A case-based reasoning system for regulatory information

Soon-Ae Yang and Dave Robertson
Department of Artificial Intelligence
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN, UK
(Email: {soonae,dr}@aisb.ed.ac.uk)

Presented at Colloquium on "Case based reasoning: Prospects for applications", held in The Institute of Electrical Engineers, London, UK, March 1994.

A case-based reasoning system for regulatory information

Soon-Ae Yang and Dave Robertson [1]

We describe a knowledge-intensive case-based reasoning system (KICS) which is under development in the domain of statutory building regulations, using case histories from The Scottish Office's Building Directorate (BD). A key aspect of reasoning in this domain is that much of the interpretation of the regulations is not stated directly in the regulatory documents but is derived from experience in applying them to particular cases.

Regulatory information: There are two major knowledge sources: statutory regulations which also include relevant standards and provisions, and case histories. Rules from statutory regulations act as *strong* constraints on new cases in the sense that a case must comply with regulations which are applicable to it. As interpretation of written regulations changes in time, so rules set out in case histories may or may not apply to new cases. New rules are set out when a new case arise in circumstances unanticipated at the time of legislation or in circumstances in which it is unreasonable for specific rules to be applied and these rules can be met in a less onerous way. In this sense rules set out in case histories act as *weak* constraints to new cases.

As case histories of interpretation of regulations accumulate, these are used to establish precedents, which are reviewed when revising statutory regulations and may be included in new versions of regulations. Arguments about the interpretation of regulations are conducted by means of an appeals procedure, in which an expert makes a decision after (if necessary) consulting case histories relevant to the given case. The argument surrounding each appeal decision is often complex, involving the balancing of a variety of assertions by the parties concerned.

We are developing a system which can assist domain experts in this appeal process. The system does not make any decision about relaxation but is intended to provide the experts with relevant information so that they can arrive at an informed final decision.

Model knowledge base: There are two different knowledge sources but knowledge in both of these basically contains legal rules. Building regulations have a variety of generic objectives (or legislative intentions, e.g. safeguarding public health and safety, and conservation of energy). These objectives are refined and translated into more specific technical requirements. Legal rules described in very abstract terms have broader meaning and can be interpreted in different ways. Refined legal rules described in more detailed and concrete concepts explain how the original rule is to be interpreted. Each legal rule is a statement of what should be done in certain circumstances to achieve a particular objective. Different standards may be given in the same criterion to achieve different objectives.

To represent the hierarchical and multidimensional nature of regulatory information, legal rules are converted into *models* and then an abstraction hierarchy is built for each generic objective. In each abstraction hierarchy, a model M is represented as a set of six parts: $M = (M_{id}, M_{pid}, M_c, M_{cd}, M_{sd}, M_s)$ where M_{id} is the id number of the model, M_{pid} is the id number of the parent model in the abstraction hierarchy, M_c is the category of buildings to which the model applies, M_{cd} is the description of key features of circumstances to which the model applies, M_{sd} is the description of required solutions, and M_s is the strength, i.e., "strong" or "weak". The strength will be given as "strong" if the model is taken from statutory regulations, or as "weak" otherwise. M_{cd} and M_{sd} are described in terms of objects, their attributes and relations between objects. An example of a model is shown on the left in

[1] Department of Artificial Intelligence, University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN, UK (Email: soonae@aisb.ed.ac.uk, dr@aisb.ed.ac.uk)

M_{id}:	m15	C_{id}:	c2
M_{pid}:	m6	C_g:	energy_conservation
M_c:	dwelling, industrial_building, storage, warehouse	C_c:	house
M_{cd}:	window(A), type(A,single_glazed), wall(B), type(B,exposed), in(A,B)	C_{cd}:	window(a), type(a,single_glazed), wall(b), type(b,exposed), in(a,b)
M_{sd}:	percentage_area(A,C), C $\leq$ 15	C_{sd}:	percentage_area(a,16)
M_s:	strong	C_d:	refused
		C_m:	m15
		C_t:	nil

Figure 1: Model m15 and case c2

Figure 1.

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 will be stored in the Case Library. Therefore these abstraction hierarchies serve as indexes to cases stored in the case library as well as the interpretation hierarchies of regulations information.

Case library and argumentation structure: We define the Case Library as a collection of cases processed by the system. In addition to an argumentation network, a case C is represented as a set of eight parts: $C = (C_{id}, C_g, C_c, C_{cd}, C_{sd}, C_d, C_m, C_t)$ where C_{id} is the id number of the case, C_g is the goal given, C_c is the category of the building, C_{cd} is the circumstance description, C_{sd} is the proposed solution, C_d is the decision taken by BD, i.e., "granted" or "refused", C_m is the id number of the model which is relevant to the case, and any other information will be stored in texts in C_t. An example of a case is shown on the right in Figure 1.

Argumentation for the relaxation of specific part(s) of regulations starts with usually two *positions*, one suggested by the applicant and the another one or two suggested by the Local Building Control Authority (the LBCA). In addition to these two *players*, there may be other players who make *arguments* to *support* or *oppose* the suggested positions such as BD itself or the firemaster. The role of BD is to arbitrate between these two main players—the applicant and the LBCA—and decide which position is reasonably acceptable in the context of the given case. Such argumentation is recorded in the Case Library along with cases in the form of the following IBIS-like argumentation structure:

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

where C_{id} is the id number of the case in which position with the id number P_{id} and argument with id number A_{id} are given, Pl is the name of the player, and $Reasons$ are reasons for providing arguments for supporting or opposing a particular position.

Domain knowledge base: Knowledge in the Domain Knowledge Base (the DKB) is used in classifying a given case, i.e., to find models whose M_c and M_{cd} are most similar to C_c and C_{cd} of the given case. Specialisation hierarchies are used to represent building categories and concepts used to describe models and cases. Aggregation hierarchies are used to represent structural knowledge about building components. Examples of specialisation/aggregation hierarchies are shown in Figure 2.

Retrieval and evaluation: To find all information relevant to a given case, the system uses case-based reasoning techniques. A case will be given as $C = (C_{id}, C_g, C_c, C_{cd}, C_{sd})$ where C_{id} is the id number of the case, C_g is the goal given, C_c is the category of the building, C_{cd} is the circumstance description,

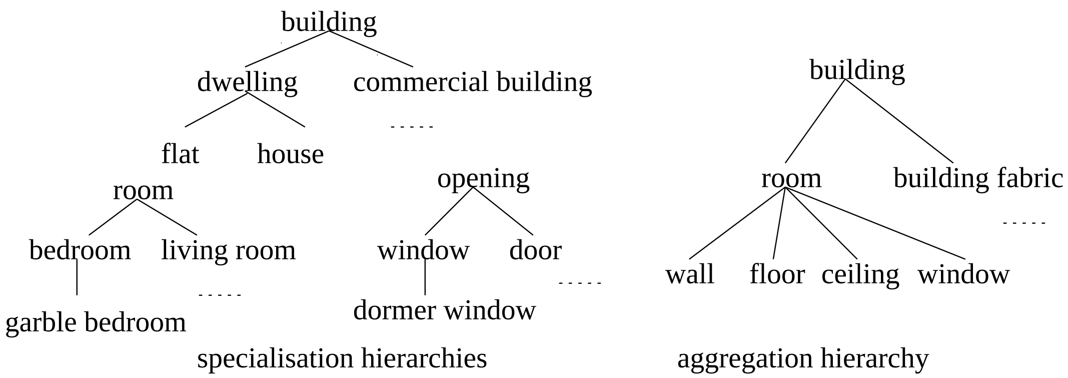

Figure 2: Examples of specialisation/aggregation hierarchy

and C_{sd} is the proposed solution. Given a case C, the system first selects an abstraction hierarchy of the given C_g. Then the system selects only models whose M_c subsumes the C_c of the given case using the specialisation hierarchy of building categories in the DKB. Since models in the selected abstraction hierarchy describe legal rules using terms at different levels of detail, the next step is to find the right level in the abstraction hierarchy which has models comparable to the C_{cd} of the given case. This is done by selecting the level at which objects in M_{cd} are on the same level of aggregation hierarchies as objects in C_{cd}. Once the right level for comparison is found, the system assesses similarity by comparing the details of M_{cd} of selected models with C_{cd} of the given case. The system first applies the notion of structural mapping. If this attempt fails, the system tries to bring out similarity hidden under the surface description of C_{cd} using knowledge stored in the DKB (implied similarity). The system transforms C_{cd} into an alternative description using knowledge in the DKB and repeats the similarity test using the notion of structural mapping.

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

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

Once the human expert decides to grant relaxation, the system will check the validity of the new rules set out by this new case. This will be done by checking whether there is conflict between new rules and rules described in the same criterion in different abstraction hierarchies. If there is no conflict, then the new rules will be integrated into the abstraction hierarchies. First, the system will guess the category of buildings to which the new rules should apply. This is done by looking at the M_c of "strong" sibling models which are similar to the case. Second, the system will collect "weak" sibling models which are relevant to the case and generalise the solutions from the C_{sd} of the case and the M_{sd} of the collected "weak" models. A new model will be created from the hypothesised category, the circumstance description of the case and the generalised solution. The strength will be set to "weak". This new model will replace the "weak" models whose solutions are integrated into the new model.