision. If the user decides to grant relaxation, rulings made are integrated into the existing abstraction hierarchies by generalising rulings and (if possible) combining rulings with existing models.

A prototype system has been implemented in Prolog and evaluated with 22 appeal cases provided by The Scottish Office. In Chapter 8, we described the detailed evaluation results and also discussed scale-up problems.

As described in Section 1.3, there have been several knowledge-based systems in the domain of building regulations which address the issue of knowledge representation, such as Fenves *et al*'s systems using the SASE model [Fenves & Garrett 86, Fenves *et al.* 87, Hakim & Garrett 92], Stone and Wilcox's system [Stone & Wilcox 88] using the rule-based approach, and Waard's system [deWaard 92a, deWaard 92b] using Cornick *et al*'s model [Cornick & Leishman 91]. However, if we look at building regulations in the context of relaxation (see Section 2.1), this view of building regulations is too narrow and these systems have failed to capture the multidimensional and hierarchical nature of building regulations (see Section 1.3). This thesis proposed to employ a case-based model to capture such aspects of building regulations, i.e., to build abstraction hierarchies of building regulations and rulings made in relaxed cases, and this model has been successfully used to process 22 appeal cases as demonstrated in Chapter 8.

The case-based approach described in this thesis makes the following contributions towards the representation and maintenance of legal knowledge in general.

First, although abstraction hierarchies need to accommodate the representation and handling of exceptions and multiple subsumption as we discuss in the next section, it provides a representational framework to formulate principles (or legislative intentions or issues) behind statutes and precedents in case histories more explicitly and clearly.

Second, it allows both statutes and precedents set out in case histories to be integrated in one knowledge base. In previous approaches to knowledge representation of law such as in Gardner's program [von derLeith Gardner 87], CABARET [Rissland & Skalak 91], GREBE [Branting 91] and PROLEXS [Walker *et al.* 91], two different knowledge representation schemes have been used, i.e., statutes are represented using If-Then rules and a separate case-based reasoning system is linked to handle precedent. Our view is that statutory rules are the results of the accumulation and generalisation of precedents established in case histories, and this view has made it possible to represent statutes and precedents in case histories using one representation scheme, i.e., abstraction hierarchies of models.

Third, it provides a framework which can maintain changes in legal rules. As regulatory information develops in time, the texts of the statutory regulations can not cover all possible cases and interpretation of a particular regulation might change in time. The system described in this thesis maintains up-to-date interpretation of regulatory information by accumulating new rulings along with existing legal rules.

9.2 Possible Extensions

Representation and handling of exceptions

Exceptions to a legal rule are sometimes permitted for a particular category of buildings or in particular circumstances. For example, in Regulation D7 in [Sco81], the requirements that the floor of any landing or passage within a stairway enclosure be constructed of non-combustible materials are subject to exception if the building is a house or a flat or not more than three storeys in height. The abstraction hierarchies proposed in this thesis do not make this kind of information explicit. The structure of abstraction hierarchies could be modified so that information about exceptions can be represented explicitly and taken into account by the retrieval algorithm.

Multiple subsumption in abstraction hierarchies

Building regulations deal with various issues at different abstract levels. A more concrete issue related to a building component may also be related to several more abstract issues. For example, the size of windows is related to the issue of how to provide adequate ventilation for the health of people. The size of windows is also related to the issue of how to achieve the conservation of energy. In the model proposed in this thesis, subsumption relations between more abstract requirements and more detailed requirements are represented strictly hierarchically. In other words, two different requirements on the size of windows are represented in two different abstraction hierarchies, one in the abstraction hierarchy of health and the other in the abstraction hierarchy of energy conservation, and the existence of two different requirements on the size of window under different subject areas are only made explicit by the conflict checking procedure (see Section 7.3.1). It would be possible to incorporate multiple subsumption relations across different abstraction hierarchies so that all issues involved in a building component can be considered, not only the issue raised in the input case but also issues which are not raised explicitly but are related to the given case.

Analogy between cases

The retrieval procedure described in Chapter 6 retrieves previous similar cases by retrieving models relevant to the input case, and retrieved similar cases are explained to the user by simply describing information stored in the Case Library. But it would be possible to draw an analogy between the input case and retrieved similar cases to make clear the similarity and difference that exist between them. First, analogy between the general descriptions of the input case and a similar case could be drawn so that the similarity and difference among the key features of circumstances could be explained. Second, analogy between the argumentation structures of the input case and a similar case could be drawn so that more detailed context-sensitive issues involved in the debate could be compared and explained.

Decision-making facility

The system does not make any decision on relaxation but absolutely relies on the human expert's discretion. The system only provides relevant information so that the experts can make an informed decision. However, it might be possible to extend the system to give advice on relaxation in two ways.

First, it might be possible to project information given in an input case into the argumentation structure of similar cases. It would be possible to find similar features in the given case which correspond to features used to describe positions and arguments in a similar case. Projection of information can be done by replacing features used to describe arguments in a similar case with corresponding similar features in the given case. There might be information which can be projected into the argumentation structure and information which cannot be projected into the argumentation structure, but it would still be possible to see whether similar arguments can be made and therefore whether the same direction of argumentation can be taken in the given case.

The second possible approach is as follows. There is a set of factors related to each subject area, for example, in the subject of fire safety, there are several factors which are usually discussed such as: building type, occupant capacity, fire loading, number of storeys, fire-fighting access, alternative escape, etc. It might be possible to find typical patterns of argumentation, expressed in terms of those factors, in each subject area. Once these patterns of argumentation are formulated, it might be possible to apply this pattern to the input case and to predict the decision on relaxation which the human expert might make.

Concept learning facility

In the system described in this thesis, only background domain knowledge which is well-established among experts are represented using specialisation and aggregation hierarchies in the Domain Knowledge Base. But, as illustrated in Section 8.5, in some cases how to interpret a legal concept or a term in a given context is in debate and newly established definitions of such terms are left in the argumentation structures in the Case Library.

The definition recorded in the argumentation structures in such cases could be used to establish the definition of such terms. The argumentation structure recorded in cases in which relaxation was granted could be used as training cases in order to form the interpretation of such terms. As rulings related to the interpretation of such terms accumulate, more accurate and abstract description of such terms could be established. A new representational framework would be needed to represent and acquire such debatable legal concepts apart from specialisation and aggregation hierarchies.

Integration with CAD systems

In the system described in this thesis, regulatory information is used (passively) to ensure that the final design of the building to be constructed or altered satisfies the requirements set out in building regulations. The information about the building is provided by the user and the user makes sure that all essential information is provided. But, as explained in Section 6.4.2, it could be possible to link a CAD system (or a design system) in order to extract information which is necessary to check the conformance with building regulations. It could be possible to select models which contain requirements relevant to the building in question. Those selected requirements are design constraints specified in a limited set of features of buildings. Therefore it could be possible to extract design values related to those features from the design database in a CAD system and compare them with selected requirements to see whether the building is compliant with building regulations.

Regulatory information stored in the abstraction hierarchies could also be used actively by the designer during the design process. As explained in Section 1.3.1, design can be seen as a goal-oriented activity in which more abstract performance or function related goals are refined and decomposed into more detailed description until a final design is produced. From the early stages of design, the designer could use regulatory information represented as the abstraction hierarchies to identify semantics of parameters which they might give to individual building components, i.e., to identify what issues are related to individual building components and how these issues are related to each other, so that the designer could organise in a more meaningful order design problems and solutions produced during the design process.

9.3 Concluding Remarks

In this thesis, we proposed abstraction hierarchies of models as a representational framework to maintain regulatory information. The system makes the assumption that information stored in the abstraction hierarchies is incomplete, and provides a framework which can accumulate changes in the detailed and prescriptive requirements, i.e., requirements represented at the lower levels of abstraction hierarchies. Requirements stored in the abstraction hierarchies are normative information. In order to be able to incorporate the necessary changes, more normative information (represented as strong models) is distinguished from less normative and experiential information (represented as weak models). The necessary changes are made by adding or modifying weak models. In the long run, however, it is imaginable that normative information represented as strong models at every abstraction levels could be changed. How the framework proposed here could be further developed to cope with such changes is out of the scope of this thesis but might be worthy of further investigation.

The prototype system was tested with 22 appeal cases provided by the Scottish Office and the evaluation of the system was carried out only through a series of controlled experiments. The model of regulatory information is simple and the methodology proposed to process regulatory information seems intuitive. But only more extensive trials by users, i.e., experts working on the relaxation of building regulations, could tell us whether the system could be practical in the real world.

Bibliography

[Ashley 91a] Kevin D. Ashley. *Modelling legal argument: reasoning with cases and hypotheticals*. MIT Press, 1991.

[Ashley 91b] Kevin D. Ashley. Reasoning with cases and hypotheticals in HYPO. *International journal of man-machine studies*, 34:753–796, 1991.

[Begeman & Conklin 88] M. L. Begeman and J. Conklin. The Right Tool for the Job. *BYTE*, 13(10), October 1988.

[Bell 89] John Bell. Policy Arguments in Statutory Interpretation. In Jan van Dunne, editor, *Legal Reasoning and Statutory Interpretation*, pages 55–79. Gouda Quint BV Arnhem, 1989.

[Bell 91] John Bell. Policy Arguments in Legal Reasoning. Technical report, University of Leeds, July 1991.

[Brachmann 79] Ronald J. Brachmann. On the epistemological status of semantic networks. In Nicholas V. Findler, editor, *Associative networks: Representation and use of knowledge by computers*. Academic Press, 1979.

[Branting 91] L. Karl Branting. Building Explanations from Rules and Structureed Cases. *International journal of man-machine studies*, 34:797–837, 1991.

[Casson 93] Andrew Casson. Integrating Hypertext, Semantic Nets and Rule-Based Systems in an Expertext Shell for Authors and Readers of Regulatory Documents. In J. Mayfield and C. Nicholas, editors, *Proceedings of Workshop on Intelligent Hypertext, Arlington VA, USA*, November 1993.

[Collins & Quillian 72] A.M. Collins and M.R. Quillian. How to make a language user. In E. Tulving and W. Donaldson, editors, *Organisation of memory*. MIT Press, 1972.

[Collins 71] Peter Collins. *Architectural Judgement*. Faber and Faber, 1971.

[Conklin & Begeman 88] J. Conklin and M. L. Begeman. gIBIS: A Hypertext Tool for Exploratory Policy Discussion. *ACM Transactions on Office Information Systems*, 6(4):303–331, October 1988.

[Conklin 88] Jeff Conklin. Hypertext: An Introduction and Survey. In Irene Greif, editor, *Computer-supported cooperative Work: A Book of Readings*. Morgan Kaufmann Publishers, 1988.

[Cornick & Leishman 91] 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.

[Dayal *et al.* 93] Surendra Dayal, Michael Harmer, Peter Johnson, and David Mead. Beyond Knowledge Representation: Commercial Uses for Legal Knowledge Bases. In *Proceedings of the Fourth International Conference on AI and Law, Amsterdam, The Netherlands*, pages 167–174, June 1993.

[deWaard 92a] Marcel de Waard. Computer Aided Conformance Checking. In *Computers and Building Standards Workshop in Montreal, Canada*, May 1992.

[deWaard 92b] Marcel de Waard. *Computer Aided Conformance Checking*. Unpublished PhD thesis, Delft University of Technology, The Netherlands, 1992.

[Dworkin 86] Ronald M. Dworkin. *Law's Empire*. Fontana, 1986.

[Fenves & Garrett 86] Steven J. Fenves and James H. Garrett. Knowledge Based Standards Processing. *Artificial Intelligence*, 1(1):3–14, 1986.

[Fenves *et al.* 87] 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.

[Ganeshan *et al.* 92] 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.

[Gentner & Landers 89] 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.

[Gentner 83] Dedre Gentner. Structure Mapping: A Theoretical Framework for Analogy. *Cognitive Science*, 7:155–170, 1983.

[Gero & Rosenman 89] John S. Gero and Michael A. Rosenman. A Conceptual Framework for Knowledge-Based Design Research Sydney University's Design Computing Unit. In John S. Gero, editor, *AI in Design*. Springer-Verlag, 1989.

[Hakim & Garrett 92] 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.

[Her59] Her Majesty's Stationery Office, London. *The Building (Scotland) Act 1959*, 1959.

[Holyoak & Thagard 89] Keith J. Holyoak and Paul Thagard. Analogical Mapping by Constraint Satisfaction. *Cognitive Science*, 13:295–355, 1989.

[Keane 88] Mark Keane. Analogical Mechanisms. *Artificial Intelligence Review*, 2:229–250, 1988.

[Kolodner 93] Janet L. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, 1993.

[Kunz & Rittel 70] Werner Kunz and Hotst W.J. Rittel. Issues as Elements of Information Systems. Technical Report 131, Center for Planning and Development Research, University of California, Berkeley, CA, USA, 1970. Working Paper.

[Lawson 90] Bryan Lawson. *How Designers Think*. Butterworth Architecture, second edition, 1990.

[Levi 48] Edward Hirsch Levi. *An Introduction to Legal Reasoning*. University of Chicago Press, 1948.

[Logan & Newton 90] Brian Logan and Sidney Newton. Design by Decomposition: the role of expert systems in design support. Technical Report 484, Department of Artificial Intelligence, University of Edinburgh, 1990. DAI Research Paper.

[Logan & Smithers 89] 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.

[Logan 89] Brian Logan. Conceptualizing Design Knowledge. *Design Studies*, 10(3):188–195, 1989.

[MacGregor 91] Robert MacGregor. The evolving technology of classification-based knowledge representation systems. In John F. Sowa, editor, *Principles of Semantic Networks*, pages 385–400. Morgan Kaufmann, 1991.

[Maida & Shapiro 85] Anthony S. Maida and Stuart C. Shapiro. Intensional concepts in propositional semantic networks. In Ronald J. Brachmann and Hector J. Levesque, editors, *Readings in knowledge representation*, pages 97–118. Morgan Kaufmann, 1985. Also appeared in *Cognitive Science* 6(4), 1982, pages 291-330.

[McCormick & Bankowski 89] 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.

[McCormick 78] Neil McCormick. *Legal Reasoning and Legal Theory*. Clarendon Press, 1978.

[Nielsen 90] J. Nielsen. Hypertext. *Communications of the ACM*, 33(3), March 1990.

[Palmer 89] Stephen E. Palmer. Levels of Description in Information Processing Theories of Analogy. In Stella Vosniadou and Andrew Ortony, editors, *Similarity and Analogical Reasoning*, pages 332–345. Cambridge University Press, 1989.

[Quillian 85] M.R. Quillian. Word concepts: A theory and simulation of some basic semantic capabilities. In Ronald J. Brachmann and Hector J. Levesque, editors, *Readings in knowledge representation*, pages 191–215. Morgan Kaufmann, 1985. Also appeared in *Behavioral Science* 12, 1967, pages 410-430.

[Rissland & Skalak 91] Edwina L. Rissland and David B. Skalak. CABARET: Rule Interpretation in a Hybrid Architecture. *International journal of man-machine studies*, 34:839–887, 1991.

[Rissland 90] Edwina L. Rissland. Artificial Intelligence and Law. *The Yale Law Journal*, 99:1957–1981, 1990.

[Schank 72] Roger C. Schank. Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology*, 3(10), October 1972.

[Sco71] Scottish Office. *The Building Standards (Scotland) (Consolidation) Regulations 1971*, 1971.

[Sco81] Scottish Office. *The Building Standards (Scotland) Regulations 1981*, 1981.

[Sco90] Scottish Office. *Technical Standards for Compliance with the Building Standards (Scotland) Regulations 1990*, 1990.

[Sergot 85] Marek Sergot. Representing Legislation As Logic Programs. Technical report, Department of Computing, Imperial College of Science and Technology, 1985.

[Sergot *et al.* 86] 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.

[Shapiro 91] Stuart C. Shapiro. Cables, paths, and "subconscious" reasoning in propositional semantic networks. In John F. Sowa, editor, *Principles of Semantic Networks*, pages 137–156. Morgan Kaufmann, 1991.

[Simon 81] Herbert A. Simon. *The Sciences of the Artificial.* MIT Press, second edition, 1981.

[Soper & Bench-Capon 94] Paul Soper and Trevor Bench-Capon. Coupling Hypertext and Knowledge-Based Systems: Two Applications in the Legal Domain. *Artificial Intelligence and Law*, 2(4):293–314, 1993-1994.

[Sowa 84] John F. Sowa, editor. *Conceptual structures : information processing in mind and machine.* Addison-Wesley, 1984.

[Sowa 91] John F. Sowa. Toward the expressive power of natural language. In John F. Sowa, editor, *Principles of Semantic Networks*, pages 157–189. Morgan Kaufmann, 1991.

[Sterling & Shapiro 86] Leon Sterling and Ehud Shapiro. *The Art of Prolog.* The MIT Press, 1986.

[Stone & Wilcox 88] D. Stone and D.A. Wilcox. Intelligent Information Systems for Building Standards. In *Proceedings of EuropIA 88, Paris*, 1988.

[Susskind 87] Richard Susskind. *Expert Systems in Law.* Oxford University Press, 1987.

[Thagard & Holyoak 89] 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.

[Toulmin 58] Stephen Toulmin. *The uses of argument.* Cambridge University Press, 1958.

[Tweed 90] 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.

[von derLeith Gardner 87] Anne von der Leith Gardner. *An Artificial Intelligence Approach to Legal Reasoning.* MIT Press, 1987.

[Walker *et al.* 91] R. F. Walker, A. Oskamp, J. A. Schrickx, G. J. Van Opdorp, and P. H. Van Den Berg. PROPLEXS: Creating Law and Order in a Heterogeneous Domain. *International journal of man-machine studies*, 35:35–67, 1991.

[Waterman *et al.* 86] D. A. Waterman, J. Paul, and M. A. Peterson. Expert Systems for Legal Decision Making. *Expert Systems*, 3(4), 1986.

[Woods 85] William A. Woods. What's in a link: Foundations for semantic networks. In Ronald J. Brachmann and Hector J. Levesque, editors, *Readings in knowledge representation*, pages 217–241. Morgan Kaufmann, 1985. Also appeared in *Representation and Understanding: Studies in Cognitive Science*, edited by D.G. Bobrow and A.M. Collins, Academic Press, 1975, pages 35-82.

Appendix A

The summaries of 24 appeal cases

The summaries of 24 appeal cases provided by the Scottish Office are listed here. The following 14 items are used to summarise information in the case files of 24 selected appeal cases into the text form shown here.

CASE FILE NUMBER The number attached to the case file.

PURPOSE GROUP The purpose for which the building is built. For example, residential building, shop, office, theatre, gym, etc. Codes in the table to Schedule 3 of the 1990 version of the regulations are used in case summaries.

STATUS Whether the building is an existing building or a new building.

REG/VERSION The section of regulations and the year of legislation for which the applicant appealed. For example, K10/1981.

CLAUSE The clause(s) particularly relevant to the case.

SUBJECT This roughly corresponds to the headings of each part of regulations. For example, fire safety, convenience, health, etc.

ELEMENTS Building components related to the **ISSUE** described below.

RELATIONSHIP Whether the issue is about a relationship between two (or more) building elements.

DATE Approximate dates during which the case was processed.

ISSUE The description of proposed construction or alteration and issues.

ARGUMENTS Arguments from players.

DETERMINATION Decision of whether the BD granted or refused relaxation.

CONDITION Conditions attached to **DETERMINATION** if there are any.

REASONS The reason why relaxation was granted or refused.

CASE FILE NUMBER: 11
PURPOSE GROUP: 3 (offices)
STATUS: existing (listed)
REG/VERSION: D6/1971
CLAUSE: paragraphs (e) and (f)
SUBJECT: fire safety; non-combustible stairs
ELEMENTS: escape stair
RELATIONSHIP:
DATES: 19:03:73 to 26:07:73

ISSUE: The applicant is appealing for relaxation of regulation D6(e,f); "In every building the following shall be constructed of non-combustible materials...-(e) any stair forming part of an exit...not being a stairway wholly within a flat;-(f) the floor of any landing or passage within a stairway enclosure...". The applicant proposes to install hardwood escape stairs due to the lack of space of the small dwellings.

ARGUMENTS:

[architect] The physical difficulty and expense of providing a non-combustible stair is considered disproportionate to its effectiveness in such a tiny property. The proposed stair is of hardwood and as such will provide superior fire resistance and flame spread properties. Each room has a window overlooking the street through which an assisted escape could be made.

[local building control authority-COMMENT] All relevant documentation, drawings and correspondence are forwarded to BD.

[local building control authority] The local building control authority state that because the two timber stairways are to be removed and a new exit stair built it is reasonable to require that the new exit stair complies with the Building Standard Regulations in as far as reasonably practicable to do so. They go on to say that there doesn't appear to be any practical difficulty in forming a stair of non-combustible materials as the proposed timber stair. However, if the aesthetic considerations are so great then hardwood treads on top of a non-combustible structure could easily be provided.

[BD-COMMENT] BD propose to substitute for the local building control authority's decision a direction relaxing regulation D6. The local building control authority, applicant and firemaster are given 21 days to forward any comments they wish to make on BD's decision.

[BD] The hardwood stair proposed by the applicant will have the required fire resistance and will be enclosed; the number of occupants will be small and in the event of a fire breaking out in a room they are likely to escape well in advance of the fire penetrating to the stair itself.

[firemaster-COMMENT] The firemaster expresses surprise that after he had backed the local building control authority's decision to refuse relaxation BD granted relaxation. The firemaster hopes that BD will have second thoughts on the matter.

[local building control authority-COMMENT] The local building control authority also express their surprise and state that after a meeting of re-considering the matter in detail they could not accept the proposition that it is not "reasonably practicable" to form a non-combustible stair; the local building control authority are of the opinion that if it were practicable to form a hardwood stair, the formation of say a steel stair (with hardwood treads and risers on top of the steel mesh) within the same area should present no practical problems.

[BD-COMMENT] BD agree that such a stair would present no practical problem but such a stair would be a metal reinforced combustible stair, since the hardwood additions would be part of the stair and would have on balance no advantages over the stair proposed by the applicants as far as complying with the regulations. The construction of a concrete non-combustible stair does not appear to be reasonably practicable in this small building so that the stair proposed by the appellants seems to be the best solution. BD gives the local building control authority 14 days to forward any comments they wish to make on BD's decision.

[local building control authority-COMMENT] The local building control authority state that they agree with BD that a steel stair having hardwood treads and risers is no more than a "metal reinforced combustible stair" and state that a concrete stair in "situ" should be asked for. The local building control authority take the view that this would create no real problems particularly as it was the applicant's intention to build some new separating walls to enclose the stairway.

[BD-COMMENT] BD state that they hold the view that it is not reasonably practicable to erect the complicated shuttering which would be necessary to form a concrete stair in this comparatively small building.

DETERMINATION: Relaxation is granted.

REASONS: It is considered that, having regard to all the circumstances, it would be unreasonable to insist on full compliance with these provisions in this case involving a listed building.

CASE FILE NUMBER: 41
PURPOSE GROUP: 2B (institutional and other residential)
STATUS: existing
REG/VERSION: E9/1971
CLAUSE: paragraph (2)
SUBJECT: fire safety; enclosure of external fire escape stairs
ELEMENTS: external fire escape staircase
RELATIONSHIP:
DATES: 01:03:76 to 11:06:76

ISSUE: the applicant appeals for the relaxation of regulation E9(2); "The exit stairway or ramp shall be enclosed within a protected zone and except as aftermentioned or in a case falling within the provisos of D11 or D12 no other part of the building containing that stairway or ramp shall be enclosed within that protected zone". The applicant proposes to erect the exit stairways as unenclosed fire escape stairways at the rear of the building, (a) from 2nd floor to existing ground level, which is situated at approximately first floor level, (b) from 1st floor by bridge over kitchen yard to ground level which is situated at approximately first floor level. The escape routes are to be formed by open steelwork construction as follows; (a) from 2nd floor, providing escape from 6 no. bedrooms only, (b) from 1st floor, providing escape from lounge bar only.

ARGUMENTS:

[architect] The applicant makes his appeal on the following grounds; 1. The subjects are on an existing building; 2. The firemaster approves of suggested scheme; 3. The local building control authority's proposal would cause undue cost in the circumstances to the applicant; and 4. It is submitted that an unenclosed stair would be satisfactory.

[local building control authority-COMMENT] All relevant documentation, drawings and correspondence are forwarded to BD as complying with regulation 33 of the Building (Procedure) (Scotland) Regulations 1975.

[local building control authority] The local building control authority refuse relaxation because regulation E9(2) states that, "the exit stairway shall be enclosed within a protected zone ...". In this regulation a protected zone provided in accordance with this regulation is referred to as a 'stairway enclosure'. The local building control authority cannot recommend relaxation on this particular regulation for safety reasons, since the staircase involved is an external one and is therefore much more liable to suffer from atmospheric conditions and possible flame spread.

[BD-COMMENT] BD propose to substitute for the local building control authority's decision a direction relaxing regulation E9(2). "It has been agreed within the local building control authority that there is no contravention of regulation E9(2) in respect of the lower stair which links the ground and first storeys. Relaxation is not therefore required and it is proposed to take no further action on that part of the appeal". The local building control authority are given 21 days to forward any comment they wish to make on BD's decision.

[firemaster-COMMENT] The firemaster advises BD that he has no objection to the relaxing of regulation E9(2) provided the conditions stated are put into effect.

[local building control authority] The local building control authority state that they continue their objection to the proposed relaxation on the following grounds; 1. The proposal does not comply with the building regulations; 2. On the grounds of safety to the public who would use the stairway; 3. That it is considered practicable to enclose the stairway; and 4. That this view is supported by the fire department.

[BD-COMMENT] BD acknowledge the local building control authority's objections but state that the stair in question is being provided under the powers contained in the Fire Precautions Act 1971 and that it is not required by the building standards regulations. There is no power under the Building (Scotland) Act to insist on an existing sub-standard building being brought up to full regulation standard in a case where no alterations or extensions to the building are being made. The provision of the stair is therefore in the nature of a 'bonus' which will increase the safety of the occupants and in BD's view would thus merit some concession being given, namely, to allow the stair to be unenclosed.

DETERMINATION: Relaxation is granted

REASONS: It is considered that it would not be reasonably practicable to enclose the external stair in this existing building, as long as;

1. Any windows or glazed openings (other than windows in bathrooms, water-closets and washrooms) directly below or within 3 metres of any part of the said stairway shall be glazed with wired glass not less than 6 mm thick and shall not be openable;
2. Any door, except the topmost door, which is directly below or within 3 metres of the said stairway shall be self-closing and shall have a period of fire resistance of not less than one-half hour.

CASE FILE NUMBER: 42
PURPOSE GROUP: 1B (dwellings)
STATUS: existing
REG/VERSION: K15/1971
CLAUSE:
SUBJECT: ventilation; window ventilation
ELEMENTS: windows
RELATIONSHIP:
DATES: 30:03:76 to 10:06:76

ISSUE: The applicant appeals for relaxation of regulation K15; "Every window, ventilator, permanent ventilator and permanent ventilation opening provided so as to comply with this Part shall be so positioned that the top of the opening part or of the permanent ventilation opening is not less than 2.0 metres above the floor....". The applicant proposes to construct a staircase to an attic and square off half the roof to the height of 2.070 metres which he believes is quite adequate for a bedroom. He appeals the regulation to allow the opening parts of the gable bedroom windows to be 1.750 metres from the floor.

ARGUMENTS:

[applicant] The applicant needs the extra bedroom space for his growing family. The relaxation needs to be granted in order that the new dormer does not rise above the ridge line of the existing roof and so that the existing dormer windows to the rear of the house can be used.

[local building control authority] The requirement of this regulation that every window, ventilator, permanent ventilator and permanent ventilation opening provided so as to comply shall be so positioned that the top of the opening part or of the permanent ventilation opening is not less than 2.0 metres above the floor. It is considered by the local building control authority that the proposals fall too short of the requirements of the regulation and would reduce housing standards to an unacceptable level.

[local building control authority-COMMENT] All relevant documentation, drawings and correspondence are forwarded to BD as complying with regulation 33 of the Building (Procedure) (Scotland) 1975.

[BD] BD propose to substitute for the local building control authority's decision a direction granting relaxation of regulation K15. BD grant the local building control authority and applicant 21 days to comment on DD's decision.

DETERMINATION: Relaxation is granted.

REASONS: The relaxation shall apply to allow the tops of the opening parts of the two existing windows to the rear of the first storey to remain at their existing height of 1.750 metres above the floor. It is considered that compliance would not be reasonably practicable in the particular circumstances obtained in this case.

CASE FILE NUMBER: 77
PURPOSE GROUP: 1C (dwellings)
STATUS: existing
REG/VERSION: S3/1971
CLAUSE: paragraph (1)
SUBJECT: safety; headroom of stairs
ELEMENTS: stair
RELATIONSHIP:
DATES: 12:04:79 to 10:08:79

ISSUE: The applicant does not feel it is reasonably practical to reconstruct the staircase dormer extension so as to comply with building regulation S3(1) concerning the minimum clear headroom requirement of 2050 mm. Only two of the treads infringe the regulation everything else on the staircase and the rest of the alterations comply with the regulations. The staircase is only used infrequently to access a parent's bedroom and is perfectly safe and comfortable in the applicant's eyes.

ARGUMENTS:

[applicant] To comply with building regulation S3(1) the applicant would have to reconstruct the staircase dormer nearer than one metre to the gable boundary wall. This would result in contradicting regulation D17(5) and directly oppose the requirements of the planning department for the dormers to be kept at least one metre in from the front and sides of the existing building.

[local building control authority] As this is a new stair it is considered that it should comply with Building Standards regulations S3(1). An accurate survey of the property at the time of preparation of plans should have highlighted the difficulties likely to be encountered on site. An approach to the local building control authority at an earlier stage could have resulted in a deviation acceptable to both the planning and building authorities.

[local building control authority-COMMENT] The local building control authority send all relevant drawings and correspondence to BD as stated in regulation 33(1) of the Building (Procedure) (Scotland) Regulations 1975.

[BD] BD propose to uphold the local building control authority's decision to refuse the appeal on the grounds that the stair being a new stair should comply with the regulations. It is considered that there has been advanced no good reason related to the purpose of this regulation which would justify granting this appeal.

[BD-COMMENT] Any comments which applicant may have on this draft determination should be sent in writing to BD within 21 days.

[applicant] The applicant admits to making an error in changing the position of the dormer haffit providing the headroom to the staircase in question. When faced with the structural problem with the existing rafters the architect decided to change the position of the dormer, quite innocently, never realising the implications of doing so. Having admitted the error the applicant asks BD for some measure of reasonableness to be taken into account and reverse the draft decision.

DETERMINATION: Relaxation of regulation S3(1) is refused.

REASONS: The regulation provides for a minimum height of 2050 mm over the whole width of a stair whereas in this case there is a reduction of 210 mm at it's center line. The center line is taken as the most reasonable part of measurement because of the slope of the roof affecting almost the entire width of the stair. Clearly the infringement over half the width of the stair is much greater and to a diminishing extent over the remainder. Even had there been the greatest difficulty at the outset such diminution of the headroom is totally unacceptable for purposes of the regulation. BD would not agree that the spirit of the regulation had been met; furthermore there is likely to be a succession of occupiers, the convenience of whom must also be secured.

CASE FILE NUMBER: 82
PURPOSE GROUP: 1B (dwellings)
STATUS: existing and change of use
REG/VERSION: D6/1971
CLAUSE: paragraph (b)
SUBJECT: fire safety; separating walls
ELEMENTS: separating walls
RELATIONSHIP:
DATES: 12:07:79 to 05:09:79

ISSUE: The applicant appeals for relaxation of regulation D6(b); "In every building the following shall be constructed of non-combustible materials...-(b) any separating wall...". To replace the loadbearing partition and construct a non-combustible would mean temporarily supporting the first floor, cutting flooring and joisting of ground floor, internal excavation and strip foundation laying and all while the owner is still occupying the house. The applicant constructed an alternative type of separating wall to that described in the building warrant.

ARGUMENTS:

[applicant] If a non-combustible wall was constructed the exit stairway width would be effectively reduced from 900 mm to approximately 856 mm due to the necessary increase in thickness of wall due to wall strapping and finishes. If the applicant took the local building control authority's advice and construct a new wall alongside that built would effectively reduce the corridor width to 806 mm proving a further obstacle and complication in the moving of furniture from the living room to other parts of the house.

[local building control authority] The local building control authority refuses relaxation on the following grounds;

1. the "load bearing wall" is not really load bearing, in that it only supports a partition wall at first floor level;
2. the light weight concrete block fire proof wall complete, would probably be cheaper to construct than the fire resistant structure;
3. if the separating wall is constructed as approved, the stairway width will not be affected at all. The wall would be an internal wall and can be plastered directly onto the block;
4. it was never suggested that a new wall be constructed alongside that built thereby reducing the width of the corridor between the living room and other parts of the house on the ground floor.

The local building control authority are of the opinion that there was no practical difficulty in constructing the separating wall in accordance with the terms of the Building Warrant granted.

[local building control authority-COMMENT] All relevant documentation, drawings and correspondence are forwarded to BD as complying with regulation 33 of the Building (Procedure) (Scotland) Regulations 1975.

[local MP-COMMENT] Local MP takes up case of applicant and writes to complain to BD about the length of time of the proceedings and hopes that BD can resolve the matter satisfactorily.

[BD-COMMENT] BD propose to substitute for the local building control authority's decision a direction relaxing regulation D6(b). The firemaster, local building control authority and applicant are given 21 days to forward any comments they wish to make on BD's decision.

[firemaster-COMMENT] The firemaster confirms that he has no representation to make against the relaxation of regulation D6(b) being granted on the specified terms.

[local building control authority-COMMENT] The local building control authority strongly object to the retrospective granting of a relaxation when, if the wall had been built as required, no "disproportionate amount of work would have been required". They go on to say that the granting of this application sets a precedent for developers openly flouting the terms of their Building Warrant and proceeding with substandard construction knowing that any necessary relaxation will be granted by the BD.

[BD-COMMENT] BD comment that the appeal was considered in relation to the circumstances which existed before any work commenced and the fact that work had already been carried out in no way influenced the decision.

DETERMINATION: Relaxation is granted.

REASONS: BD consider that a disproportionate amount of work would be required to construct a non-combustible wall and that compliance with the requirement of this regulation would not therefore be reasonably practicable in the particular circumstances of this case.

CASE FILE NUMBER: 85
PURPOSE GROUP: 3 (offices)
STATUS: existing
REG/VERSION: E9/1971
CLAUSE: paragraph (5)
SUBJECT: fire safety; escape routes
ELEMENTS: escape route
RELATIONSHIP: between exit opening out or leading to 'place of safety'
DATES: 25:07:79 to 21:04:80

ISSUE: The applicant appeals for relaxation of regulation E9(5); "Every stairway enclosure shall give access at ground level to an exit to the open air....". In the interests of safety the applicant decided to provide an additional exit rather than to apply for a relaxation on travel distance. The exit to the rear passageway was therefore reduced, this being the only position acceptable to the Planning Authority and the Historic Building Council for a new opening in this 'A' listed building. The passageway complies with the regulations in every aspect except that it may be considered not to be at ground level. However, with regard to this side of the building it is indeed at ground level due to the slope of the site. The case for relaxation is therefore based on whether or not the passageway is accepted as being at ground level in relation to the exit point.

ARGUMENTS:

[local building control authority] The local building control authority refuse relaxation because the proposal to discharge from the first floor to the rear passageway and embankment is unacceptable as these areas are not considered 'places of safety' in terms of the regulations. The regulations require exits to lead directly to a 'place of safety'.

[local building control authority-COMMENT] All relevant documentation, drawings and proposals are forwarded to BD as complying with regulation 33 of the Building (Procedure) (Scotland) Regulations 1975.

[BD-COMMENT] BD propose to substitute for the local building control authority's decision a direction relaxing the regulation E9(5) under the specified terms. BDgives the firemaster, applicant and local building control authority 21 days to forward any comments they wish to make on BD's decision.

[local building control authority-COMMENT] The local building control authority consider that BD's proposed grant of relaxation is not at all appropriate in the circumstances, and ask that before approval is issued, a meeting be held on site between a suitable representative of BD and a chief building control officer from the local building control authority, so that the full matter of the proposal can be examined thoroughly.

[firemaster-COMMENT] The firemaster comments that the footpath at first floor level is not suitable for a means of escape because in one direction it leads past the unprotected windows of the building it affords egress from and in the other direction it gives access to an embankment which is not considered suitable for a means of escape.

[BD-COMMENT] BD in reply to the firemaster's and local building control authority's comments state that given the choice of exits available and, to some extent, the constraints imposed by the existing building, relaxation is considered to be justified. The firemaster's comments about the footpath have been considered but although BD agree that it leads past certain windows within the same unit at first storey level this is a feature which could occur within an enclosed place of safety. The width of the footpath, measured between the building and the face of the retaining wall is 1.32 metres at its narrowist point and 1.52 metres at its widest. The footpath therefore complies with the exit width requirement of 1.1 metres.

DETERMINATION: Relaxation is granted.

REASONS: Relaxation shall apply only to allow a stairway enclosure to give access at first storey level to a footpath in the open air having the width required by regulation E7, the said footpath in turn leading to a 'place of safety'. It is considered that this arrangement will adequately fulfill the purpose of the respective requirements.

CASE FILE NUMBER: 87
PURPOSE GROUP: 5B (assembly and recreational)
STATUS: existing
REG/VERSION: K10/1971
CLAUSE: paragraph 4(b)(ii)
SUBJECT: ventilation; mechanical ventilation
ELEMENTS: extract fan
RELATIONSHIP:
DATES: 26:09:79 to 22:04:80

ISSUE: As a consequence of building a proposed extension across the ventilating windows of the existing church hall, the ventilation of that hall would be even more deficient than it was previously. The architect agreed and indicated, as an amendment to their proposals, that two mechanical extract fans would be installed in the existing hall to make good the loss of ventilating windows. The work is completed with the exception of the installation of these two fans. The applicant now seeks relaxation of regulation K10 4(b)ii; "...where the cubic space per occupant exceeds 2.8 cubic metres but does not exceed 21 cubic metres by mechanical means to provide a fresh air supply at the rate set out in Table 12...". (ref. Table 12 1971-'79 regulations, showing rate of supply in cubic metres of fresh air per person per hour). The applicant is appealing to allow the fans and inlet vents to be omitted from the works saying that he feels the ventilation is more than adequate.

ARGUMENTS:

[applicant] The existing hall has operated very satisfactorily, on no occasion has it ever been necessary to open all the windows. This may be due to several factors; the draughty nature of the building; its exposed position; the extra air space above the stage and cupboards; and the number of doors (three single and one double internal, one double external fire door and a service hatch).

[local building control authority] The local building control authority state that the appeal should be denied on the grounds that no particular difficulties would be encountered in complying with the regulations.

[local building control authority-COMMENT] Papers falling under reg. 33 of Building (Procedure) (Scotland) Regulations 1975 forwarded to BD.

[BD-COMMENT] Applicant is asked to supply additional information in order that due consideration can be given to the appeal;

1. the floor area of the existing hall, and
2. the aggregate area of the opening parts of the windows which are to remain.

[applicant-COMMENT] The applicant forwards all relevant information and adds that over a period of almost two years, never have all the windows been open. During this time all possible events which can take place in the hall have taken place, e.g busy,smokey concerts, sales of work etc. The applicant states he is unable to foresee any event which would induce him to make use of the fans should their installation be insisted upon.

[BD] BD propose to uphold the local building control authority's decision to refuse the relaxation of regulation K10 because they consider that the standard set out in the regulations is wholly appropriate and that there would be no undue difficulty in complying with it.

[BD-COMMENT] Any comments which the applicant may have on the draft determination should be sent in writing to BD within 21 days.

[applicant-COMMENT] The applicant expresses his disappointment and goes on to describe the regulation's requirements as excessively strict and oppressive.

DETERMINATION: Relaxation is refused.

REASONS: It is quite evident that it is not a case of there being adequate ventilation in the existing hall but one of a diminution to an unacceptable degree of the standard which existed before alterations were made to the building. Provision is made to allow the standard of ventilation of the hall to remain as it was before the erection of the extension but only where the building as extended or altered would not fail to conform to the regulations a greater degree than it did at the time application for building warrant was made. At that time the local building control authority drew attention to the adverse effect the extension would have on the standard of ventilation in the existing hall. To overcome the local building control authority's objection to the granting of building warrant it was agreed with the architect to amend the proposed works by installing two mechanical extract fans in the existing hall in order to compensate for the consequential loss of ventilating windows.

CASE FILE NUMBER: 90
PURPOSE GROUP: 2A (institutional and other residential)
STATUS: existing
REG/VERSION: D6/1971
CLAUSE: paragraphs (c), (e) and (f)
SUBJECT: fire safety; escape stairs
ELEMENTS: escape stair
RELATIONSHIP:
DATES: 11:07:80 to 10:09:80

ISSUE: The applicant is appealing for relaxation of regulation D6(c,e,f);" In every building the following shall be constructed of non-combustible materials ...-(c) any fire division wall which forms a stairway enclosure...or which separates a lift shaft from the remainder of the building..;..-(e) any stair forming part of an exit...not being a stairway wholly within a flat;-(f) the floor of any landing or passage within a stairway enclosure...". The applicant's existing 3 storey building has only one means of escape from the first and top floors. He proposes, voluntarily, to attempt to improve the situation by proposing a new concrete exit stair from first to ground floor level. For space reasons it is impractical to extend the stair to the top floor therefore the applicant proposes to construct a second exit stair of timber construction with surrounding one hour minimum fire resistant partitions from the top floor to ground.

ARGUMENTS:

[architect] Over the six or seven years that the applicant has owned these premises he has provided a fire alarm system, smoke detectors and a 'tele-nurse' system of intercommunication between every bedroom and the duty sisters room. The scheme also includes provision of a fire screen around the existing exit stair from the top floor, additional fire doors and another exit from the ground floor level.

[local building control authority-COMMENT] All relevant documentation, drawings and correspondence are forwarded to BD as complying with regulation 25 of the Building (Procedure) (Scotland) Regulations 1975.

[local building control authority] The local building control authority refuse relaxation on the grounds that the exit stairway should be positioned in another location which could comply with the regulations.

[architect] The position of the exit stairways are the most practical without losing valuable bedroom accommodation. The applicant is voluntarily expending a lot of money on this scheme and cannot afford the loss of income that the loss of any bedrooms would incur.

[BD-COMMENT] BD propose to substitute for the local building control authority's decision a direction granting relaxation of regulation D6. BD allows the firemaster, local building control authority and applicant 21 days to forward any comments they wish to make on BD's decision.

[firemaster-COMMENT] The firemaster confirms that no representation with respect to BD's decision will be made.

[local building control authority-COMMENT] The local building control authority inform BD that they have no comments to make on the proposed relaxation as submitted.

DETERMINATION: Relaxation is granted.

REASONS: It is considered that full compliance, with the requirements of paragraphs (c,e,f) of regulation D6 as to the non-combustibility of the fire division walls forming the enclosure of the new exit stair and the floors of landings or passages within the stairway enclosure leading from the second storey to the first storey of the building, will not be reasonably practicable in this existing building.

CASE FILE NUMBER: 93
PURPOSE GROUP: 1B (dwellings)
STATUS: existing
REG/VERSION: K15/1971
CLAUSE: (1971 to 1980 Regs.)
SUBJECT: ventilation; window ventilation
ELEMENTS: windows
RELATIONSHIP:
DATES: 23:08:81 to 09:12:81

ISSUE: Allow tops of the openings, of 3 bedroom windows, to be 1.8 metres above floor instead of at least 2 metres.

ARGUMENTS:

[applicant] Proposal for 3 new bedroom windows to be installed into existing eighteenth century house. The tops of the windows are at a height of 1.8 metres above the floor and the opening areas of the windows are sufficient to meet the requirements of the regulations.

[local building control authority] No good reason has been advanced to justify relaxation of the requirements of this regulation.

[local building control authority-COMMENT] BD receive all relevant drawings and correspondence according to the Building Regulations as complying with regulation 33 of the Building (Procedure) (Scotland) Regulations 1975.

[BD] BD notify building control authority that they are to relax regulation K15. BD allow building control authority 2 weeks to comment on proposed decision.

[local building control authority] The local building control authority consider that it is possible and reasonable to meet the standards and to relax the general requirement of reg. K15, particularly in sleeping rooms, is contrary to the principal of providing suitable conditions in which to live which in this case would be exacerbated by the deficiency in the area of opening provided by the existing windows.

DETERMINATION: Regulation is relaxed.

REASONS: It would not be reasonably practicable to raise the height of the height of the top of the window opening part.

CASE FILE NUMBER: 95
PURPOSE GROUP: 5A (assembly and recreational)
STATUS: existing and change of use
REG/VERSION: E6/1981
CLAUSE:
SUBJECT: fire safety; the number of fire exits
ELEMENTS: fire exits
RELATIONSHIP:
DATES: 25:11:81 to 25:03:82

ISSUE: The applicant proposes to refurbish an existing retail shop and develop part of it into a games room. The applicant is appealing for relaxation of regulation E6 regarding the number of exits necessary for a building warrant.

ARGUMENTS:

[applicant] By classing the small games room as an amusement arcade the local building control authority place the premises under excessive regulation requirements than are necessary. Being eleven feet away from the main entrance the games room does not need an extra exit as precautions taken are adequate enough.

[local building control authority-COMMENT] Copies of the application for relaxation and relevant correspondence and drawings are sent to BD as complying with regulation 25 of the Building (Procedure) (Scotland) Regulations 1981.

[local building control authority] The local building control authority refuse relaxation because;

1. The only exit from the amusement area is through the retail shop premises;
2. It is possible to provide a second exit from the amusement area to the rear as a bricked up opening exists at the moment;
3. The amusement area of the premises is capable of holding a reasonably large number of people therefore, maintaining a second exit is preferable;

[BD-COMMENT] BD send copies of relevant documents to the firemaster and inform him that they propose to substitute for the local building control authority's decision a directive granting relaxation of regulation E6. The firemaster and local building control authority are allowed 21 days to forward any comments they wish to make on BD's decision.

[local building control authority-COMMENT] The local building control authority appeal for another week thus enabling the building control sub-committee to consider BD's draft decision.

[firemaster-COMMENT] The firemaster does not intend to make representation with respect to the decision set out in BD's draft.

DETERMINATION: Relaxation is granted.

REASONS: Whilst it is noted that on the basis of a classification of an amusement arcade occupancy sub group two exits are required, the combined occupant capacity of the shop and games room, calculated by occupant load factors which may not be wholly appropriate in this case, is not greatly in excess of 60, for which number one exit is acceptable. Since it is impracticable to form an exit at the rear of the premises the only feasible location for an additional exit is the front wall of the games room, creating additional work and cost disproportionate to the benefit to be obtained from such a small gain in travel distance. It is considered that, taken together, these factors justify the granting of relaxation.

CASE FILE NUMBER: 97
PURPOSE GROUP: 7A (storage and warehousing)
STATUS: existing
REG/VERSION: S5/1981
CLAUSE: paragraph (6)
SUBJECT: safety; guarding of balustrades
ELEMENTS: stair
RELATIONSHIP:
DATES: 21:06:82 to 07:12:82

ISSUE: The applicant appeals for relaxation of regulation S5(6); "No opening in any balustrade or between any railings....shall be of such a size as will permit the passage through it of a sphere 100 mm in diameter". In order to achieve a new fire certificate the local building control authority advised that stairways and enclosures be constructed. Having open timber slatted floors the applicant proposes that the warehouse buildings be classified as single storey and therefore not require stairway enclosures.

ARGUMENTS:

[applicant] As the existing floors are of open slatted timber flooring supported on timber joists, it would appear unreasonable to require new stairways to be constructed in materials other than timber. Only a small number of distillery personnel, who are fully conversant with the layout of the buildings, will be employed within same at any one time, and these periods will be of an intermittent nature. Also, as the buildings are under Customs and Excise jurisdiction, no unauthorised persons are allowed to enter the premises. It appears unreasonable for the company to incur the additional costs of the construction of stairways and enclosures in a building where occupancy is of an occasional nature, of short duration and only involving on average, four persons.

[local building control authority-COMMENT] Under the terms of regulation 20 of the Procedure Regulation the draft decisions were sent to the architect and the firemaster. After the allotted 14 days no representations objecting to the draft decisions were received.

[BD-COMMENT] BD propose to substitute for the local building control authority's decision a direction relaxing regulation S5 with reference to the three bonded warehouse buildings (2,5 and 6).

[BD-COMMENT] BD notify the local building control authority and the firemaster and give them 21 days to forward any comments on the draft decision.

[firemaster-COMMENT] The firemaster informs BD that he has no comments to make regarding the draft proposal.

[applicant-COMMENT] Applicant asks BD why relaxation has been restricted to warehouses 2,5 and 6 and not 1,3 and 4.

[BD-COMMENT] BD state that warehouses 1,3 and 4 do not satisfy other regulations relating to 'escape from fire' regulations which have to be brought into consideration at another time.

DETERMINATION: Relaxation is granted.

REASONS: Because the definition of a storey of a building (reg. A3(9)) any openwork floor is taken to be part of the storey in which it is situated. Therefore the upper floors being timber slatted openwork floors, it follows that each of the stairways do not form part of an exit therefore regulation 9 does not affect them, therefore enclosures don't have to be built. It is considered that each of the balustrades guarding landings of stairways in each of the bonded warehouse buildings (2,5 and 6) shall consist of an upper horizontal rail and a mid-level rail together with uprights required for strength and stability.

CASE FILE NUMBER: 110
PURPOSE GROUP: 4 (shops and commercial)
STATUS: existing
REG/VERSION: D4/1981
CLAUSE: paragraph (1) to (5)
SUBJECT: fire safety; places of special fire risk
ELEMENTS: places of special fire risk
RELATIONSHIP: interpretation of high or low fire risk
DATES: 23:09:85 to 09:12:85

ISSUE: The applicant appeals for relaxation of regulation D4. the applicant proposes to install a gas fired warm air heater with rating in excess of 60 KW. The appeal is based on the fact that the applicant feels the local building control authority's interpretation of the regulation is too onerous.

ARGUMENTS:

[applicant] The regulation states a place of special fire risk as being any place within a building in which one or more heat producing appliances of a certain minimum capacity are installed. This infers that the place of special fire risk exists. The applicant's contention is that the regulation relates to places, by nature of the content or occupation does not imply that the installation alone creates the place of "special fire risk" and that the applicant's premises do not fall within this category.

[local building control authority] The local building control authority refuse relaxation on the following grounds pulled from regulation D4. "Where there is installed one or more heat producing appliances having a total installed rating exceeding, in the case of gas burning appliances, 60 KW input the appliance shall be separated from the remainder of the building by a wall or floor or both which shall have a fire resistance for a period not less than one hour". They go on to say that they consider, in this instance, there is no reason why this regulation cannot be complied with in full.

[local building control authority-COMMENT] All relevant drawings, documents and correspondence are forwarded to BD as complying with regulation 25 of the Building (Procedure) (Scotland) Regulations 1981.

[BD-COMMENT] BD propose to substitute for the local building control authority's decision a direction relaxing regulation D4. The local building control authority are given 14 days to make any comment on BD's decision.

[firemaster-COMMENT] The firemaster confirms that he has no representation to make on the matter.

DETERMINATION: Relaxation is granted.

REASONS: Relaxation is considered to be reasonable in view of the low fire risk associated with the proposed gas-fired appliance.

CASE FILE NUMBER: 116
PURPOSE GROUP: 5A (assembly and recreation)
STATUS: existing and change of use
REG/VERSION: T3/1981
CLAUSE: paragraph (3)
SUBJECT: convenience; sanitary facilities for the disabled
ELEMENTS: facilities for the disabled
RELATIONSHIP: disabled watercloset/ambulant disabled conversion
DATES: 02:07:86 to 19:01:87

ISSUE: The applicant states that his clients intend to reorganise the layout of their shop to provide a small coffee shop to try and utilise his business and encourage additional business. The proposed coffee shop area has a floor at a level of five steps down from the main shop floor level and main entrance. There is a second exit from the lower level which gives access to a small street with a very narrow footpath not giving enough width to satisfactorily manoeuvre a wheelchair. If this second door was used for a disabled persons entrance, there is insufficient space within the coffee shop for a wheelchair to move. There is simply insufficient space to enable a further toilet, for disabled persons or otherwise to be formed and still leave an area in the coffee shop which would accommodate approximately 20 persons. This is the reason for the appeal for relaxation of regulation T3(3);"Where sanitary conveniences are provided on a floor level to, which this regulation applies, a reasonable number of conveniences (and never less than one) shall be accessible to and designed to meet the needs of disabled persons".

ARGUMENTS:

[applicant] The desired floor area of the coffee shop would accommodate 20 persons but if this number would be reduced to 14 or 15, with the introduction of a disabled toilet the scheme would not be financially viable. There is a large cafe/restaurant sited close by giving easy access for disabled persons at street level which would encourage disabled persons more than the lower level premises. If relaxation is granted the owners will provide grab rails etc. to existing toilet to assist the ambulant disabled.

[local building control authority-COMMENT] The local building control authority forward all relevant drawings and correspondence to BD as complying with regulation 25 of the Building (Procedure) (Scotland) Regulations 1981.

[BD-COMMENT] BD propose to substitute for the local building control authority's decision a direction to grant the relaxation of regulation T3(3). The local building control authority is given 14 days to comment on BD's decision.

[local building control authority-COMMENT] The local building control authority agree with BD's decision but are concerned that the draft directive gives little or no guidance on the criteria on which the decision was based. They enquire if guidance could be made available to them on BD's most up to date decisions on Part T.

[BD-COMMENT] Each case is dealt with on merit and it would be difficult for BD to make available to the local building control authority guidance based on Part T decisions made by BD. Some general guidance was offered by BD in Building Control Note No. 1/85 but the Part T consultation paper issued on 30:06:86 should be more helpful. The paper suggests that a concession be allowed for shops and offices of less than 100 sq. metres to allow ambulant disabled, rather than a full disabled watercloset. In this case the premises well below that limit.

DETERMINATION: Relaxation is granted with the condition that at least one watercloset compartment shall be provided for the use of ambulant disabled persons.

REASONS: With reference to the Part T consultation paper (mentioned above) it was considered that full compliance with this regulation is not reasonably practical in the circumstances of this case.

CASE FILE NUMBER: 131
PURPOSE GROUP: 1C (dwellings)
STATUS: existing
REG/VERSION: S3/1981
CLAUSE: paragraph (6)
SUBJECT: safety; guarding of external stairs
ELEMENTS: stair
RELATIONSHIP:
DATES: 26:11:87 to 06:01:89

ISSUE: The client would consider installation of a simple upright and rope rather than the local building control authority's directive to enclose the external stair and landing with a balustrade. The client appeals to BD to relax regulation S3(6); "Every stair which rises more than 600 mm above an adjacent floor or landing or above ground external to a building shall be guarded on each side by a wall or by a secure balustrade or railing ...".

ARGUMENTS:

[applicant] The external stair is no longer in use so there should be no need for a balustrade rail. They suggest erecting a simple upright and rope detail. They feel that by enclosing the steps, which lead to a blocked up doorway, with a balustrade would have a detrimental effect on the appearance of the building. It is clear from the structure that no balustrade rail detail has existed in the past.

[local building control authority-COMMENT] The local building control authority send all the relevant drawings and correspondence to BD as complying with regulation 25 of the Building (Procedure) (Scotland) Regulations 1981.

[BD] The architect has waited too long to appeal against the local building control authority's decision. Regulation 23 of Building (Procedure) (Scotland) Regulations 1981 require that notice of an appeal must be given in proper form within 28 days of the local building control authority's decision. BD have the power to allow a longer period of time if there are special circumstances which excuse the delay in giving notice, however, none have been forwarded.

[architect] An extended period of illness and holidays resulted in the delay of the appeal and it was 3 months before further action could be taken.

[BD] BD accept appeal.

[local building control authority] The regulations exist to ensure the safety of persons occupying or having access to buildings and the protection of stairs is an important element of the regulations. Although the stair does not lead to a door, it is the case that the window at the top of the stair is constructed to have the appearance of a door, and it is therefore likely that the stair will be used by visitors unfamiliar with the building who wish to gain access to it.

[BD] BD propose to substitute for the local building control authority's refusal to grant relaxation saying that the stair and landing were not included in the alteration work and as a result of that work neither the stair nor the landing fail to conform to regulations.

[BD] The local building control authority are allowed 14 days to forward any comments on BD's decision.

DETERMINATION: The regulation is relaxed.

REASONS: The stairway and landing in question are dis-used and do not give access to any part of the building; and they were not provided with a means of guarding before the alteration work was carried out and are not part of that work.

CASE FILE NUMBER: 132
PURPOSE GROUP: 5A (assembly and recreation)
STATUS: existing
REG/VERSION: T3/1981
CLAUSE: paragraph (3)
SUBJECT: convenience; sanitary facilities for the disabled
ELEMENTS: facilities for the disabled
RELATIONSHIP: disabled watercloset/ ambulant disabled conversion
DATES: 15:02:88 to 15:07:88

ISSUE: The applicant is appealing for relaxation of regulation T3(3); "Where sanitary conveniences are provided on a floor level, to which this regulation applies, a reasonable number of conveniences (and never less than one) shall be accessible to and designed to meet the needs of disabled persons". The applicant proposes to alter the lounge bar of his premises by reducing the floor area of the gent's toilet facilities in order to increase the floor area of the lounge bar. It should be stressed that the existing number of sanitary fittings within the gent's toilet would remain unaltered. The floor area of the toilet facility would be reduced to allow the floor area of the lounge bar to increase from 29.15 sq. metres to 34.75 sq. metres (19.2 These proposals, however, cannot proceed until they comply with Part E and Part T of the Building Standards (Scotland) Regulations 1981. The proposals can be amended to comply with Part E but the provision of a disabled toilet (to comply with Part T (regulation T3(3)) would, if implemented, effectively eliminate the possibility of expansion proposals for the bar.

ARGUMENTS:

[applicant] If the proposals are not permitted to be implemented then the applicant will be forced to sell his business and seek employment elsewhere as the business does not at present provide him with an income which maintains the "standard of living" or "quality of life" which he and his wife/family require. The lounge bar does not rely on "passing trade" for the bulk of its business. Indeed the bar has a distinctly "local" clientelle and only a few of the local residents who use or potentially could use the bar facility are disabled. From experience gained over the past 5 no. years the applicant can confirm that the need for a disabled toilet facility within his premises could never be considered to have a high priority.

[local building control authority-COMMENT] The local building control authority forward all relevant correspondence and drawings to BD as complying with regulation 25 of the Building (Procedure) (Scotland) Regulations 1981.

[BD-COMMENT] BD informs the local building control authority of proposed decision to grant relaxation and gives them two weeks to comment on the decision.

DETERMINATION: Relaxation is granted.

REASONS: Relaxation is considered to be reasonable in the particular circumstances of this case as long as at least one water-closet compartment shall be provided for the use of ambulant disabled persons in male toilets.

CASE FILE NUMBER: 134
PURPOSE GROUP: 7B (storage and warehousing)
STATUS: existing
REG/VERSION: K10/1981
CLAUSE: paragraph (2)
SUBJECT: ventilation; mechanical ventilation
ELEMENTS: ventilation fan
RELATIONSHIP: between ventilation rate per person and volume of air to be changed
DATES: 24-03-88 to 07-09-88

ISSUE: The applicant is appealing for relaxation of regulation K10(2); "...it shall be ventilated...by mechanical means so as to provide a fresh air supply at not less than 1 air change per hour". (ref. schedule 11 1981 regulations). The applicant goes on to say that one fan will be adequate to ventilate a warehouse that is unmanned except during loading and unloading procedures; instead of the 3 fans required by regs.

ARGUMENTS:

[applicant] Applicant applies for relaxation to allow ventilation of warehouse to be undertaken by a single fan instead of the regulation 3 fans. He argues that the warehouse is unmanned for a majority of the time and only needs ventilation during loading and unloading procedures. During these procedures the large bay doors are open and the warehouse is effectively naturally ventilated.

[local building control authority] The local building control authority do not feel that the reasons the appellant gives in support of relaxation show that their warehouse operates in a way different from many others. The standard required by regulation is fairly nominal. Since there is no regulatory control over what could be stored in the building at any future date relaxation from the already nominal level would be irresponsible.

[local building control authority-COMMENT] In accordance with regulation 25 of the Building (Procedure) (Scotland) regulations 1981; the local building control authority forwarded copies of relevant drawings and correspondence to BD.

[BD] BD propose to side with the applicant and go against the decision of the building control authority saying that in this case they will calculate the amount of ventilation needed by the number of people rather than volume of building so only one extract fan need be located suitably to allow an air moving capacity of not less than 250 cubic. metres/hour. BD allow 2 weeks for building control authority to comment on proposed decision.

[local building control authority-COMMENT] Building control authority reply stating they do not wish to comment on proposed decision.

DETERMINATION: Relaxation is granted.

CONDITIONS: Building is provided with a suitably located extract fan having air moving capacity of at least 250 cubic metres/hour.

REASONS: It was decided to base the ventilation rate per person. Therefore 3 fans were not necessary, just one well located fan.

CASE FILE NUMBER: 141
PURPOSE GROUP: 1B (dwellings)
STATUS: existing
REG/VERSION: K6/1981
CLAUSE: paragraph (a)
SUBJECT: ventilation; watercloset ventilation
ELEMENTS: extractor fan
RELATIONSHIP:
DATES: 12:01:89 to 02:03:89

ISSUE: The applicant appeals for relaxation of regulation K6; "Every bathroom, washroom or watercloset forming part of a house shall be ventilated - a) direct to the external air by a window, rooflight or ventilator....". The applicant has constructed a porch only to be told that it is unsatisfactory to have the ventilator window of the watercloset open into the porch. He is now faced with a bill for a full extractor system to bring the ventilation of the watercloset up to standards.

ARGUMENTS:

[applicant] If the applicant was prepared to spend this amount of money he would not be so aggrieved now, in fact there is doubt to whether or not he would have carried out the construction of the porch. At present the toilet is well aired all the time, if a fan was fitted the watercloset would only be aired when the light was switched on making the watercloset stuffy and smelly. The window opening into the porch is a large window and with the porch door opening and closing many times a day the applicant feels the watercloset is ventilated enough.

[local building control authority] The local building control authority repeat that the requirement of the regulation is that a watercloset shall be ventilated direct to the external air by a window, rooflight or ventilator. They consider that the proposed arrangement whereby the W.C compartment is ventilated by a window which opens into the conservatory is unsatisfactory.

[local building control authority-COMMENT] All relevant documentation, drawings and correspondence are forwarded to BD as complying with regulation 25 of the Building (Procedure) (Scotland) Regulations 1981.

[BD] BD propose to uphold the local building control authority's decision and refuse relaxation of regulation K6. BD gives the local building control authority and applicant 14 days to comment on BD's decision.

DETERMINATION: Relaxation is refused.

REASONS: It is considered reasonable that, in relation to the purpose of the regulation which is to ensure the availability of satisfactory ventilation to the watercloset, the appeal be refused.

CASE FILE NUMBER: 150
PURPOSE GROUP: 4 (shops and commercial)
STATUS: existing
REG/VERSION: T3/1981
CLAUSE: paragraph (1)
SUBJECT: convenience; means of access for the disabled
ELEMENTS: disabled access ramp
RELATIONSHIP:
DATES: 19:04:90 to 12:07:90

ISSUE: The applicant is appealing for relaxation of regulation T3(1); "The means of access - a) to any storey to which there is access at ground level shall include suitable provision for access by disabled persons; and b) within such a storey as aforesaid shall include suitable provision for access by disabled persons." During alteration to the premises the step from the footpath to the new access door increased from 75 mm to 125 mm. The premises are listed and consequently the new fenestration of the front elevation was the subject of prolonged discussions with the local building control authority and in the course of these discussions the position of the access door was agreed. The applicants confirm their willingness to provide a portable timber ramp which would be kept on the premises and be placed in position on the pavement when necessary.

ARGUMENTS:

[architect] A building warrant for the alteration work was issued by the local building control authority. During the building operation the progress of the contract was continually monitored by the local building control authority's building inspectors, and at no time was any comment made about the provision of a ramped access. It was only when the project was finished and the application for a completion certificate was made that the matter was raised. The applicant feels that it is grossly unfair to expect their clients to provide a ramped access at such a late stage in the proceedings.

[local building control authority-COMMENT] All relevant drawings and correspondence are sent to BD as complying with regulation 25 of the Building (Procedure) (Scotland) Regulations 1981.

[local building control authority] The work indicated, involved internal alterations and a new frontage which included the re-siting of the original door entry. The work was carried out, but through the course of the local building control authority's inspections it was found that the re-sited door opening did not give level access between the street and the floor level of the shop as had been indicated on the formally granted building warrant. Instead a single step of between 100 mm and 125 mm had been formed at the one and only entry into the premises creating a contravention of regulation T3(1).

[BD-COMMENT] BD propose to uphold the local building control authority's decision and gives appellant two weeks to forward any comments.

DETERMINATION: Relaxation is refused.

REASONS: The applicant's suggestion of the provision of a portable timber ramp which would be kept on the premises and be placed in position on the pavement when necessary is considered to be a totally unsatisfactory solution in respect of the contravention. BD find that no good reason related to the purpose of regulation T3(1) has been given which would justify dispensing with the regulation.

CASE FILE NUMBER: 154
PURPOSE GROUP: 4 (shops and commercial)
STATUS: existing
REG/VERSION: E6/1981
CLAUSE: paragraph (1)(a)
SUBJECT: fire safety; the number of fire exits
ELEMENTS: fire escape stair
RELATIONSHIP:
DATES: 24:05:90 to 01:10:90

ISSUE: The applicant appeals for relaxation of regulation E6(1)(a) and provide one escape route as opposed to the statutory two, for a building in this class. The applicant proposes to install an exterior staircase for escape in respect of second and third floor levels of the premises.

ARGUMENTS:

[applicant] The shop previously only had one escape route from the upper floors which was of timber construction and did not open directly to the outside. It is the intention to build a new non-combustible stair to exit directly to the outside in the proposals to reconstruct the property. As the shop has a restricted floor area the space taken up by another stair would adversely effect the sales area. Also the cost of providing an additional stair would add greatly to the total cost of the project.

[local building control authority] A more recent scheme for the redevelopment of the building including two stairways has been approved for building warrant. At the time of the application for relaxation it was recognised that a very considerable reconstruction of the building was required giving every opportunity to comply with the requirements for two escape routes. In these circumstances the substantial relaxation to one escape route was considered unreasonable. Now that it is clear that the rebuild will be total, and as plans showing compliance have been submitted and approved, it is considered that the refusal was the reasonable decision.

[local building control authority-COMMENT] All relevant documentation, drawings and correspondence are forwarded to BD as complying with regulation 25 of the Building (Procedure) (Scotland) Regulations 1981.

[BD-COMMENT] BD propose to substitute for the local building control authority's decision a direction relaxing regulation E6(1)(a). The local building control authority are given 14 days to forward any comments they wish to make on BD's decision.

[firemaster-COMMENT] The firemaster confirms that he has no representation to make against the relaxation of the regulation being granted.

DETERMINATION: Relaxation is granted.

REASONS: It is considered that it would not be reasonable in the circumstances related to this existing building to comply with the regulation standards. This ruling, however, shall apply only to a storey not exceeding 11 metres in height.

CASE FILE NUMBER: 156
PURPOSE GROUP: 1C (dwellings)
STATUS: existing
REG/VERSION: S3/1981
CLAUSE: paragraph (1)
SUBJECT: safety; headroom of stairs
ELEMENTS: stair
DATES: 01:08:90 to 28:09:90

ISSUE: The applicant is appealing for relaxation of reg S3(1) to allow the minimal lowering of headroom of a stair, accessing two proposed bedrooms, from 2050 to 1950 mm. The application was made because of the restricted existing conditions of the slope of the roof and ridge height and because of the need of space for the family living in the house.

ARGUMENTS:

[architect] The architect states the family require the extra two bedrooms because of growing children, who need their own space. The boundaries of the site make it impossible to extend the floor area therefore the only feasible way is to extend upwards. Relaxation of the stair headroom is necessitated solely because of the low existing roof slope and ridge height.

[local building control authority-COMMENT] All relevant correspondence and drawings are sent to BD complying with reg 25(5) Building (Procedure) (Scotland) Regulations 1981.

[BD] BD propose to confirm the local building control authority's decision to refuse the relaxation of regulation S3(1) as it is considered that the standard proposed for the stair headroom is unacceptable.

[BD] BD allow two weeks for any comments, to be forwarded in writing, on this decision.

[architect] The architect asks BD to be reasonably practical stating that it is only the last couple of treads which will be 100 mm below the standard 2050 mm. If the family are denied this relaxation, they will reluctantly be forced into considering moving house, even though they are very happy and settled where they are. Such a move would be disruptive to them, and particularly to their young family.

DETERMINATION: Relaxation concerning stair headroom is refused.

REASONS: Relaxation has been granted for the minimal lowering of headroom in the two proposed bedrooms, however, it is considered that as it is a new stair it should comply with Building Standards regulation S3(1).

CASE FILE NUMBER: 165
PURPOSE GROUP: 3 (offices)
STATUS: existing
REG/VERSION: E14/1981
CLAUSE: paragraph (1)
SUBJECT: fire safety; fire doors
ELEMENTS: fire doors
RELATIONSHIP: sliding and swing doors
DATES: 30:04:91 to 16:08:91

ISSUE: Relaxation is sought for regulation E14,paragraph (1) of the provision that any door across an escape route shall open in the direction of travel. The architects replaced two sets of regulation doors in the entrance foyer with automatic sliding doors, therefore, only one set were considered in the context of regulation E14(1); "where the occupant capacity of a room or storey exceeds in the case of a building of occupancy group B (commercial), 50; every door across an escape route from that room or storey, or from any such accommodation shall open in the direction of travel towards the open air".

ARGUMENTS:

[applicant] The applicant states that the doors are fitted with a 'fail open' facility, supported by a battery pack which will automatically open the doors in the event of a power failure. The door mechanism can also be wired to the fire alarm panel so that the doors will open automatically if the fire alarm is activated. They also say the 'fail open' facility conforms to BS 5588 and that the doors across the escape route do not jeopardise the integrity of the escape route or the safety of the building's occupants.

[local building control authority] The local building control authority oppose relaxation for the following reasons;

1. The proposed 'fail open' facility is reliant entirely upon a mechanical device which may become inoperative, thus negating the doors use in the event of an outbreak of fire;
2. All the doors referred to by the appellant are required for escape purposes and therefore each must conform to the requirements of regulation E14(1);
3. The appellant's claim that the 'fail open' facility conforms to BS 5588 is of no consequence as a 'magic eye' is not permitted under the provision of part E of the Building Standards (Scotland) Regulations 1981-'87

[local building control authority-COMMENT] All relevant correspondence and drawings are sent to BD complying with regulation 25(5) Building (Procedure) (Scotland) Regulations 1981. The local building control authority ask BD to confirm their decision and refuse relaxation on the basis of the three reasons given.

[BD-COMMENT] BD propose to confirm the local building control authority's decision to refuse the relaxation of regulation E14(1) for the reasons given by the local authority. BD allow two weeks for any comments to be forwarded in writing on this decision.

DETERMINATION: Relaxation is refused.

REASONS: The standards proposed would not meet the minimum requirements for safe escape from fire in the building at the proposed doors across the escape route.

CASE FILE NUMBER: 172
PURPOSE GROUP: 4 (shops and commercial)
STATUS: change of use
REG/VERSION: T3/1981
CLAUSE: paragraph (1)
SUBJECT: convenience; means of access for the disabled
ELEMENTS: access ramp for disabled
RELATIONSHIP:
DATES: 06:04:92 to 01:07:92

ISSUE: The applicant is appealing for relaxation of regulation T3(1); "The means of access - a) to any storey to which there is access at ground level shall include suitable provision for access by disabled persons; and b) within such a storey as aforesaid shall include suitable provision for access by disabled persons." The applicant is appealing for dereliction of an existing concrete ramp leading to the professional shop.

ARGUMENTS:

[applicant] The applicant states that the professional golf shop is not open to the general public and only allows club members and temporary members who have paid their green fees to play on the course. The business is not advertised outwith the club and there is no indication from the public road that such a retail outlet exists. Disabled persons have suitable access to the club house but no disabled person has ever wished to visit the shop which is there only for members and visiting golfers. If a ramp was constructed it would have to be 16-18 feet long, blocking sub floor vents, window displays and two sun canopies would have to be taken down due to the lack of headroom and the fact that they can't be raised. The ramp would also be dangerous and slippery for golfers wearing studded shoes when approaching the shop.

[local building control authority-COMMENT] The local building control authority forward all relevant drawings and correspondence to BD as complying with regulation 25 of the Building (Procedure) (Scotland) Regulations 1981.

[local building control authority] The local building control authority oppose relaxation on the following grounds;

1. As the golf shop is open to the public a ramp is required to satisfy the Building Standards;
2. A concrete ramp can be designed or treated with non slip surface treatment to cope with the effects of frost, leaves etc;
3. Provisions of handrails and surface treatments can negate personal injuries and liabilities making the club safer;
4. No good reason has been given related to the purpose of the regulation which would justify dispensing with the regulation;

[BD-COMMENT] BD propose to side with the local building control authority and refuse relaxation. They gives the applicant and the authority 14 days to forward any comments on BD's decision.

[applicant-COMMENT] The applicant informs BD that he has taken the matter further to their local MP.

[firemaster-COMMENT] The MP writes to BD to receive comment on the situation.

[BD-COMMENT] BD write a scathing letter to the MP about the equal rights of disabled persons to enter public shops.

DETERMINATION: Relaxation is refused.

REASONS: It was concluded that it was reasonable that disabled people should wish to purchase goods at the retail shop area. From the drawings it was apparent that there would be no difficulty in providing the ramp.

CASE FILE NUMBER: 0004
PURPOSE GROUP: 2A (institutional and other residential)
STATUS: existing
REG/VERSION: reg 9 and reg 32 and S2.14/1990
CLAUSE: paragraph (14)
SUBJECT: safety; the gradient of ramps
ELEMENTS: ramp
RELATIONSHIP: 'ramp' and 'part of the building'
DATES: 27:04:92 to 06:07:92

ISSUE: To allow a warrant for a ramp of 1:8 relaxation is sought for regulation 9 as read technical standard S2.14 "there are no requirements for a ramp of a slope of less than 1:20. The gradient of any other ramp must not be steeper than 1:12; except, the gradient of a ramp giving access to a single dwelling must not be steeper than 1:8". Instead of repairing garden steps the applicants decided to construct a two stage ramp. Planning consent was granted but a building warrant wasn't applied for since the building regulation indicated that the regulations apply only to a ramp "forming part of the building". The ramp in question was a garden ramp and so did not form part of the building.

ARGUMENTS:

[architect]

1. Examination of regulation 32(1) and technical standard S2.14 leads to belief that regulations apply only to "stair or ramp which forms part of a building or which is provided to meet a requirement of these regulations";
2. It was never admitted by the applicant that the regulations apply: ramp is now constructed;
3. Notwithstanding 1 and 2 above, the ramp is constructed to gradient 1:8; it is submitted therefore that the ramp is not dangerous;
4. In any case, gradient of 1:8 is less steep than St John's Road bounding the property;
5. Notwithstanding non admission in item 2 above, all other technical standard requirements have been met;
6. The ramp is not for wheelchair use.

[local building control authority] The local building control authority refuse relaxation because "it is considered that no good reason related to the purpose of these regulations has been advanced which would justify relaxation in the particular circumstances of this case". They go on to comment on the grounds of the appellant's appeal;

1. The appellant's admission that the construction was carried out without an application for building warrant approval cannot be considered on grounds of appeal;
2. The application for relaxation was not refused on the grounds that the ramp was considered to be dangerous;
3. The fact that all other requirements of the technical standards have been complied with is irrelevant when considering the grounds for application and of appeal;
4. The fact that the ramp is currently not used for wheelchair use cannot be considered as grounds for relaxation or of appeal as the current purpose to which the ramp is utilised is also irrelevant, as the building's occupants and needs will change throughout the building's existence and the fact that ramps are provided for people of all disabilities.

[local building control authority-COMMENT] The local building control authority send all relevant correspondence and drawings to BD as complying with regulation 25 of the Building (Procedure) (Scotland) Regulations 1981.

[BD-COMMENT] BD propose to confirm the local building control authority's decision to refuse relaxation and allow 21 days for any comments to be forwarded on this decision. BD receive more extensive drawings and photographs of the site which the local building control had not seen.

DETERMINATION: Relaxation is relaxed.

REASONS: After receiving more extensive plans and photographs of the site BD states that the ramp is still part of the building, but thinks that it is coincidental as the ramp is intended to provide a route from the rear garden levels to the front garden. The ramp is an improvement on the previous route which lead from the old stair to a garden stair by way of a path.

CASE FILE NUMBER: 0012
PURPOSE GROUP: 7A (storage and warehousing)
STATUS: new building
REG/VERSION: regulation 9 as read with technical standard D2.1/1990
CLAUSE:
SUBJECT: fire safety; compartmentation
ELEMENTS: compartment areas
RELATIONSHIP:
DATES: 30:11:92 to 19:01:93

ISSUE: The applicant appeals for the relaxation of standard D2.1, "a building, or part of a building, greater than the appropriate limits given in the table to this standard must be sub divided into compartments... Sub division must be by 'compartment walls' or 'compartment floors' or by 'separating walls' or 'separating floors'.

ARGUMENTS:

[applicant] The area to be used is for temporary storage and dispatch of cased goods and as such the potential fire loading will always be less than that from a traditionally racked or palletised warehouse. A full automatic fire control system will be installed.

[local building control authority] The local building control authority refuse relaxation on the grounds of public safety that the building must be sub divided into compartments which do not exceed the limits set out in the regulations.

[firemaster-COMMENT] The firemaster states that he does not object to the local building control authority's refusal of regulation 9 as read with technical standard D2.1.

[local building control authority-COMMENT] All relevant drawings, documentation and correspondence are forwarded to BD as complying with regulation 25 of the Building (Procedure) (Scotland) Regulations 1981.

[BD-COMMENT] BD propose to confirm the local building control authority's decision and refuse to relax regulation 9 as read with technical standard D2.1.

[BD] It is considered that no good reason has been submitted which would justify relaxation of the technical standard to allow the storage compartment area as proposed. The proposed use of the compartment falls within the range of uses taken into account when the standard was reviewed after full consultation, as the envisaged storage use would frustrate the objective of the regulation which limits the potential fire load by compartmentation to a maximum which can be reasonably contained to the site regardless of the amount of storage at any time.

[BD-COMMENT] The firemaster, local building control authority and applicant are given 14 days to forward any comments on BD's decision.

[local building control authority] After a planning and development committee submission and having discussed the matter with a member of the fire research section of the Building Research Establishment the local building control authority would recommend that the application be granted subject to the following conditions;

1. relaxation shall apply only to allow the area in the wet goods hall to be not greater than 2500 sq. metres and the cubic capacity measured to the underside of the eaves to be not more than 18,000 cubic metres and that each compartment in the wet goods hall is fitted with an automatic sprinkler system complying with the recommendations of the Rules for Automatic Sprinkler Installations 29th Edition as amended to date issued by the Fire Officer's committee or the Rules for Automatic Sprinkler Installations 1990;
2. relaxation shall apply only to allow the cubic capacity of the office building to be not greater than 28,000 cubic metres;
3. that the office building is provided with an automatic fire control system;

The reason for this decision is because the local building control authority consider that the standard proposed will fulfill the object of the regulations.

[BD-COMMENT] BD propose to refuse relaxation of regulation 9 as read with technical standard D2.1 and gives the firemaster, local building control authority and applicant 14 days to forward any comments on BD's decision.

DETERMINATION: Relaxation is refused.

REASONS: The potential fire load in the compartment proposed is clearly a basic concern. The applicant proposes to increase the fire load but does not offer any protective measures to restrict the hazard to that the regulation seeks to impose.

Appendix B

The Model Knowledge Base

This appendix shows a Prolog file which contains a Model Knowledge Base formulated from requirements in the 1981 version of the Scottish building regulations. Two Prolog predicates are used in this file: goal/2 and model/6.

goal($Goal$, M_{id}).
: M_{id} is the id number of the root model of the abstraction hierarchy of a goal $Goal$.

model(M_{id}, M_{pid}, M_c, M_{cd}, M_{sd}, M_s).
: M_{id} is the id number of the model, M_{pid} is the id number of the parent of the model or "nil" if the model is the root model of an abstraction hierarchy, M_c is the building categories 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".

```
/*-------
  goals
-------*/

goal(safety, m1).
goal(convenience, m2).
goal(energy_conservation, m3).
goal(health, m4).

/*-----------------
  non-leaf models
-----------------*/

%---- safety

model(m1, nil,
      [building],
      [building(X)],
      [ensure(X,safety_of_people)],
      strong
     ).

model(m5, m1,
      [building],
      [building(X)],
      [ensure(X,safety_in_case_of_fire)],
      strong
     ).

model(m6, m5,
      [building],
      [building(X)],
      [ensure(X,structural_stability_in_case_of_fire)],
      strong
     ).

model(m7, m5,
      [building],
      [building(X)],
      [provide(X,adequate_means_of_escape_from_fire)],
      strong
     ).

model(m10, m1,
      [building],
      [building(X)],
      [provide(X,safe_means_of_passage)],
      strong
     ).

%---- convenience

model(m2, nil,
      [building],
      [building(X)],
      [ensure(X,convenience_of_people)],
      strong
     ).

model(m200, m2,
      [building],
      [building(X)],
      [provide(X,facility_for_disabled)],
      strong
     ).

%---- energy conservation
```

```
model(m3, nil,
      [building],
      [building(X)],
      [ensure(X,energy_conservation)],
      strong
     ).

%---- health

model(m4, nil,
      [building],
      [building(X)],
      [ensure(X,health_of_people)],
      strong
     ).

model(m100, m4,
      [building],
      [building(X)],
      [provide(X,adequate_supply_of_air)],
      strong
     ).

/*-------------
  leaf models
-------------*/

%---- From D3

model(m11, m6,
      [low_hazard_storage],
      [building(A), number_of_storey(A,N),  N = 1,
       storey(C), part_of(C,A), area(C,X,square_metre), X >= 14000,
       compartment(B), part_of(B,A)],
      [area(B,Y,square_metre), Y =< 14000],
      strong
     ).

model(m12, m6,
      [low_hazard_storage],
      [building(A), number_of_storey(A,N), N > 1,
       cubic_capacity(A,X,cubic_metre), X >= 21000,
       compartment(B), part_of(B,A)],
      [cubic_capacity(B,Y,cubic_metre), Y =< 21000],
      strong
     ).

model(m13, m6,
      [low_hazard_storage],
      [building(A), number_of_storey(A,N), N > 1,
       storey(C), part_of(C,A), area(C,X,square_metre), X >= 2800,
       compartment(B), part_of(B,A)],
      [area(B,Y,square_metre), Y =< 2800],
      strong
     ).

model(m14, m6,
      [high_hazard_storage],
      [building(A), number_of_storey(A,N), N = 1,
       storey(C), part_of(C,A), area(C,X,square_metre), X >= 1000,
       compartment(B), part_of(B,A)],
      [area(B,Y,square_metre), Y =< 1000],
      strong
     ).

model(m15, m6,
      [high_hazard_storage],
      [building(A), number_of_storey(A,N), N > 1,
```

```
        cubic_capacity(A,X,cubic_metre), X >=4200,
        compartment(B), part_of(B,A)],
       [cubic_capacity(B,Y,cubic_metre), Y =< 4200],
       strong
      ).

model(m16, m6,
       [high_hazard_storage],
       [building(A), number_of_storey(A,N), N > 1,
        storey(C), part_of(C,A), area(C,X,square_metre), X >= 460,
        compartment(B), part_of(B,A)],
       [area(B,Y,square_metre), Y =< 460],
       strong
      ).

%---- From D4

model(m17, m6,
       [building],
       [place(A), building(B), part_of(A,B),
        heat_producing_appliance(C), part_of(C,A),
        number(C,X), X >= 1, heat_producing_rate(C,Y,kw), Y >= 44,
        fuel(C,solid)],
       [is(A,place_of_special_fire_risk)],
       strong
      ).

model(m18, m6,
       [building],
       [place(A), building(B), part_of(A,B),
        heat_producing_appliance(C), part_of(C,A),
        number(C,X), X >= 1, heat_producing_rate(C,Y,kw), Y >= 60,
        fuel(C,gas)],
       [is(A,place_of_special_fire_risk)],
       strong
      ).

model(m19, m6,
       [building],
       [place(A), building(B), attached(A,B),
        heat_producing_appliance(C), part_of(C,A),
        number(C,X), X >= 1, heat_producing_rate(C,Y,kw), Y >= 44,
        fuel(C,solid)],
       [is(A,place_of_special_fire_risk)],
       strong
      ).

model(m20, m6,
       [building],
       [place(A), building(B), attached(A,B),
        heat_producing_appliance(C), part_of(C,A),
        number(C,X), X >= 1, heat_producing_rate(C,Y,kw), Y >= 60,
        fuel(C,gas)],
       [is(A,place_of_special_fire_risk)],
       strong
      ).

model(m21, m6,
       [building],
       [place(A), roof(B), on(A,B),
        heat_producing_appliance(C), part_of(C,A),
        number(C,X), X >= 1, heat_producing_rate(C,Y,kw), Y >= 44,
        fuel(C,solid)],
       [is(A,place_of_special_fire_risk)],
       strong
      ).

model(m22, m6,
```

```
       [building],
       [place(A), roof(B), on(A,B),
        heat_producing_appliance(C), part_of(C,A),
        number(C,X), X >= 1, heat_producing_rate(C,Y,kw), Y >= 60,
        fuel(C,gas)],
       [is(A,place_of_special_fire_risk)],
       strong
      ).

%---- From D7

model(m23, m6,
       [building],
       [stair(A), stairway(B), used_as(B,exit), attribute(B,enclosed),
        part_of(A,B)],
       [material(A,non_combustible)],
       strong
      ).

model(m24, m6,
       [building],
       [stair(A), stairway(B),
        attribute(B,unenclosed), attribute(B,external),
        part_of(A,B)],
       [material(A,non_combustible)],
       strong
      ).

model(m25, m6,
       [building],
       [separating_wall(A)],
       [material(A,non_combustible)],
       strong
      ).

model(m26, m6,
       [building],
       [compartment_wall(A), lift(B), separate_from(A,B)],
       [material(A,non_combustible)],
       strong
      ).

model(m27, m6,
       [building],
       [floor(A), stairway(B), attribute(B,enclosed), landing(C),
        part_of(A,C), part_of(C,B)],
       [material(A,non_combustible)],
       strong
      ).

model(m28, m6,
       [building],
       [compartment_floor(A)],
       [material(A,non_combustible)],
       strong
      ).

model(m29, m6,
       [building],
       [separating_floor(A)],
       [material(A,non_combustible)],
       strong
      ).

model(m30, m6,
       [building],
       [wall(A), attribute(A,external)],
       [material(A,non_combustible)],
```

```
      strong
     ).

model(m31, m6,
      [building],
      [wall(A), boundary_of_building(B),
       distance(A,B,X,metre), X < 10],
      [material(A,non_combustible)],
      strong
     ).

model(m32, m6,
      [building],
      [enclosing_structure(A), pretected_zone(B), part_of(A,B)],
      [material(A,non_combustible)],
      strong
     ).

model(m33, m6,
      [building],
      [ramp(A), escape_route(B), part_of(A,B)],
      [material(A,non_combustible)],
      strong
     ).

model(m34, m6,
      [building],
      [floor(A), landing(B), access_route_to_house(C),
       part_of(A,B), part_of(B,C)],
      [material(A,non_combustible)],
      strong
     ).

model(m35, m6,
      [building],
      [stair(A), access_route_to_house(B),
       part_of(A,B)],
      [material(A,non_combustible)],
      strong
     ).

model(m36, m6,
      [building],
      [ramp(A), access_route_to_house(B),
       part_of(A,B)],
      [material(A,non_combustible)],
      strong
     ).

model(m37, m6,
      [building],
      [balcony(A), access_route_to_house(B),
       part_of(A,B)],
      [material(A,non_combustible)],
      strong
     ).

%---- From E6

model(m38, m7,
      [building],
      [room(A), occupancy_capacity(A,X), X >= 1, X =< 60],
      [escape_route(B), part_of(B,A), number(B,Y), Y >= 1],
      strong
     ).

model(m39, m7,
      [building],
```

```
       [room(A), occupancy_capacity(A,X), X >= 61, X =< 600],
       [escape_route(B), part_of(B,A), number(B,Y), Y >= 2],
       strong
      ).

model(m40, m7,
       [building],
       [room(A), occupancy_capacity(A,X), X > 600],
       [escape_route(B), part_of(B,A), number(B,Y), Y >= 3],
       strong
      ).

model(m41, m7,
       [building],
       [storey(A), occupancy_capacity(A,X), X >= 1, X =< 60],
       [escape_route(B), part_of(B,A), number(B,Y), Y >= 1],
       strong
      ).

model(m42, m7,
       [building],
       [storey(A), occupancy_capacity(A,X), X >= 61, X =< 600],
       [escape_route(B), part_of(B,A), number(B,Y), Y >= 2],
       strong
      ).

model(m43, m7,
       [building],
       [storey(A), occupancy_capacity(A,X), X > 600],
       [escape_route(B), part_of(B,A), number(B,Y), Y >= 3],
       strong
      ).

model(m44, m7,
       [shop],
       [storey(A), floor(B), part_of(B,A),
        height_above_ground(B,X,millimetre), X =< 4500],
       [escape_route(C), part_of(C,A), number(C,Y), Y >= 1],
       strong
      ).


model(m45, m7,
       [shop],
       [storey(A), floor(B), part_of(B,A),
        height_above_ground(B,X,millimetre), X > 4500],
       [escape_route(C), part_of(C,A), number(C,Y), Y >= 2],
       strong
      ).


model(m46, m7,
       [office],
       [storey(A), floor(B), part_of(B,A),
        height_above_ground(B,X,millimetre), X =< 11000],
       [escape_route(C), part_of(C,A), number(C,Y), Y >= 1],
       strong
      ).


model(m47, m7,
       [office],
       [storey(A), floor(B), part_of(B,A),
        height_above_ground(B,X,millimetre), X > 11000],
       [escape_route(C), part_of(C,A), number(C,Y), Y >= 2],
       strong
      ).
```

```
%---- From E10

model(m48, m7,
      [building],
      [stairway(A), used_as(A,exit)],
      [attribute(A,enclosed), protected_zone(B), part_of(A,B)],
      strong
     ).

model(m49, m7,
      [building],
      [ramp(A), used_as(A,exit)],
      [attribute(A,enclosed), protected_zone(B), part_of(A,B)],
      strong
     ).

model(m50, m7,
      [building],
      [stairway_enclosure(A), place_of_safety(B), at_level(B,ground)],
      [connected(A,B)],
      strong
     ).

model(m52, m7,
      [building],
      [stairway_enclosure(A), place_of_safety(B), at_level(B,ground),
       corridor(C)],
      [connected(A,C), connected(C,B)],
      strong
     ).

%---- E14

model(m53, m7,
      [residential_building, commercial_building, assembly_building],
      [room(A), occupancy_capacity(A,X), X > 50,
       escape_route(B), door(C), part_of(C,B), connected(A,B)],
      [can(C,open_outwards)],
      strong
     ).

model(m54, m7,
      [residential_building, commercial_building, assembly_building],
      [room(A), occupancy_capacity(A,X), X > 50,
       escape_route(B), door(C), part_of(C,B), connected(A,B),
       can(C,open_both_ways)],
      [transparent_upper_panel(D), part_of(D,C)],
      strong
     ).

model(m55, m7,
      [residential_building, commercial_building, assembly_building],
      [room(A), occupancy_capacity(A,X), X > 50,
       escape_route(B), door(C), part_of(C,B), connected(A,B),
       is(C,secured_from_outside)],
      [can(C,open_from_inside), indication_mark(D), part_of(D,C)],
      strong
     ).

model(m56, m7,
      [residential_building, commercial_building, assembly_building],
      [storey(A), occupancy_capacity(A,X), X > 50,
       escape_route(B), door(C), part_of(C,B), part_of(B,A)],
      [can(C,open_outwards)],
      strong
     ).

model(m57, m7,
```

```
      [residential_building, commercial_building, assembly_building],
      [storey(A), occupancy_capacity(A,X), X > 50,
       escape_route(B), door(C), part_of(C,B), part_of(B,A),
       can(C,open_both_ways)],
      [transparent_upper_panel(D), part_of(D,C)],
      strong
     ).

model(m58, m7,
      [residential_building, commercial_building, assembly_building],
      [storey(A), occupancy_capacity(A,X), X > 50,
       escape_route(B), door(C), part_of(C,B), part_of(B,A),
       is(C,secured_from_outside)],
      [can(C,open_from_inside), indication_mark(D), part_of(D,C)],
      strong
     ).

model(m59, m7,
      [industrial_building, storage],
      [room(A), occupancy_capacity(A,X), X > 10,
       escape_route(B), door(C), part_of(C,B), connected(A,B)],
      [can(C,open_outwards)],
      strong
     ).

model(m60, m7,
      [industrial_building, storage],
      [room(A), occupancy_capacity(A,X), X > 10,
       escape_route(B), door(C), part_of(C,B), connected(A,B),
       can(C,open_both_ways)],
      [transparent_upper_panel(D), part_of(D,C)],
      strong
     ).

model(m61, m7,
      [industrial_building, storage],
      [room(A), occupancy_capacity(A,X), X > 10,
       escape_route(B), door(C), part_of(C,B), connected(A,B),
       is(C,secured_from_outside)],
      [can(C,open_from_inside), indication_mark(D), part_of(D,C)],
      strong
     ).

model(m62, m7,
      [industrial_building, storage],
      [storey(A), occupancy_capacity(A,X), X > 10,
       escape_route(B), door(C), part_of(C,B), part_of(B,A)],
      [can(C,open_outwards)],
      strong
     ).

model(m63, m7,
      [industrial_building, storage],
      [storey(A), occupancy_capacity(A,X), X > 10,
       escape_route(B), door(C), part_of(C,B), part_of(B,A),
       can(C,open_both_ways)],
      [transparent_upper_panel(D), part_of(D,C)],
      strong
     ).

model(m64, m7,
      [industrial_building, storage],
      [storey(A), occupancy_capacity(A,X), X > 10,
       escape_route(B), door(C), part_of(C,B), part_of(B,A),
       is(C,secured_from_outside)],
      [can(C,open_from_inside), indication_mark(D), part_of(D,C)],
      strong
     ).
```

```
%---- From S3

model(m65, m10,
      [building],
      [stair(A), headroom(B), part_of(B,A)],
      [height_above_pitch(B,X,millimetre), X >= 2050],
      strong
     ).

model(m66, m10,
      [building],
      [access_stairway(A), attribute(A,external), stair(B), part_of(B,A),
       floor(C), connected(B,C),
       height_above_floor(B,X,millimetre), X > 600],
      [guarding_wall(D), part_of(D,B), height(D,Y,millimetre), Y >= 900],
      strong
     ).

model(m67, m10,
      [building],
      [access_stairway(A), attribute(A,external),
       stair(B), part_of(B,A), landing(C), connected(B,C),
       height_above_landing(B,X,millimetre), X > 600],
      [guarding_wall(D), part_of(D,B), height(D,Y,millimetre), Y >= 900],
      strong
     ).

model(m68, m10,
      [building],
      [access_stairway(A), attribute(A,external),
       stair(B), part_of(B,A), connected(B,ground),
       height_above_ground(B,X,millimetre), X > 600],
      [guarding_wall(D), part_of(D,B), height(D,Y,millimetre), Y >= 900],
      strong
     ).

model(m69, m10,
      [building],
      [access_stairway(A), attribute(A,external),
       stair(B), part_of(B,A), floor(C), connected(B,C),
       height_above_floor(B,X,millimetre), X > 600],
      [balustrade(D), part_of(D,B), height(D,Y,millimetre), Y >= 900],
      strong
     ).

model(m70, m10,
      [building],
      [access_stairway(A), attribute(A,external),
       stair(B), part_of(B,A), landing(C), connected(B,C),
       height_above_landing(B,X,millimetre), X > 600],
      [balustrade(D), part_of(D,B), height(D,Y,millimetre), Y >= 900],
      strong
     ).

model(m71, m10,
      [building],
      [access_stairway(A), attribute(A,external),
       stair(B), part_of(B,A), connected(B,ground),
       height_above_ground(B,X,millimetre), X > 600],
      [balustrade(D), part_of(D,B), height(D,Y,millimetre), Y >= 900],
      strong
     ).

model(m72, m10,
      [building],
      [access_stairway(A), attribute(A,external),
       stair(B), part_of(B,A), floor(C), connected(B,C),
```

```
        height_above_floor(B,X,millimetre), X > 600],
       [railing(D), part_of(D,B), height(D,Y,millimetre), Y >= 900],
       strong
      ).

model(m73, m10,
       [building],
       [access_stairway(A), attribute(A,external),
        stair(B), part_of(B,A), landing(C), connected(B,C),
        height_above_landing(B,X,millimetre), X > 600],
       [railing(D), part_of(D,B), height(D,Y,millimetre), Y >= 900],
       strong
      ).

model(m74, m10,
       [building],
       [access_stairway(A), attribute(A,external),
        stair(B), part_of(B,A), connected(B,ground),
        height_above_ground(B,X,millimetre), X > 600],
       [railing(D), part_of(D,B), height(D,Y,millimetre), Y >= 900],
       strong
      ).

model(m75, m10,
       [building],
       [escape_stairway(A), attribute(A,external),
        stair(B), part_of(B,A), floor(C), connected(B,C),
        height_above_floor(B,X,millimetre), X > 600],
       [guarding_wall(D), part_of(D,B), height(D,Y,millimetre), Y >= 900],
       strong
      ).

model(m76, m10,
       [building],
       [escape_stairway(A), attribute(A,external),
        stair(B), part_of(B,A), landing(C), connected(B,C),
        height_above_floor(B,X,millimetre), X > 600],
       [guarding_wall(D), part_of(D,B), height(D,Y,millimetre), Y >= 900],
       strong
      ).

model(m77, m10,
       [building],
       [escape_stairway(A), attribute(A,external),
        stair(B), part_of(B,A), connected(B,ground),
        height_above_floor(B,X,millimetre), X > 600],
       [guarding_wall(D), part_of(D,B), height(D,Y,millimetre), Y >= 900],
       strong
      ).

model(m78, m10,
       [building],
       [escape_stairway(A), attribute(A,external),
        stair(B), part_of(B,A), floor(C), connected(B,C),
        height_above_floor(B,X,millimetre), X > 600],
       [balustrade(D), part_of(D,B), height(D,Y,millimetre), Y >= 900],
       strong
      ).

model(m79, m10,
       [building],
       [escape_stairway(A), attribute(A,external),
        stair(B), part_of(B,A), landing(C), connected(B,C),
        height_above_floor(B,X,millimetre), X > 600],
       [balustrade(D), part_of(D,B), height(D,Y,millimetre), Y >= 900],
       strong
      ).
```

```
model(m80, m10,
      [building],
      [escape_stairway(A), attribute(A,external),
       stair(B), part_of(B,A), connected(B,ground),
       height_above_floor(B,X,millimetre), X > 600],
      [balustrade(D), part_of(D,B), height(D,Y,millimetre), Y >= 900],
      strong
     ).

model(m81, m10,
      [building],
      [escape_stairway(A), attribute(A,external),
       stair(B), part_of(B,A), floor(C), connected(B,C),
       height_above_floor(B,X,millimetre), X > 600],
      [railing(D), part_of(D,B), height(D,Y,millimetre), Y >= 900],
      strong
     ).

model(m82, m10,
      [building],
      [escape_stairway(A), attribute(A,external),
       stair(B), part_of(B,A),
       landing(C), connected(B,C),
       height_above_floor(B,X,millimetre), X > 600],
      [railing(D), part_of(D,B), height(D,Y,millimetre), Y >= 900],
      strong
     ).

model(m83, m10,
      [building],
      [escape_stairway(A), attribute(A,external),
       stair(B), part_of(B,A), connected(B,ground),
       height_above_floor(B,X,millimetre), X > 600],
      [railing(D), part_of(D,B), height(D,Y,millimetre), Y >= 900],
      strong
     ).

%---- From S5

model(m84, m10,
      [building],
      [balustrade(A), opening(B), part_of(B,A)],
      [width(B,X,millimetre), X < 100],
      strong
     ).

model(m85, m10,
      [building],
      [balustrade(A), rail(B), part_of(B,A)],
      [width_between(B,X,millimetre), X < 100],
      strong
     ).

%---- From T3

model(m201, m200,
      [residential_building, commercial_building,
       assembly_building, industrial_building],
      [storey(A), access_by_disabled(B), part_of(B,A),
       sanitary_convenience(C), part_of(C,A)],
      [sanitary_convenience_for_disabled(D), part_of(D,C)],
      strong
     ).

model(m202, m200,
      [residential_building, commercial_building, assembly_building,
       industrial_building],
      [building(A), storey(B), storey(C),
```

```
        part_of(B,A), part_of(C,A),
        at_level(B,ground), access_to_building(D), part_of(D,B)],
       [access_by_disabled(E), part_of(E,C)],
       strong
      ).

%---- From K6

model(m101, m100,
       [house],
       [bathroom(A), opening(B), part_of(B,A),
        floor(C), part_of(C,A)],
       [area(B,X,square_metre), area(C,Y,square_metre), X >= Y / 20],
       strong
      ).

model(m102, m100,
       [house],
       [bathroom(A), opening(B), part_of(B,A)],
       [area(B,X,square_metre), X >= 0.1],
       strong
      ).

model(m103, m100,
       [house],
       [washroom(A), opening(B), part_of(B,A),
        floor(C), part_of(C,A)],
       [area(B,X,square_metre), area(C,Y,square_metre), X >= Y / 20],
       strong
      ).

model(m104, m100,
       [house],
       [washroom(A), opening(B), part_of(B,A)],
       [area(B,X,square_metre), X >= 0.1],
       strong
      ).

model(m105, m100,
       [house],
       [watercloset(A), opening(B), part_of(B,A),
        floor(C), part_of(C,A)],
       [percentage_area(B,X), X >= 5],
       strong
      ).

model(m106, m100,
       [house],
       [watercloset(A), opening(B), part_of(B,A)],
       [area(B,X,square_metre), X >= 0.1],
       strong
      ).

%---- From K10

model(m107, m100,
       [assembly_building, recreational_building],
       [room(A), not_used_as(A,storage), not_require(A,temperature_control),
        cubic_space_per_occupant(A,X,cubic_metre), X > 0, X =< 2.8,
        fan(B), part_of(B,A)],
       [rate_of_air_supply_per_hour_per_person(B,Y,cubic_metre),
        Y >= 28],
       strong
      ).

model(m108, m100,
       [assembly_building, recreational_building],
       [room(A), not_used_as(A,storage), not_require(A,temperature_control),
```

```
        cubic_space_per_occupant(A,X,cubic_metre), X > 2.8, X =< 21,
        floor(B), part_of(B,A), opening(C), part_of(C,A)],
       [percentage_area(C,Y), Y >= 5],
       strong
      ).

model(m109, m100,
       [assembly_building, recreational_building],
       [room(A), not_used_as(A,storage), not_require(A,temperature_control),
        cubic_space_per_occupant(A,X,cubic_metre), X > 2.8, X =< 21,
        opening(C), part_of(C,A)],
       [area_per_occupant(C,Z,square_millimetre), Z >= 6500],
       strong
      ).

model(m110, m100,
       [assembly_building, recreational_building],
       [room(A), not_used_as(A,storage), not_require(A,temperature_control),
        cubic_space_per_occupant(A,X,cubic_metre), X > 2.8, X =< 8,
        fan(B), part_of(B,A)],
       [rate_of_air_supply_per_hour_per_person(B,Y,cubic_metre), Y >= 28],
       strong
      ).

model(m111, m100,
       [assembly_building, recreational_building],
       [room(A), not_used_as(A,storage), not_require(A,temperature_control),
        cubic_space_per_occupant(A,X,cubic_metre), X > 8, X =< 11,
        fan(B), part_of(B,A)],
       [rate_of_air_supply_per_hour_per_person(B,Y,cubic_metre), Y >= 20],
       strong
      ).

model(m112, m100,
       [assembly_building, recreational_building],
       [room(A), not_used_as(A,storage), not_require(A,temperature_control),
        cubic_space_per_occupant(A,X,cubic_metre), X > 11, X =< 14,
        fan(B), part_of(B,A)],
       [rate_of_air_supply_per_hour_per_person(B,Y,cubic_metre), Y >= 17],
       strong
      ).

model(m113, m100,
       [assembly_building, recreational_building],
       [room(A), not_used_as(A,storage), not_require(A,temperature_control),
        cubic_space_per_occupant(A,X,cubic_metre), X > 14,
        fan(B), part_of(B,A)],
       [rate_of_air_supply_per_hour_per_person(B,Y,cubic_metre), Y >= 12],
       strong
      ).

model(m114, m100,
       [storage,warehouse],
       [room(A), occupancy_capacity(A,0),
        opening(B), part_of(B,A)],
       [area(B,X,square_millimetre), X >= 6500],
       strong
      ).

model(m115, m100,
       [storage,warehouse],
       [room(A), occupancy_capacity(A,0),
        fan(B), part_of(B,A)],
       [air_change_per_hour(B,X), X >= 1],
       strong
      ).

%---- From K14
```

```
model(m116, m100,
      [building],
      [room(A), opening(B), part_of(B,A)],
      [top_height_above_floor(B,X,metre), X >= 2],
      strong
     ).
```

Appendix C

The Case Library

Six Prolog predicates are used in the Case Library: case/8, additional_fact/2, position/4, argument/4, support/4 and oppose/4.

case(C_{id}, C_g, C_c, C_{cd}, C_{sd}, C_d, C_m, C_t).
: C_{id} is the id number of the case, which is the case file number, 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, C_d is the decision about relaxation taken by the BD, i.e., "granted" or "refused", C_m contains the id number of models which were retrieved as relevant to this case, or "nil" if there was no relevant model, and C_t contains the id number of a strong model (or strong models) which the case were supposed to satisfy but it did not satisfy.

additional_fact(C_{id}, $Facts$).
: C_{id} is the id number of the case, which is the case file number, and $Facts$ are the description of facts given about the case and not described in the C_{cd} and C_{sd} of the case.

The argumentation involved in each case is represented using the following four predicates: position/4, argument/4, support/4 and oppose/4.

position(C_{id}, P_{id}, $Player$, $Position$).
: P_{id} is the id number of the position. C_{id} is the id number of a case in which a position $Position$ is put forward by a player $Player$.

argument(C_{id}, A_{id}, $Player$, $Argument$).
: A_{id} is the id number of the argument. C_{id} is the id number of a case in which an argument $Argument$ is put forward by a player $Player$.

support(C_{id}, A_{id}, P_{id}, $Reasons$).
: In a case with the id number C_{id}, an argument with the id number A_{id} supports a position with the id number P_{id} and $Reasons$ are the reasons for providing arguments for supporting the position.

oppose(C_{id}, A_{id}, P_{id}, *Reasons*).

In a case with the id number C_{id}, an argument with the id number A_{id} opposes a position with the id number P_{id} and *Reasons* are the reasons for providing arguments for supporting the position.

Reasons in support/4 and oppose/4 are the reasons for providing arguments for supporting or opposing a particular position, i.e., “full compliance”, “partial compliance”, “non-compliance”, “reasonable”, “unreasonable”, “practicable”, “impracticable” or predicates which represent reasons other than these predefined seven reasons.

```
case(c11, safety, [office],
    [stair(a), stairway(c), used_as(c, exit), attribute(c, enclosed),
     part_of(a, c), building(b), b is small, part_of(c, b)],
    [material(a, hardwood)],
    granted, [m23, m24, m300], [m23, m24]).
position(c11, p1, applicant, [material(a, hardwood)]).
position(c11, p2, lbca, [material(a, concrete)]).
position(c11, p3, lbca, [material(a, steel), covered_with(a, timber)]).
argument(c11, a1, applicant, [b is small]).
argument(c11, a2, applicant, [hardwood is fire_resistant]).
argument(c11, a3, applicant,
     [room(r), part_of(r, b), window(w), part_of(w, r)]).
argument(c11, a4, lbca, [hardwood is combustible]).
argument(c11, a5, lbca, [concrete is non_combustible]).
argument(c11, a6, bd, [hardwood is combustible]).
argument(c11, a7, bd, [b is small]).
argument(c11, a8, bd, [occupancy_capacity(b, small)]).
support(c11, a1, p1, [practicable]).
support(c11, a2, p1, [partial_compliance]).
support(c11, a3, p1, [[provide(b, alternative_escape)]]).
support(c11, a5, p2, [practicable, full_compliance]).
oppose(c11, a4, p1, [non_compliance]).
oppose(c11, a6, p3, [non_compliance]).
oppose(c11, a7, p2, [impracticable]).

case(c41, safety, [residential_building],
     [stairway(a), used_as(a, exit)],
     [attribute(a, open), material(a, steel)],
     granted, [m48, m301], [m48]).
position(c41, p1, applicant, [attribute(a, open), material(a, steel)]).
position(c41, p2, lbca, [attribute(a, enclosed), material(a, steel)]).
argument(c41, a1, applicant,
     [building(b), part_of(a, b), b is existing, a is new]).
argument(c41, a2, lbca, [attribute(a, external)]).
argument(c41, a3, bd, [attribute(a, external)]).
support(c41, a1, p1, [practicable, partial_compliance]).
support(c41, a2, p2, [full_compliance, practicable]).
oppose(c41, a2, p1, [non_compliance]).
oppose(c41, a3, p2, [impracticable]).

case(c42, health, [dwelling],
     [gable_bedroom(a), height_above_floor(a, 2070, millimetre),
      dormer_window(b), part_of(b, a)],
     [top_height_above_floor(b, 1.75, metre)],
     granted, [m116, m302], [m116]).
position(c42, p1, applicant, [top_height_above_floor(b, 1.75, metre)]).
position(c42, p2, lbca, [top_height_above_floor(b, A, metre), A>=2]).
argument(c42, a1, applicant, [top_height_above_floor(a, 2.07, metre)]).
argument(c42, a2, lbca, [b is opening]).
support(c42, a1, p1, [reasonable]).
support(c42, a2, p2, [full_compliance]).
oppose(c42, a1, p2, [impracticable]).
oppose(c42, a2, p1, [non_compliance]).

case(c77, safety, [house],
     [dormer_staircase(a), tread(b), part_of(b, a),
      headroom(c), part_of(c, b)],
     [height_above_pitch(c, 1840, millimetre)],
     refused, [m65], [m65]).
position(c77, p1, applicant, [height_above_pitch(c, 1840, millimetre)]).
position(c77, p2, lbca, [height_above_pitch(c, A, millimetre), A>=2050]).
argument(c77, a1, applicant, [a is used_infrequently]).
argument(c77, a2, lbca, [a is new]).
support(c77, a1, p1, [[provide(a, safe_means_of_passage)]]).
support(c77, a2, p2, [full_compliance]).
oppose(c77, a2, p1, [non_compliance]).

case(c82, safety, [dwelling],
```

```
     [loadbearing_partition_wall(a)],
     [material(a, combustible)],
     granted, [m25, m30, m303], [m25, m30]).
position(c82, p1, applicant, [material(a, combustible)]).
position(c82, p2, lbca, [material(a, non_combustible)]).
argument(c82, a1, applicant, [a is existing]).
argument(c82, a2, applicant, [reduce(a, stairway_width)]).
argument(c82, a3, lbca, [separating_wall(a)]).
support(c82, a1, p1, [practicable]).
support(c82, a3, p2, [full_compliance]).
oppose(c82, a1, p2, [impracticable]).
oppose(c82, a2, p2, [[provide(a, obstacle)]]).
oppose(c82, a3, p1, [[flout(a, building_warrant)]]).

case(c87, health, [church],
     [churchhall(a), not_used_as(a, storage),
      not_require(a, temperature_control),
      cubic_space_per_occupant(a, 12, cubic_metre), window(b), part_of(b, a),
      floor(c), part_of(c, a), door(d), part_of(d, a)],
     [number(d, 5), area(c, 50, square_metre), area(b, 0.8, square_metre)],
     refused, [m108, m109, m116], [m108, m109, m116]).
position(c87, p1, applicant,
     [area(c, 50, square_metre), area(d, 0.8, square_metre), number(b, 5)]).
position(c87, p2, lbca,
     [fan(e), part_of(e, a),
      rate_of_air_supply_per_hour_per_person(e, A, cubic_metre), A>=17]).
argument(c87, a1, applicant,
     [a is draughty, a is exposed, stage(f), part_of(f, a),
      cupboard(g), part_of(g, a)]).
argument(c87, a2, lbca, [d is small]).
support(c87, a1, p1, [partial_compliance]).
support(c87, a2, p2, [full_compliance, practicable]).
oppose(c87, a2, p1,
     [[not_provide(a, adequate_supply_of_air)], non_compliance]).

case(c90_1, safety, [house],
     [stair(a), stairway(b), used_as(b, exit),
      attribute(b, enclosed), part_of(a, b)],
     [material(a, timber), fire_resistance(a, 1, hour)],
     granted, [m23, m304], [m23]).
position(c90_1, p1, applicant,
     [material(a, timber), fire_resistance(a, 1, hour)]).
position(c90_1, p2, lbca, [material(a, non_combustible)]).
argument(c90_1, a1, applicant, [building(x), part_of(b, x), b is existing]).
argument(c90_1, a2, applicant,
     [building(x), part_of(b, x), has(x, fire_alarm_system),
      has(x, smoke_detector), has(x, tele_nurse_system)]).
argument(c90_1, a3, lbca, [a is new]).
support(c90_1, a1, p1, [practicable]).
support(c90_1, a2, p1, [[building(x), ensure(x, safety_in_case_of_fire)]]).
support(c90_1, a3, p2, [full_compliance]).
oppose(c90_1, a1, p2, [impracticable]).
oppose(c90_1, a3, p1, [non_compliance]).

case(c90_2, safety, [house],
     [division_wall(a), lift(b), separate_from(a, b)],
     [material(a, timber), fire_resistance(a, 1, hour)],
     granted, [m26, m305], [m26]).
position(c90_2, p1, applicant,
     [material(a, timber), fire_resistance(a, 1, hour)]).
position(c90_2, p2, lbca, [material(a, non_combustible)]).
argument(c90_2, a1, applicant, [building(x), part_of(b, x), b is existing]).
argument(c90_2, a2, applicant,
     [building(x), part_of(b, x), has(x, fire_alarm_system),
      has(x, smoke_detector), has(x, tele_nurse_system)]).
argument(c90_2, a3, lbca, [a is new]).
support(c90_2, a1, p1, [practicable]).
support(c90_2, a2, p1, [[building(x), ensure(x, safety_in_case_of_fire)]]).
```

```
support(c90_2, a3, p2, [full_compliance]).
oppose(c90_2, a1, p2, [impracticable]).
oppose(c90_2, a3, p1, [non_compliance]).

case(c90_3, safety, [house],
     [floor(a), stairway(b), attribute(b, enclosed),
      landing(c), part_of(a, c), part_of(c, b)],
     [material(a, timber), fire_resistance(a, 1, hour)],
     granted, [m27, m306], [m27]).
position(c90_3, p1, applicant,
     [material(a, timber), fire_resistance(a, 1, hour)]).
position(c90_3, p2, lbca, [material(a, non_combustible)]).
argument(c90_3, a1, applicant, [building(x), part_of(b, x), b is existing]).
argument(c90_3, a2, applicant,
     [building(x), part_of(b, x), has(x, fire_alarm_system),
     has(x, smoke_detector), has(x, tele_nurse_system)]).
argument(c90_3, a3, lbca, [a is new]).
support(c90_3, a1, p1, [practicable]).
support(c90_3, a2, p1, [[building(x), ensure(x, safety_in_case_of_fire)]]).
support(c90_3, a3, p2, [full_compliance]).
oppose(c90_3, a1, p2, [impracticable]).
oppose(c90_3, a3, p1, [non_compliance]).

case(c93, health, [house],
     [bedroom(a), window(b), part_of(b, a)],
     [number(b, 3), top_height_above_floor(b, 1.8, metre)], `
     granted, [m116, m307], [m116]).
position(c93, p1, applicant, [top_height_above_floor(b, 1.8, metre)]).
position(c93, p2, lbca, [top_height_above_floor(b, A, metre), A>=2]).
argument(c93, a1, applicant,
     [building(c), part_of(b, c), c is eighteenth_century_house]).
argument(c93, a2, lbca, [b is new]).
argument(c93, a3, applicant, [number(b, 3)]).
support(c93, a1, p1, [partial_compliance]).
support(c93, a2, p2, [full_compliance]).
support(c93, a3, p1, [[provide(b, adequate_supply_of_air)]]).
oppose(c93, a1, p2, [impracticable]).
oppose(c93, a2, p1, [non_compliance]).

case(c95, safety, [shop],
     [games_room(a), occupancy_capacity(a, 62), at_level(a, ground)],
     [escape_route(b), part_of(b, a), number(b, 1)],
     granted, [m39, m308], [m39]).
position(c95, p1, applicant, [escape_route(b), number(b, 1)]).
position(c95, p2, lbca, [escape_route(b), number(b, 2)]).
argument(c95, a1, applicant, [building(c), part_of(a, c), c is existing]).
argument(c95, a2, applicant,
     [building(c), main_entrance(d), part_of(d, c), distance(a, d, 11, feet)]).
argument(c95, a3, lbca, [a is new]).
argument(c95, a4, lbca, [occupancy_capacity(a, large)]).
argument(c95, a5, bd,
     [building(c), part_of(a, c), shop(c), occupancy_capacity(a, 62)]).
argument(c95, a6, applicant, [building(c), part_of(a, c), c is small]).
support(c95, a1, p1, [partial_compliance]).
support(c95, a2, p1,
     [[building(c), provide(c, adequate_means_of_escape_from_fire)]]).
support(c95, a3, p2, [practicable, full_compliance]).
support(c95, a4, p2, [reasonable]).
support(c95, a5, p1, [reasonable]).
oppose(c95, a1, p2, [impracticable]).
oppose(c95, a4, p1, [unreasonable]).
oppose(c95, a6, p2, [impracticable]).

case(c97, safety, [warehouse],
     [floor(a), stairway(b), connected(a, b), a is open],
     [attribute(b, unenclosed)],
     granted, [m84, m309], [m84]).
additional_facts(c97,
```

```
     [balustrade(c), stair(d), opening(e), part_of(c, d), part_of(e, c),
      part_of(c, b), part_of(e, b)]).
position(c97, p1, applicant, [attribute(b, unenclosed)]).
position(c97, p2, lbca, [attribute(b, enclosed)]).
argument(c97, a1, applicant,
     [building(c), part_of(a, c), a is open, number_of_storey(c, 1)]).
argument(c97, a2, applicant, [building(c), c is occupied_occasionally]).
argument(c97, a3, lbca, [building(c), part_of(a, c), number_of_storey(c, 2)]).
argument(c97, a4, bd,
     [building(c), exit(d), part_of(d, c), not_part_of(b, d)]).
support(c97, a1, p1, [partial_compliance]).
support(c97, a2, p1, [reasonable]).
support(c97, a3, p2, [full_compliance]).
support(c97, a4, p1, [reasonable]).
oppose(c97, a1, p2, [unreasonable]).
oppose(c97, a3, p1, [non_compliance]).

case(c110, safety, [commercial_building],
     [room(a), part_of(a, b), building(b), heat_producing_appliance(c),
      part_of(c, a), heat_producing_rate(c, 60, kw),
      fuel(c, gas), has(c, low_fire_risk)],
     [not_is(a, a_place_of_special_fire_risk)],
     granted, [m18, m310], [m18]).
position(c110, p1, applicant, [not_is(a, a_place_of_special_fire_risk)]).
position(c110, p2, lbca, [a is a_place_of_special_fire_risk]).
argument(c110, a1, applicant, [has(c, low_fire_risk)]).
argument(c110, a2, lbca, [heat_producing_rate(c, 60, kw)]).
support(c110, a1, p1, [partial_compliance]).
support(c110, a2, p2, [full_compliance, practicable]).
oppose(c110, a1, p2, [unreasonable]).
oppose(c110, a2, p1, [non_compliance]).

case(c116, convenience, [shop],
     [coffee_shop(a), access_by_disabled(b), part_of(b, a),
      sanitary_convenience(c), part_of(c, a)],
     [ambulant_disabled(g), part_of(g, a)],
     granted, [m201, m311], [m201]).
additional_facts(c116, [storey(d), part_of(a, d)]).
position(c116, p1, applicant, [ambulant_disabled(g), part_of(g, a)]).
position(c116, p2, lbca,
     [sanitary_convenience_for_disabled(g), part_of(g, a)]).
argument(c116, a1, applicant, [a is small]).
argument(c116, a2, applicant,
     [restaurant(d), near(d, a), d is large, access_by_disabled(e),
      part_of(e, d), sanitary_convenience_for_disabled(f), part_of(f, d)]).
support(c116, a1, p1, [reasonable, partial_compliance]).
support(c116, a2, p1, [reasonable]).
oppose(c116, a1, p2, [impracticable]).
oppose(c116, a2, p2, [unreasonable]).

case(c131, safety, [house],
     [access_stairway(a), attribute(a, external), stair(b), part_of(b, a),
      landing(c), connected(b, c), height_above_landing(b, 620, millimetre)],
     [upright(d), part_of(d, b), connected_by(d, rope)],
     granted, [m67, m70, m73, m312], [m67, m70, m73]).
position(c131, p1, applicant,
     [upright(d), part_of(d, b), connected_by(d, rope)]).
position(c131, p2, lbca,
     [balustrade(d), part_of(d, b), height(d, A, millimetre), A>=900]).
argument(c131, a1, applicant, [b is disused]).
argument(c131, a2, bd, [not_give_access_to(b, building)]).
argument(c131, a3, lbca, [can(b, be_used_by_visitors)]).
support(c131, a1, p1, [reasonable]).
support(c131, a2, p1, [reasonable]).
support(c131, a3, p2, [full_compliance]).
oppose(c131, a1, p2, [[block(d, doorway)]]).
oppose(c131, a2, p2, [unreasonable]).
```

```
case(c132, convenience, [shop],
     [lounge_bar(a), floor(b), part_of(a, b),
      access_by_disabled(c), part_of(c, b),
      gent_toilet_facility(d), part_of(d, a)],
     [disabled_watercloset(e), part_of(e, d)],
     granted, [m201, m313], [m201]).
additional_facts(c132, [storey(e), part_of(b, e), part_of(a, e)]).
position(c132, p1, applicant, [disabled_watercloset(e), part_of(e, d)]).
position(c132, p2, lbca,
     [sanitary_convenience_for_disabled(e), part_of(e, d)]).
argument(c132, a1, applicant, [a is small]).
argument(c132, a2, applicant, [a is local]).
support(c132, a1, p1, [practicable]).
support(c132, a2, p1, [reasonable]).
oppose(c132, a1, p2, [impracticable]).
oppose(c132, a2, p2, [unreasonable]).

case(c134, health, [warehouse],
     [storage_room(a), fan(b), part_of(b, a)],
     [number(b, 1), rate_of_air_supply_per_hour(b, 250, cubic_metre)],
     granted, [m115, m314], [m115]).
position(c134, p1, applicant,
     [number(b, 1), rate_of_air_supply_per_hour(b, 250, cubic_metre)]).
position(c134, p2, lbca, [number(b, 3)]).
argument(c134, a1, applicant, [a is unmanned]).
argument(c134, a2, lbca, [storage_room(a)]).
support(c134, a1, p1, [partial_compliance]).
support(c134, a2, p2, [reasonable]).
oppose(c134, a1, p2, [unreasonable]).

case(c141, health, [house],
     [watercloset(a), window(b), part_of(b, a)],
     [area(b, 0.16, square_metre), porch(c), next_to(b, c)],
     refused, [m106, m116], [m106, m116]).
position(c141, p1, applicant, [area(b, 0.16, square_metre)]).
position(c141, p2, lbca, [fan(d), part_of(d, a)]).
argument(c141, a1, applicant, [b is large, porch(c), next_to(b, c)]).
argument(c141, a2, lbca, [not_direct_to(b, external_air)]).
support(c141, a1, p1,
     [partial_compliance, [provide(a, adequate_supply_of_air)]]).
support(c141, a2, p2, [full_compliance]).
oppose(c141, a1, p2, [unreasonable]).
oppose(c141, a2, p1,
     [non_compliance, [not_provide(a, adequate_supply_of_air)]]).

case(c150, convenience, [commercial_building],
     [building(a), storey(b), at_level(b, ground), part_of(b, a)],
     [portable_ramp_access(c), part_of(c, b)],
     refused, [m202], [m202]).
additional_facts(c150,
     [storey(d), part_of(d, a), access_to_building(e), part_of(e, a)]).
position(c150, p1, applicant, [portable_ramp_access(c), part_of(c, b)]).
position(c150, p2, lbca, [access_by_disabled(f), part_of(f, b)]).
argument(c150, a1, applicant, [door(g), g is new, part_of(g, b)]).
argument(c150, a2, lbca,
     [door(g), g is new, part_of(g, b),
      height_above_ground(g, 125, millimetre)]).
support(c150, a1, p1, [practicable, partial_compliance]).
support(c150, a2, p2, [full_compliance]).
oppose(c150, a1, p2, [unreasonable]).
oppose(c150, a2, p1, [non_compliance]).

case(c154_1, safety, [shop],
     [storey(a), at_level(a, second)],
     [escape_route(b), part_of(b, a), number(b, 1)],
     granted, [m45, m315], [m45]).
additional_facts(c154_1, [floor(c), part_of(c, a)]).
position(c154_1, p1, applicant,
```

```
     [escape_route(b), part_of(b, a), number(b, 1)]).
position(c154_1, p2, lbca, [escape_route(b), part_of(b, a), number(b, 2)]).
argument(c154_1, a1, applicant, [floor(c), part_of(c, a), c is small]).
argument(c154_1, a2, lbca, [at_level(a, second)]).
support(c154_1, a1, p1, [partial_compliance]).
support(c154_1, a2, p2, [full_compliance]).
oppose(c154_1, a1, p2, [impracticable, unreasonable]).
oppose(c154_1, a2, p1, [non_compliance, [flout(a, building_warrant)]]).

case(c154_2, safety, [shop],
     [storey(a), at_level(a, third)],
     [escape_route(b), part_of(b, a), number(b, 1)],
     granted, [m45, m316], [m45]).
additional_facts(c154_2, [floor(c), part_of(c, a)]).
position(c154_2, p1, applicant,
     [escape_route(b), part_of(b, a), number(b, 1)]).
position(c154_2, p2, lbca, [escape_route(b), part_of(b, a), number(b, 2)]).
argument(c154_2, a1, applicant, [floor(c), part_of(c, a), c is small]).
argument(c154_2, a2, lbca, [at_level(a, third)]).
support(c154_2, a1, p1, [partial_compliance]).
support(c154_2, a2, p2, [full_compliance]).
oppose(c154_2, a1, p2, [impracticable, unreasonable]).
oppose(c154_2, a2, p1, [non_compliance, [flout(a, building_warrant)]]).

case(c156, safety, [house],
     [stair(a), headroom(b), part_of(b, a)],
     [height_above_pitch(b, 1950, millimetre)],
     refused, [m65], [m65]).
position(c156, p1, applicant, [height_above_pitch(b, 1950, millimetre)]).
position(c156, p2, lbca, [height_above_pitch(b, A, millimetre), A>=2050]).
argument(c156, a1, applicant,
     [roof(c), c is low, ridge(d), d is low, part_of(d, c)]).
argument(c156, a2, lbca, [a is new]).
support(c156, a1, p1, [reasonable]).
support(c156, a2, p2, [full_compliance]).
oppose(c156, a1, p2, [impracticable]).
oppose(c156, a2, p1, [non_compliance]).

case(c165, safety, [office],
     [storey(a), occupancy_capacity(a, 70), escape_route(b),
      automatic_sliding_door(c), part_of(c, b), part_of(b, a)],
     [cannot(c, open_outwards)],
     refused, [m56], [m56]).
position(c165, p1, applicant, [cannot(c, open_outwards)]).
position(c165, p2, lbca, [can(c, open_outwards)]).
argument(c165, a1, applicant,
     [fail_open_facility(d), part_of(d, c), conform(d, BS5588)]).
argument(c165, a2, lbca,
     [fail_open_facility(d), part_of(d, c), d is mechanical]).
argument(c165, a3, lbca, [escape_route(e), part_of(c, e)]).
support(c165, a1, p1,
     [partial_compliance, [provide(c, adequate_means_of_escape)]]).
support(c165, a2, p2, [full_compliance]).
support(c165, a3, p2, [full_compliance]).
oppose(c165, a2, p1,
       [non_compliance, [not_provide(c, adequate_means_of_escape)]]).

case(c172, convenience, [shop],
     [building(a), storey(b), at_level(b, ground), part_of(b, a)],
     [concrete_ramp(c), part_of(c, b), remove(c)],
     refused, [m202], [m202]).
additional_facts(c172,
     [storey(d), part_of(d, a), access_to_building(e), part_of(e, a)]).
position(c172, p1, applicant, [concrete_ramp(c), part_of(c, b), remove(c)]).
position(c172, p2, lbca, [concrete_ramp(c), part_of(c, b)]).
argument(c172, a1, applicant,
     [professional_golf_shop(a), not_is(a, open_to_the_public)]).
argument(c172, a2, applicant, [c is slippery]).
```

```
argument(c172, a3, lbca, [shop(a), a is open_to_the_public]).
support(c172, a1, p1, [reasonable]).
support(c172, a2, p1, [reasonable]).
support(c172, a3, p2, [full_compliance]).
oppose(c172, a1, p2, [unreasonable]).
oppose(c172, a3, p1, [non_compliance]).

case(c0012, safety, [high_hazard_storage],
     [building(a), area(a, 1200, square_metre), compartment(b), part_of(b, a),
      has(a, full_automatic_fire_control_system)],
     [area(b, 1200, square_metre)],
     refused, [m14, m41, m42, m43], [m14, m41, m42, m43]).
additional_facts(c0012, [storey(c), part_of(c, a), part_of(b, c)]).
position(c0012, p1, applicant, [area(b, 1200, square_metre)]).
position(c0012, p2, lbca, [area(b, A, square_metre), A=<1000]).
argument(c0012, a1, applicant,
     [b is used_for_temporary_storage,
      b is used_for_dispatch_of_cased_goods,
      has(a, full_automatic_fire_control_system)]).
argument(c0012, a2, lbca, [area(a, 1200, square_metre)]).
argument(c0012, a3, bd, [area(a, 1200, square_metre)]).
support(c0012, a1, p1, [partial_compliance, reasonable]).
support(c0012, a2, p2, [full_compliance]).
support(c0012, a3, p2, [full_compliance]).
oppose(c0012, a1, p2, [unreasonable]).
oppose(c0012, a2, p1, [non_compliance]).
oppose(c0012, a3, p1, [[not_ensure(a, safety_in_case_of_fire)]]).
```

Appendix D

The Domain Knowledge Base

Four Prolog predicates are used to represent background domain knowledge: subpart/2, subclass/2, default_property/2 and rule/3.

subpart(X,Y).
: X is a subpart of Y. subpart/2 is used to represent aggregation hierarchies.

subclass(X,Y).
: X is a subclass of Y. subclass/2 is used to represent specialisation hierarchies.

default_property($Property$, $DefaultProperty$).
: $DefaultProperty$ is a default value of a property $Property$. default_property/2 is used in the similarity assessment to infer property values which are missing in Ccd.

rule(R_{id}, $Preconditions$, $Conclusion$).
: R_{id} is the id number of the rule, $Preconditions$ are known facts and $Conclusion$ is a conclusion that can be hypothesised from $Preconditions$. rule/3 is used to derive new relations from the known facts in the similarity assessment and in the case evaluation.

```
/*--------------------------------------------------
  aggregation relations between building components
--------------------------------------------------*/

subpart(access_balcony, building).
subpart(access_route_to_house, building).
subpart(access_to_building, building).
subpart(apartment, building).
subpart(balustrade, building).
subpart(bathroom, building).
subpart(bedroom, building).
subpart(building_fabric, building).
subpart(building_service, building).
subpart(chapel, building).
subpart(compartment, building).
subpart(conservatory, building).
subpart(corridor, building).
subpart(gent_toilet_facility, building).
subpart(lady_toilet_facility, building).
subpart(lift, building).
subpart(main_entrance, building).
subpart(opening, building).
subpart(passageway, building).
subpart(porch, building).
subpart(room, building).
subpart(stairway, building).
subpart(stairway_enclosure, building).
subpart(storey, building).
subpart(sanitary_convenience, building).
subpart(sanitary_convenience_for_disabled, building).
subpart(washroom, building).
subpart(watercloset, building).

subpart(churchall, building).
subpart(coffee_shop, building).
subpart(games_room, building).
subpart(garble_bedroom, building).
subpart(lounge_bar, building).
subpart(place, building).
subpart(place_of_safety, building).
subpart(restaurant, building).
subpart(storage_room, building).
subpart(access_by_disabled, coffee_shop).
subpart(ambulant_disabled, coffee_shop).
subpart(sanitary_convenience_for_disabled, coffee_shop).

subpart(access_by_disabled, restaurant).
subpart(sanitary_convenience_for_disabled, restaurant).

subpart(floor, lounge_bar).
subpart(gent_toilet_facility, lounge_bar).
subpart(lady_toilet_facility, lounge_bar).

subpart(sanitary_convenience_for_disabled, sanitary_convenience).

subpart(sanitary_convenience_for_disabled, gent_toilet_facility).
subpart(sanitary_convenience_for_disabled, lady_toilet_facility).

subpart(disabled_watercloset, gent_toilet_facility).
subpart(disabled_watercloset, lady_toilet_facility).

subpart(embankment, place).
subpart(place_of_safety, place).
subpart(heat_producing_appliance, place).
subpart(heat_producing_appliance, room).
subpart(ventilator, room).

subpart(wall, building_fabric).
```

```
subpart(roof, building_fabric).
subpart(floor, building_fabric).
subpart(division_wall, building_fabric).
subpart(partition_wall, building_fabric).
subpart(compartment_wall, building_fabric).
subpart(separating_wall, building_fabric).
subpart(loadbearing_wall, building_fabric).
subpart(loadbearing_partition_wall, building_fabric).

subpart(chapel, storey).
subpart(coffee_shop, storey).
subpart(compartment, storey).
subpart(concrete_ramp, storey).
subpart(door, storey).
subpart(floor, storey).
subpart(room, storey).
subpart(ventilator, storey).
subpart(access_by_disabled, storey).
subpart(portable_ramp_access, storey).
subpart(escape_route, storey).

subpart(landing, access_route_to_house).
subpart(stair, access_route_to_house).
subpart(step, access_route_to_house).
subpart(balcony, access_route_to_house).

subpart(enclosing_structure, protected_zone).
subpart(stairway, protected_zone).
subpart(ramp, protected_zone).

subpart(door, escape_route).
subpart(ramp, escape_route).
subpart(automatic_sliding_door, escape_route).

subpart(fail_open_facility, automatic_sliding_door).

subpart(stair, stairway).
subpart(step, stairway).
subpart(landing, stairway).
subpart(tread, stairway).

subpart(floor, landing).

subpart(headroom, tread).

subpart(headroom, stair).
subpart(guarding_wall, stair).
subpart(balustrade, stair).
subpart(railing, stair).
subpart(upright, stair).

subpart(headroom, step).
subpart(guarding_wall, step).
subpart(balustrade, step).
subpart(railing, step).
subpart(upright, step).

subpart(opening, balustrade).
subpart(rail, balustrade).

subpart(stair, access_stairway).
subpart(step, access_stairway).
subpart(stair, escape_stairway).
subpart(step, escape_stairway).

subpart(opening, room).
subpart(window, room).
subpart(door, room).
```

```
subpart(floor, room).
subpart(ceiling, room).
subpart(wall, room).
subpart(fan, room).
subpart(escape_route, room).
subpart(dormer_window, room).
subpart(ventilator, room).
subpart(rooflight, room).
subpart(skylight, room).

subpart(opening, chapel).
subpart(window, chapel).
subpart(door, chapel).
subpart(floor, chapel).
subpart(ceiling, chapel).
subpart(wall, chapel).
subpart(rooflight, chapel).
subpart(skylight, chapel).
subpart(ventilator, chapel).

subpart(window, wall).

subpart(ventilator, roof).
subpart(rooflight, roof).
subpart(skylight, roof).
subpart(ridge, roof).

subpart(window, bedroom).
subpart(floor, bedroom).
subpart(ceiling, bedroom).
subpart(wall, bedroom).
subpart(ventilator, bedroom).
subpart(rooflight, bedroom).
subpart(skylight, bedroom).

subpart(window, gable_bedroom).
subpart(dormer_window, gable_bedroom).
subpart(wall, gable_bedroom).
subpart(ceiling, gable_bedroom).
subpart(floor, gable_bedroom).
subpart(ventilator, gable_bedroom).

subpart(opening, bathroom).
subpart(window, bathroom).
subpart(floor, bathroom).
subpart(ceiling, bathroom).
subpart(wall, bathroom).
subpart(rooflight, bathroom).
subpart(skylight, bathroom).
subpart(ventilator, bathroom).

subpart(opening, washroom).
subpart(window, washroom).
subpart(floor, washroom).
subpart(ceiling, washroom).
subpart(wall, washroom).
subpart(rooflight, washroom).
subpart(skylight, washroom).
subpart(ventilator, washroom).

subpart(opening, watercloset).
subpart(window, watercloset).
subpart(floor, watercloset).
subpart(ceiling, watercloset).
subpart(wall, watercloset).
subpart(fan, watercloset).
subpart(rooflight, watercloset).
subpart(skylight, watercloset).
```

```
subpart(ventilator, watercloset).

subpart(door, churchhall).
subpart(floor, churchhall).
subpart(window, churchhall).
subpart(fan, churchhall).
subpart(stage, churchhall).
subpart(opening, churchhall).
subpart(rooflight, churchhall).
subpart(skylight, churchhall).
subpart(ventilator, churchhall).

subpart(escape_route, games_room).
subpart(floor, games_room).
subpart(wall, games_room).
subpart(ceiling, games_room).
subpart(opening, games_room).
subpart(window, games_room).
subpart(rooflight, games_room).
subpart(skylight, games_room).
subpart(ventilator, games_room).

subpart(fan, storage_room).
subpart(floor, storage_room).
subpart(wall, storage_room).
subpart(ceiling, storage_room).
subpart(window, storage_room).
subpart(opening, storage_room).
subpart(ventilator, storage_room).

/*-------------------------
  specialisation relations
-------------------------*/

%----  categorisation of buildings by purpose

subclass(residential_building, building).
subclass(commercial_building, building).
subclass(assembly_building, building).
subclass(recreational_building, building).
subclass(industrial_building, building).
subclass(storage, building).
subclass(warehouse, building).

subclass(dwelling, residential_building).
subclass(institutional_building, residential_building).

subclass(flat, dwelling).
subclass(maisonette, dwelling).
subclass(house, dwelling).

subclass(hospital, institutional_building).
subclass(nursing_home, institutional_building).

subclass(swimming_pool_building, assembly_building).
subclass(church, assembly_building).
subclass(library, assembly_building).
subclass(school, assembly_building).
subclass(surgery, assembly_building).
subclass(health_centre, assembly_building).

subclass(shop, commercial_building).
subclass(office, commercial_building).

subclass(bingo_hall, recreational_building).
subclass(restaurant, recreational_building).
subclass(public_house, recreational_building).
```

```
subclass(low_hazard_storage, storage).
subclass(high_hazard_storage, storage).

%---- specialisation relations between building components

subclass(upper_storey, storey).
subclass(lower_storey, storey).

subclass(bedroom, room).
subclass(livingroom, room).
subclass(bathroom, room).
subclass(kitchen, room).
subclass(storeroom, room).
subclass(utilityroom, room).
subclass(conservatory, room).
subclass(apartment, room).
subclass(churchhall, room).
subclass(chapel, room).
subclass(games_room, room).
subclass(storage_room, room).
subclass(washroom, room).
subclass(watercloset, room).

subclass(gable_bedroom, bedroom).

subclass(inner_room, room).
subclass(outer_room, room).

subclass(conservatory, outer_room).

subclass(bedroom, apartment).
subclass(livingroom, apartment).
subclass(bathroom, apartment).

subclass(door, opening).
subclass(window, opening).

subclass(window, glazing).

subclass(display_window, window).
subclass(emergency_window, window).
subclass(rooflight, window).
subclass(dormer_window, window).

subclass(window, ventilator).
subclass(grille, ventilator).
subclass(door, ventilator).
subclass(fan, ventilator).

subclass(automatic_sliding_door, door).

subclass(wall, building_fabric).

subclass(separating_wall, wall).
subclass(loadbearing_wall, wall).
subclass(division_wall, wall).
subclass(partition_all, wall).

subclass(compartment_wall, partition_wall).
subclass(compartment_wall, division_wall).
subclass(compartment_wall, separating_wall).

subclass(loadbearing_partition_wall, loadbearing_wall).
subclass(loadbearing_partition_wall, partition_wall).
subclass(loadbearing_partition_wall, division_wall).
subclass(loadbearing_partition_wall, separating_wall).

subclass(tread, stair).
```

```
subclass(gent_toilet_facility, sanitary_convenience).
subclass(lady_toilet_facility, sanitary_convenience).

subclass(disabled_watercloset, sanitary_convenience).
subclass(disabled_watercloset, sanitary_convenience).
subclass(ambulant_disabled, sanitary_convenience).
subclass(sanitary_convenience_for_disabled, sanitary_convenience).

subclass(room, place).

%---- specialisation relations between attributes

subclass('single-glazed',glazed).
subclass('double-glazed',glazed).
subclass('triple-glazed',glazed).

subclass(exposed,external).
subclass('semi-exposed',external).

/*-------------------------------------------
  rules to introduce default property values
-------------------------------------------*/

default_property([Obj,number,_],[RepObj,number,[number(RepObj,1)]],Theta) :-
    is_reparg(Obj,Theta,RepObj).  % Theta contains correspondence relations
                                  % between objects in Mcd and Ccd, and,
                                  % according to Theta, an object RepObj
                                  % corresponds to an object Obj.

default_property([Obj,number_of_storey,_],
                 [RepObj,number_of_storey,[number_of_storey(RepObj,1)]],
                 Theta) :-
    is_reparg(Obj,Theta,RepObj),
    is_known(building(RepObj)).  % a fact building(RepObj) is known

default_property([Obj,occupancy_capacity,_],
                 [RepObj,occupancy_capacity,[occupancy_capacity(RepObj,0)]],
                 Theta) :-
    is_reparg(Obj,Theta,RepObj),
    ( is_known(storage_room(RepObj)) ; is_known(storeroom(RepObj)) ).

/*-----------------------------------------------------
  rules to infer new information from given information
-----------------------------------------------------*/

rule(r1,
    (window(X), [X,height,[height(X,H,metre)]], [X,width,[width(X,W,metre)]],
     bedroom(Y), [Y,floor_area,[floor_area(Y,F,square_metre)]]
    ),
    ([X,percentage_area,[percentage_area(X, 100*H*W/F)]])
    ).

rule(r2,
    (window(X), [X,area,[area(X,A1,square_metre)]],
     floor(Y), [Y,area,[area(Y,A2,square_metre)]]
    ),
    ([X,percentage_area,[percentage_area(X, 100*A1/A2)]])
    ).

rule(r3,
    (part_of(X,Y),part_of(Y,Z)),
    part_of(X,Z)
    ) :- X \== Z.

rule(r4,
    (connected(X,Y),connected(Y,Z)),
    connected(X,Z)
```

```
    ) :- X \== Z.

rule(r5,
    (connected(X,Y)),
    connected(Y,X)
    ).

rule(r6,
    (access_to_building(X),storey(Y),at_level(Y,ground)),
    part_of(X,Y)
    ).

rule(r7,
    (building(X),[X,area,[area(X,N,square_metre)]],storey(Y),part_of(Y,X)),
    [Y,area,[area(Y,N,square_metre)]]
    ).

rule(r8,
    (storey(X),at_level(X,first), floor(Y), part_of(Y,X)),
    ([Y,height_above_ground,[height_above_ground(Y,N,millimetre),N >= 2500]])
    ) :-
    new_variable(N).  % N is a new variable created by the prototype system

rule(r9,
    (storey(X),at_level(X,second), floor(Y), part_of(Y,X)),
    ([Y,height_above_ground,[height_above_ground(Y,N,millimetre),N >= 5000]])
    ) :-
    new_variable(N).

rule(r10,
    (storey(X),at_level(X,third), floor(Y), part_of(Y,X)),
    ([Y,height_above_ground,[height_above_ground(Y,N,millimetre),N >= 7500]])
    ) :-
    new_variable(N).
```

Appendix E

Weak models added to the MKB

This appendix shows 17 weak models which were added into the MKB in Appendix B when 25 test cases were processed by the prototype system. These 17 weak models represents weak requirements generalised from rulings made in 17 relaxed cases. As in the MKB shown in Appendix B, two Prolog predicates, goal/2 and model/6, are used here. Numbers prefixed by an underscore are Prolog variables.

```
model(m300, m6, [building],
      [stair(_338), stairway(_334), used_as(_334, exit),
       attribute(_334, enclosed), part_of(_338, _334),
       building(_315), _315 is small, part_of(_334, _315)],
      [material(_338, hardwood)],
      weak).

model(m301, m7, [residential_building],
      [stairway(_315), used_as(_315, exit)],
      [material(_315, steel), attribute(_315, open)],
      weak).

model(m302, m100, [building],
      [gable_bedroom(_330), height_above_floor(_330, 2070, millimetre),
       dormer_window(_320), part_of(_320, _330)],
      [top_height_above_floor(_320, _310, metre), _310>=1.75],
      weak).

model(m303, m6, [building],
      [loadbearing_partition_wall(_305)],
      [material(_305, combustible)],
      weak).

model(m304, m6, [house],
      [stair(_335), stairway(_331), used_as(_331, exit),
       attribute(_331, enclosed), part_of(_335, _331)],
      [material(_335, timber), fire_resistance(_335, _306, hour), _306=1],
      weak).

model(m305, m6, [house],
      [division_wall(_325), lift(_321), separate_from(_325, _321)],
      [material(_325, timber), fire_resistance(_325, _306, hour), _306=1],
      weak).

model(m306, m6, [house],
      [floor(_339), stairway(_335), attribute(_335, enclosed),
       landing(_326), part_of(_339, _326), part_of(_326, _335)],
      [material(_339, timber), fire_resistance(_339, _306, hour), _306=1],
      weak).

model(m307, m100, [house],
      [bedroom(_334), window(_330), part_of(_330, _334)],
      [number(_330, _321), _321=3,
       top_height_above_floor(_330, _310, metre), _310>=1.8],
      weak).

model(m308, m7, [shop],
      [games_room(_329), occupancy_capacity(_329, 62), at_level(_329, ground)],
      [number(_314, _315), _315>=1, escape_route(_314), part_of(_314, _329)],
      weak).

model(m309, m10, [warehouse],
      [floor(_319), stairway(_315), connected(_319, _315), _319 is open],
      [attribute(_315, unenclosed)],
      weak).

model(m310, m6, [commercial_building],
      [room(_339), part_of(_339, _335), building(_335),
       heat_producing_appliance(_326), part_of(_326, _339),
       heat_producing_rate(_326, 60, kw), fuel(_326, gas),
       has(_326, low_fire_risk)],
      [not_is(_339, a_place_of_special_fire_risk)],
      weak).

model(m311, m200, [shop],
      [coffee_shop(_327), access_by_disabled(_323), part_of(_323, _327),
       sanitary_convenience(_314), part_of(_314, _327)],
```

```
      [ambulant_disabled(_305), part_of(_305, _327)],
      weak).

model(m312, m10, [house],
      [access_stairway(_343), attribute(_343, external),
       stair(_334), part_of(_334, _343), landing(_325), connected(_334, _325),
       height_above_landing(_334, 620, millimetre)],
      [upright(_310), part_of(_310, _334), connected_by(_310, rope)],
      weak).

model(m313, m200, [shop],
      [lounge_bar(_336), floor(_332), part_of(_336, _332),
       access_by_disabled(_323), part_of(_323, _332),
       gent_toilet_facility(_314), part_of(_314, _336)],
      [disabled_watercloset(_305), part_of(_305, _314)],
       weak).

model(m314, m100, [warehouse],
      [storage_room(_330), fan(_326), part_of(_326, _330)],
      [number(_326, _317), _317=1,
       rate_of_air_supply_per_hour(_326, _306, cubic_metre), _306=250],
      weak).

model(m315, m7, [shop],
      [storey(_324), at_level(_324, second)],
      [number(_314, _315), _315>=1, escape_route(_314), part_of(_314, _324)],
      weak).

model(m316, m7, [shop],
      [storey(_324), at_level(_324, third)],
      [number(_314, _315), _315>=1, escape_route(_314), part_of(_314, _324)],
      weak).
```

Appendix F

An example of running a case: case 93

Section 8.2 explained how case 93 is processed by the prototype system and this appendix shows the interaction between the prototype system and the user (i.e., the expert) while case 93 is being processed.

```
| ?- run(c93).

===================================================
KICS---The Prototype System Version 1

The system maintains requirements extracted from
The Building  Standards (Scotland) Regulations 1981
and appeal cases related to it.
===================================================

> A case c93 is given.....
SUBJECT: health
BUILDING CATEGORY: house
IN THE FOLLOWING CIRCUMSTANCES:
the window is part of the bedroom
PROPOSED SOLUTION:
there are 3 windows
the top height above the floor of the window is 1.8 metres

** The following requirements are relevant under the subject health.

1. From K14
BUILDING CATEGORY: building
IN CIRCUMSTANCES:
the opening is part of the room
REQUIREMENTS:
the top height above the floor of the opening is not less than 2 metres

> Do you want to see the explanation of how c93 is relevant to it (y/n)? [y] y

house (c93) is a kind of building (K14)
all components match:
the bedroom (c93) is a kind of the room (K14)
the window (c93) is a kind of the opening (K14)
there are matching relations:
the window is part of the bedroom (c93)
the opening is part of the room (K14)

K14 requires that:
the top height above the floor of the opening is not less than 2 metres
c93 proposed that:
there are 3 windows
the top height above the floor of the window is 1.8 metres
** c93 does not satisfy K14.

==> Press return to continue

> Do you want to see the arguments put forward in this case (y/n)? [y] y

> Arguments put forward in this case are as follows.

The following proposal was suggested by the applicant:
the top height above the floor of the window is 1.8 metres
```

```
The proposal was supported by the following arguments:
the applicant argued that:
the proposal constitutes partial compliance based on the facts that
the window is part of the building
the building is an eighteenth century house
the applicant argued that:
under the circumstances in which
there are 3 windows
the proposal has the following consequences
the window provides adequate supply of air

The proposal was opposed by the following arguments:
the Local Building Control Authority argued that:
the proposal constitutes non-compliance based on the facts that
the window is new

==> Press return to continue

The other proposal suggested was:
the Local Building Control Authority suggested that:
the top height above the floor of the window is not less than 2 metres

The proposal was supported by the following arguments:
the Local Building Control Authority argued that:
the proposal constitutes full compliance based on the facts that
the window is new

The proposal was opposed by the following arguments:
the applicant argued that:
the proposal is an impracticable solution when
the window is part of the building
the building is an eighteenth century house

** Before making a decision about relaxation
please consider the following information.....

1. Principles behind non-compliant requirements are as follows.

Principles behind K14 are as follows:
the building provides adequate supply of air
therefore
the building ensures health of people

==> Press return to continue

2. There are 2 similar cases stored in the case library: c42,c87

CASE: c42
SUBJECT: health
BUILDING CATEGORY: dwelling
IN THE FOLLOWING CIRCUMSTANCES:
the height above the floor of the gable bedroom is 2070 millimetres
the dormer window is part of the gable bedroom
PROPOSED SOLUTION:
```

```
the top height above the floor of the dormer window is 1.75 metres
DECISION MADE:
The case was not compliant with K14.
Relaxation was granted.
Argumentation recorded in this case is as follows.

==> Press return to continue

The following proposal suggested by the applicant was granted:
the top height above the floor of the dormer window is 1.75 metres

The proposal was supported by the following arguments:
the applicant argued that:
the proposal is a reasonable solution when
the top height above the floor of the gable bedroom is 2.07 metres

The proposal was opposed by the following arguments:
the Local Building Control Authority argued that:
the proposal constitutes non-compliance based on the facts that
the dormer window is an opening

==> Press return to continue

The other proposal suggested was:
the Local Building Control Authority suggested that:
the top height above the floor of the dormer window is not less than 2 metres

The proposal was supported by the following arguments:
the Local Building Control Authority argued that:
the proposal constitutes full compliance based on the facts that
the dormer window is an opening

The proposal was opposed by the following arguments:
the applicant argued that:
the proposal is an impracticable solution when
the top height above the floor of the gable bedroom is 2.07 metres

==> Press return to continue

CASE: c87
SUBJECT: health
BUILDING CATEGORY: church
IN THE FOLLOWING CIRCUMSTANCES:
the churchhall is not used as a storage
the churchhall does not require a temperature control
the cubic space per occupant of the churchhall is 12 cubic metres
the window is part of the churchhall
the floor is part of the churchhall
the door is part of the churchhall
PROPOSED SOLUTION:
there are 5 doors
the area of the floor is 50 square metres
the area of the window is 0.8 square metres
DECISION MADE:
```

```
The case was not compliant with K10 and K14.
Relaxation was refused.
Argumentation recorded in this case is as follows.

==> Press return to continue

The following proposal suggested by the applicant was refused:
the area of the floor is 50 square metres
the area of the door is 0.8 square metres
there are 5 windows

The proposal was supported by the following arguments:
the applicant argued that:
the proposal constitutes partial compliance based on the facts that
the churchhall is draughty
the churchhall is exposed
the stage is part of the churchhall
the cupboard is part of the churchhall

The proposal was opposed by the following arguments:
the Local Building Control Authority argued that:
under the circumstances in which
the door is small
the proposal has the following consequences
the churchhall does not provide adequate supply of air
the Local Building Control Authority argued that:
the proposal constitutes non-compliance based on the facts that
the door is small

==> Press return to continue

The other proposal suggested was:
the Local Building Control Authority suggested that:
the fan is part of the churchhall
the rate of air supply per hour per person of the fan is not less than 17
cubic metres

The proposal was supported by the following arguments:
the Local Building Control Authority argued that:
the proposal constitutes full compliance based on the facts that
the door is small
the Local Building Control Authority argued that:
the proposal is a practicable solution when
the door is small

==> Press return to continue

3. The proposal suggested by the applicant is as follows:
there are 3 windows
the top height above the floor of the window is 1.8 metres
It does not conflict with requirements under other subjects.

> Can this case be relaxed (y/n)? [y] y
```

```
** Relaxation is granted and the following weak requirements are acquired.

1. SUBJECT: health
TO ACHIEVE THE GOAL:
the building provides adequate supply of air
BUILDING CATEGORY: house
IN CIRCUMSTANCES:
the window is part of the bedroom
REQUIREMENTS:
the number of the window is 3
the top height above the floor of the window is not less than 1.8 metres

** c93 is added to the case library.

** GOOD BYE.....

yes
| ?-
```

Published Papers

Finally, attached are four published or to-be-published papers which include parts of works described in this thesis.

1. Soon-Ae Yang and Dave Robertson. A case-based reasoning system to support the relaxation of building regulations. To be published in *International Journal of Construction Information Technology.*

2. Soon-Ae Yang, Dave Robertson and John Lee. Use of case-based reasoning in the domain of building regulations. *Proceedings of the Second European Workshop on Case-Based Reasoning, November 1994, Chantilly, France.*

3. Soon-Ae Yang and Dave Robertson. A case-based reasoning system for regulatory information. *Digest of Colloquium on "Case based reasoning: Prospects for applications"*, held in The Institute of Electrical Engineers, London, UK, March 1994.

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