DANIEL: Integrating Case-Based and Rule-Based Reasoning in Law

Stefanie Brüninghaus
Lehrstuhl für Praktische Informatik I
Universität Mannheim
D - 68131 Mannheim, GERMANY
steffi@pi1.informatik.uni-mannheim.de

Motivation

This paper introduces DANIEL,[1] an architecture for the integration of case-based reasoning and rule-based reasoning for legal interpretation. Rather than interleaving the reasoners and assuming their complementarity, like in previous approaches, they are applied concurrently. Conflicting interpretations are handled explicitly, based on domain knowledge and on the notion of redundancy.

The principal problems of legal interpretation are the lack of deep models for legal reasoning, the existence of inherently ill-defined predicates and the frequent use of open-textured concepts, as pointed out in (Rissland & Skalak 1991). A hybrid approach to representing the legal sources and the use of meta-knowledge seems to be appropriate to solve these problems. The scope of DANIEL is not limited to this particular domain, since the noted difficulties do not occur exclusively, but prototypically in the law.

Architecture and Function of DANIEL

The main knowledge sources in the legal domain are legislation and case law. It has been shown comprehensively in (Rissland & Skalak 1991) that case law and legislation are best mapped on a case base and a rule base, respectively.

In the proposed architecture, the reasoners are integrated via a blackboard, which allows an easy exchange of data and a hierarchical integration of multiple statutes. Since in a given case each knowledge source is likely to contribute to the solution and can be assumed to obtain a result on its own, the problem solvers are applied concurrently and autonomously. Their local results are evaluated by a *rule-based coordination component*, whose meta-knowledge is derived from general legal doctrine, and from the capabilities of the problem solvers. In case of differing local results, the most probable solution is chosen according to the legally determined binding force of the respective legal source, the degree of open-texturedness of the predicates, and the similarity between the given and the retrieved case.

The function of DANIEL can be illustrated by a simplified example: From the statutes, it can be derived that built-in laundry facilities belong to a building, and that therefore the respective expenses must be manufacturing cost. The German Supreme Tax Court, however, decided that a washing machine fixed to the concrete is an extra asset not included in the manufacturing cost of the building. Since the definition of the manufacturing cost is rather open-textured, while the cited case is an exact match, case law has to be applied.

Related Work and Discussion

For space limitations, only the most prominent and obviously very similar system, CABARET (Rissland & Skalak 1991) will be discussed here. It is a domain-independent reasoning shell that incorporates a case-based and a rule-based reasoner via a blackboard. While CABARET uses control heuristics to *interleave* the two reasoners, the reasoners in DANIEL are applied *concurrently*. Also, the coordination component does not work heuristically, but rather it contains domain specific knowledge, in order to overcome the lack of a deep domain model. In other approaches, from various domains, cases are generally considered complementary to rules, and applied accordingly. Even though it can be demonstrated that cases cannot be transformed to rules and vice versa without loss of information, the mutual influence of rules and cases is not explicitly modeled.

Merging the reasoning chains of a case-based and a rule-based reasoner is generally not advisable for the following reasons:

- different binding force/validity of the knowledge sources,
- incompatible partial results of the problem solvers due to their different internal representations and semantics,
- redundant and contradictory knowledge in the problem solvers (only part of it is complementary and disjunct),
- interdependency of the legal sources (existing case law influences future legislation and vice versa).

Apart from avoiding these problems by separating the reasoners, the concurrent application enables their mutual control and takes advantage of their complementarity. In this way, the solution quality can be increased considerably.

References

Rissland, E., and Skalak, D. 1991. CABARET: Rule Interpretation in a Hybrid Architecture. *International Journal on Man-Machine Studies* 34(1):839–887.

[1] Distributed Architecture for the kNowledge-based Integration of Expert systems in Law