Legal Knowledge and Information Systems
A. Wyner and G. Casini (Eds.)

doi:10.3233/978-1-61499-838-9-79

Noise Induced Hearing Loss: An Application of the Angelic Methodology

Latifa Al-Abdulkarim, Katie Atkinson, Trevor Bench-Capon,
Department of Computer Science, The University of Liverpool, UK

Stuart Whittle, Rob Williams and Catriona Wolfenden
Weightmans LLP, Liverpool, UK

Abstract. We describe the use of the ANGELIC methodology, developed to encapsulate knowledge of particular legal domains, to build a full scale practical application for internal use by a firm of legal practitioners. We describe the application, the sources used, the stages in development and the application. Some evaluation of the project and its potential for further development is given. The project represents an important step in demonstrating that academic research can prove useful to legal practitioners confronted by real legal tasks.

1. Introduction

Although AI and Law has produced much interesting research over the last three decades, [6], there has been disappointingly little take-up from legal practice. One important exception is the approach to moving from written regulations to an executable expert system based on the methods proposed in [9], which has been developed through a series of ever larger companies: Softlaw, Ruleburst, Haley Systems and, currently, Oracle[1], where it is known as *Oracle Policy Automation*. Key strengths of Softlaw and its successors were its well defined methodology, and its close integration with the working practices of its customer organisations. In the past year or so, however, there has been an unprecedented degree of interest in AI and its potential for supporting legal practice. There have been many articles in the legal trade press such as *Legal Business*[2] and *Legal Practice Management*[3]; UK national radio programmes such as *Law in Action*[4] and *Analysis*[5] and Professional Society events, such as panels run by the Law Society of England and

[1] http://www.oracle.com/technetwork/apps-tech/policy-automation/overview/index.html
[2] *AI and the law tools of tomorrow: A special report.* www.legalbusiness.co.ukindex.phpanalysis4874-ai-and-the-law-tools-of-tomorrow-a-special-report. All websites accessed in September 2017.
[3] *The Future has Landed.* www.legalsupportnetwork.co.uk. The article appeared in the March 2015 edition.
[4] *Artificial Intelligence and the Law.* www.bbc.co.uk/programmes/b07dlxmj.
[5] *When Robots Steal Our Jobs.* www.bbc.co.uk/programmes/b0540h85.

Wales[6]. At the ICAIL 2017 conference there was a very successful workshop on *AI in Legal Practice*[7]. The legal profession has never been so interested in, and receptive to, the possibilities of AI for application to their commercial activities. There are, therefore, opportunities which need to be taken. In this paper we describe the use of the ANGELIC (ADF for kNowledGe Encapsulation of Legal Information for Cases) methodology [2], developed to encapsulate knowledge of particular legal domains, to build a full scale practical application for internal use by a firm of legal practitioners, to enable mutual exploration of these opportunities.

In section 2 we provide an overview of the law firm for which the application was developed, the domain and the particular task in that domain at which the application was directed. Section 3 gives an overview of the ANGELIC methodology, while section 4 describes the sources used to develop the application. Section 5 discusses the process of capturing and refining the domain knowledge and section 6 the development of an interface to enable the knowledge to be deployed for the required task. Section 7 provides an evaluation of the project and section 8 concludes the paper.

2. Application Overview

The application was developed for Weightmans LLP, a national law firm with offices throughout the UK. Amongst other things, Weightmans act for employers and their insurance companies and advise them when they face claims from claimants for Noise Induced Hearing Loss (NIHL) where it is alleged the hearing loss is attributable to negligence on the part of the employer(s), or former employer(s), during the period of the claimant's employment. Weightmans advise whether the claimant has a good claim in law and, if appropriate, the likely amount of any settlement. Their role is thus to identify potential arguments which the employers or their insurance companies might use to defend or mitigate the claim. Compensators are thus looking to use the ADF primarily to improve how they can settle valid claims and pay proper and fair compensation in a timely manner when appropriate, whilst using the ADF to challenge cases which may have no basis in law or may be otherwise be defendable. In such an application it is essential that the *arguments* be identified. Black box pronouncements are of no use: it is the reasons that are needed. Note that the idea is to identify usable arguments: not to model any process of argumentation. The knowledge will here be deployed in a program not dissimilar from a "good old fashioned" expert system. This seems to meet the current task requirements, which are to support and so speed up decision making. The novelty resides in the methodology, which improves the elicitation process, and the form in which the knowledge is captured and recorded: unlike Softlaw it does not restrict itself to encoding written rules, but draws on other forms of documentation and expert knowledge, which may

[6] The full event of one such panel can be seen on youtube at www.youtube.com/watch?v=8jPB-4Y3jLg. Other youtube videos include Richard Susskind at www.youtube.com/watch?v=xs0iQSyBoDE and Karen Jacks at www.youtube.com/watch?v=v0B5UNWN-eY.

[7] https://nms.kcl.ac.uk/icail2017/ailp.php

include specific experiences such as previous dealings with a particular site and common sense knowledge, and structures this knowledge. In this way the ADF is not restricted to specific items of law, or to particular precedents, but can capture the wider negligence principles that experts distill from the most pertinent decisions. Since the knowledge encapsulated is a superset of what is produced in the CATO system [4], it could, were the task teaching law students to distinguish cases, equally well be deployed in that style of program. Note too that the analysis producing the knowledge, as in CATO, is performed by a human analyst and then applied to cases: the knowledge is not derived from the cases, nor is it a machine learning system.

3. Methodology Overview

The ANGELIC methodology builds on traditional AI and Law techniques for reasoning with cases in the manner of HYPO [5] and CATO [4] and draws on recent developments in argumentation, in particular Abstract Dialectical Frameworks (ADFs) [7] and ASPIC+ [11]. Formally ADFs form a three tuple: a set of nodes, a set of directed links joining pairs of nodes (a *parent* and its *children*), and a set of acceptance conditions. The nodes represent statements which, in this context relate to issues, intermediate factors and base level factors. The links show which nodes are used to determine the acceptability of other nodes, so that the acceptability of a parent node is determined by its children. The acceptance conditions for a node states how precisely its children relate to that node. In ANGELIC the acceptance conditions for non-leaf nodes are a set of individually sufficient and jointly necessary conditions for the parent to be accepted or rejected. For leaf nodes, acceptance and rejection is determined by the user, on the basis of the facts of the particular case being considered. Essentially the methodology generates an ADF, the nodes and links of which correspond to the factor hierarchy of CATO [4]. The acceptance condition for a node contain a prioritised set of sufficient conditions for acceptance and rejection and a default. Collectively, the acceptance conditions can form a knowledge base akin to that required by the ASPIC+ framework [10], but distributed into a number of tightly coherent and loosely coupled modules to conform with best software engineering practice [12]. Thus the acceptance conditions are used to generate arguments, and the ADF structure to guide their deployment.

The methodology is supported by tools [3] developed in parallel with, and informed by, this project to guide the knowledge acquisition, visualise the information, record information about the nodes such as provenance, and to generate a prototype to enable expert validation, and support refinement and enhancement. Once the knowledge is considered acceptable, a user interface is developed, in conjunction with those who will use the system in practice, to facilitate the input of the information needed for particular cases and present the results needed to support a particular task.

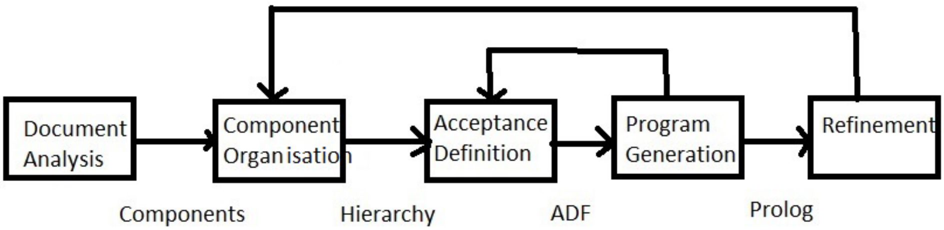

Figure 1. Knowledge Acquisition and Representation Stages

4. Sources

Several sources which were supplied by Weightmans were used to provide the knowledge of the Noise Induced Hearing Loss domain to which the ANGELIC methodology was applied.

- **Experts**: Weightmans made available domain experts to introduce the domain, provide specific documents and to comment on and discuss the developing representation.
- **Documents:** The documents included a 35 page information document produced by Weightmans for their clients, an 18 question check list produced by Weightmans to train and guide their employees and a number of anonymised example cases illustrating different aspects of the domain.
- **Users**: Potential users of the system were made available to assist in building and refining the interface.

Each of these sources played an invaluable role at various stages of the knowledge representation process, each making useful and complimentary contributions by providing different perspectives on the domain.

5. Representing the Knowledge

Following an introductory discussion of the domain, the workflow of the knowledge representation process comprises five steps, as shown in Figure 1:

1. Analyse the available documents and identify components and links between them.
2. Organise the components into an ADF.
3. Define the acceptance conditions for these components; the initial ADF is then reviewed by Weightmans and updated to accommodate the changes provided by the domain experts.
4. Extract a Prolog program from the acceptance conditions.
5. Run the program on the example cases to confirm the structure can generate the arguments in those cases and identify any necessary modification.

These stages are further described in the following sections.

5.1. Document Analysis

The initial discussion with the domain experts provided an excellent orientation in the domain and the key issues. These issues included the fact that claims were time limited, and so had to be made within 3 years of the claimant becoming aware of the hearing loss. Both actual awareness, usually the date of an examination, and constructive awareness (the date on which the claimant should have been aware that there was a problem) need to be considered. Then there is a question of the nature of the hearing loss: there are many reasons why hearing deteriorates, and only some of them can be attributed to exposure to noise. Then there is the possibility of contributory negligence: there is a Code of Practice with which the employers should have complied, and it is also possible that the employee was in part to blame, by not wearing the ear defenders provided, for example.

Next, the information document was used to identify the components that would appear in the ADF, putting some flesh on the skeleton that emerged from the initial discussion. The main document provides summaries of the main definitions, the development of the legal domain rules, the assessment of general damages for noise induced hearing loss cases, and Judicial College Guidelines for the assessment of general damages. Other medical conditions related to hearing loss are listed and described. At this stage components were identified, and where these were elaborated in terms of the conditions that were associated with them, links between these components could be identified. For example hearing loss can be sensioneural, but can also be attributed to a number of other factors: natural loss through aging, loss accompanying cardio-vascular problems, infections, certain drugs, etc. Only sensioneural loss can be noise induced, and so hearing loss arising from the other factors cannot be compensated. The document gives an indication of the various different kinds of hearing loss, and then further information of what may cause the various kinds of loss.

At the end of this phase we have a number of concepts, some of which are elaborated in terms of less abstract concepts, and some potential links. The next step is to organise these concepts in an appropriate structure.

The check list was kept back to be used after the concepts had been organised into a hierarchy, to determine whether the hierarchy bottomed out in sensible base-level factors. The check list comprised a set of 18 questions and a "traffic light" system indicating their effect on the claim. The idea was to associate base level factors with the answers to these questions. For example Question 1 asks whether the exposure ceased more than 3 years before the letter of claim: if it did not, the claim is *ipso facto* within limitation and other kinds of defence must be considered.

Similarly the cases were not used to build the initial ADF but were held back to provide a means of working through the ADF to check that the arguments deployed in those cases could be recovered from the ADF.

5.2. Component Organisation

The main goal now is to move from unstructured information gathered from the documentation to structured information. The main issues had been identified in the initial discussion and the document analysis. These were used to identify and cluster the relevant intermediate predicates from the documents. These nodes were further expanded as necessary to produce further intermediate predicates and possible base level factors. The checklist was then used to identify, and where necessary add, base level factors. The documents from the sample of particular cases were used to provide examples of possible facts, and the effect these facts had on decisions. The result was a factor hierarchy diagram where the root shows the question to be answered, while the leaves show some facts from the sample cases. All this was recorded in a table that described the factors in the domain and their related children. For example, the Breach of Duty factor includes:

Factor: Breach of duty
Description: The employer did not follow the code of practice in some respect.
Children: Risk assessments were undertaken; employee was told of risks; methods to reduce noise were applied, protection zones were identified, there was health surveillance, training.

The children are the main things required of an employer under the code of practice, and so provide a list of the ways in which a breach of duty might have occurred. They may be further elaborated: for example noise reduction includes measures such as shielding the machinery and providing appropriate ear protection.

The final version of the ADF contains 3 issues, 20 intermediate nodes and 14 base level factors, with 39 links. For comparison, the ADF equivalent of CATO given in [2] contained 5 issues, 11 intermediate nodes, and 26 base level factors with 48 links. Thus CATO is larger, but NIHL has more internal structure. The nodes in the visual presentation of the ADF are annotated to show their provenance (the document and section in which they are defined or explained), and any of the checklist questions to which they relate.

5.3. Defining Acceptance Conditions

Once the nodes had been identified, acceptance conditions providing sufficient conditions for acceptance and rejection of the nodes in terms of their children were provided. These were then ordered by priority and a default provided. The particular cases were used to confirm that the arguments used in them could be recovered from the ADF. Continuing the Breach of Duty example:

Factor: Breach of duty
Acceptance conditions: Employee was not told of risks through the provision of education and training,
There were no measures taken to reduce noise,
Protection zones were not identified,
There was no health surveillance and no risk assessment.

Any of these are sufficient conditions to identify a breach of duty. If none of them apply to the case, we can assume, as a default, that there was no breach of duty, and so include rejection of the node as the default.

After this stage, the analysts and domain experts met to discuss and revise the initial ADF. Once a final ADF had been agreed, a Prolog program was produced from the acceptance conditions to suggest whether, given a set of facts, there might be a plausible defence against the claim.

5.4. Program Implementation

The program is implemented using Prolog. The program was created by ascending the ADF, rewriting the acceptance conditions as groups of Prolog clauses to determine the acceptability of each node in terms of its children. This required re-stating the tests using the appropriate syntax. Some reporting was added to indicate whether or not the node is satisfied, and through which condition. Also some control was added to call the procedure to determine the next node, and to maintain a list of accepted factors. We do not give the output here for reasons of space and commercial sensitivity, but its form is identical to that produced for Trade Secrets in [2]. The closeness between Prolog procedures and expressions of the acceptance conditions, each condition mapping to a clause within the Prolog procedure, makes the implementation quick, easy and transparent. The process of moving from acceptance conditions to Prolog code is essentially a mechanical rewriting into a template (supplying the reporting and control) and so is highly amenable to automation. Automated generation of the Prolog program from the ADF is planned as part of the development of the ANGELIC environment [3]. The program operates by:

- Instantiating the base level factors using the case facts;
- Working up the tree. Nodes are represented as heads of clauses, and each acceptance condition forms the body of a clause for the corresponding head, determining acceptance or rejection, with the set of clauses for the head completed by a default [8]. The program reports the status of the node and the particular condition which led to this status before moving to the next node.

The program provides a very transparent output that identifies precisely the path up the hierarchy and hence where any divergences from the expected outcomes occur. The program has been tested on a range of cases (additional to those originally supplied) identified by their base factors to evaluate the output and help the analysts and experts in detecting any errors or potential improvements.

5.5. Refinement

Both the initial ADF and the program were, again, shown to and discussed with the domain experts, who suggested corrections and enhancements. The corrections varied: some suggestions were made about considering missing information from the document, modifying the interpretation of existing acceptance conditions, or adding base factors or new parents to base facts. No changes were related to the main issues or intermediate predicates. As stated in [1], responding to changes in ANGELIC can be easily controlled since the changes affect nodes individually but, because of the modularisation achieved by the ADF, do not ramify through the rest of the structure. Refinement was an iterative process which was repeated until an ADF acceptable to the domain experts was obtained.

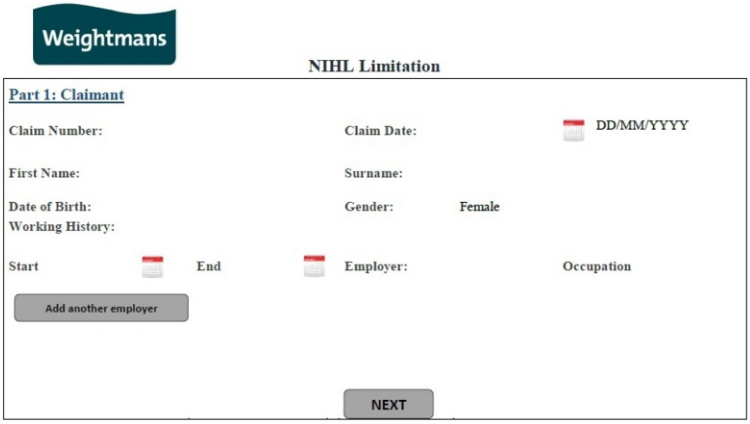

Figure 2. Screen for User Interface

6. User Interface

The ADF encapsulates knowledge of the domain, but this is required not for its own sake, but to add support in handling the analysis of the cases in the legal domain. To fulfill this task, a forms-based interface was designed in conjunction with some of the case handlers who carry out the task and so are the target users of the implemented system.

- The interface is designed to take as input the base level factors which correspond to the questions in the checklist used by the case handlers.
- These questions are organized in an order which makes good sense in terms of the task. First the questions related to the claimant's actual knowledge of the hearing loss are displayed. The answers to questions are used to limit the options provided in later questions and, where possible, to provide automatic answers to other questions.
- Three to four questions are used per screen to maintain simplicity.
- The input to the questions is as a drop down list with the given options (facts), or radio buttons when one option needs to be selected, or checklists for multiple options. Text boxes are also provided to input information particular to individual cases, such as names and dates, or to allow further information for some questions.
- All the questions must be answered, but default answers can be provided for some questions.

The designed interface enables the ADF to assist as a decision augmentation tool for the particular task. A sample screen is shown in Figure 2: the gender is pre-completed, but can be changed from a drop down menu.

7. Evaluation

Developing the application was intended to realise a number of goals, each offering a perspective for evaluation. Note that the system has not been fielded: it was intended as a feasibility study and the programs are prototypes. Validation of its practical utility must await the fielding of a robust system engineered for operational use.

- The ANGELIC methodology had previously been applied only to academic examples. The desire here was to see whether it would also be effective when applied to a reasonably sized, independently specified, domain, intended to produce a system for practical use.
- Weightmans wished to come to a better understanding of the technology and what it could do for them and their clients.
- The methodology was designed to encapsulate knowledge of the domain using techniques representing the state of the art in computational argumentation. It was desired to see whether a domain encapsulated in this way could be the basis for a particular, practically useful, task in that domain.

Each of these produced encouraging results. The methodology proved to be applicable to the new domain without significant change, and could be used with the sources provided. Some desirable additional information that should be recorded about the nodes (such as provenance) was identified. The result was the specification and development of a set of tools to record and support the use of the methodology - the ANGELIC Environment [3].

Weightmans were encouraged that these techniques could prove useful to their business, and are currently exploring, with the University of Liverpool and others, options to take their investigations further.

From the academic standpoint, as well as confirming the usefulness and applicability of the ANGELIC methodology, the customisation for a particular task showed that the general knowledge encapsulated in the ADF can be deployed for a specific task by the addition of a suitable interface.

8. Concluding Remarks

The application of the ANGELIC methodology to a practical task enabled the academic partners in this project to demonstrate the utility of the methodology and identify possible extensions and improvements. The legal partners in the project were able to improve their understanding of the technology, what it could do for their business, and what development of an application would require of them. For the kind of application described here, the argumentation is all-important: the system is not meant to make a decision as to, or a prediction of, entitlement. Rather the case handlers are interested in whether there are any plausible arguments that could be advanced to challenge or mitigate the claim, or whether the arguments suggest that the claim should be accepted.

We believe that the success experienced for this task and domain is reproducible and look forward to using the methodology and supporting tools to build further applications, and to evaluating their practical utility when fielded. It should however, be recognised that the application developed here addresses only part, albeit a central part, of the pipeline. There is still a gap between the unstructured information which appears in a case file and the structured input necessary to drive the program. In the above application this step relies on the skills of the case handlers, but there are other developments which could potentially provide support for this task, such as the tools developed by companies such as Kira Systems for contract analysis and lease abstraction[8]. It is to be hoped that this kind of machine learning tool might provide support for this aspect of the task in future. Similarly the interface is currently hand crafted and one off. It is likely that the process of developing a robust implementation from the animated specification provided by the Prolog program could benefit from tool support, such as that available from companies such as Neota Logic[9]. What has been described is essentially an exploratory study, but one which provides much encouragement and suggests directions for further exploration, and the promise that eventually robust decision support tools based on academic research will be used in practice.

References

[1] L Al-Abdulkarim, K Atkinson, and T Bench-Capon. Accommodating change. *Artificial Intelligence and Law*, 24(4):1–19, 2016.

[2] L Al-Abdulkarim, K Atkinson, and T Bench-Capon. A methodology for designing systems to reason with legal cases using Abstract Dialectical Frameworks. *Artificial Intelligence and Law*, 24(1):1–49, 2016.

[3] L Al-Abdulkarim, K Atkinson, and T Bench-Capon. Angelic environment: Demonstration. In *Proceedings of the 16th International Conference on Artificial Intelligence and Law*, pages 267–268. ACM, 2017.

[4] V Aleven. *Teaching case-based argumentation through a model and examples*. PhD thesis, University of Pittsburgh, 1997.

[5] K Ashley. *Modeling legal arguments: Reasoning with cases and hypotheticals*. MIT press, Cambridge, Mass., 1990.

[6] T Bench-Capon, M Araszkiewicz, K Ashley, K Atkinson, F Bex, F Borges, D Bourcier, P Bourgine, J Conrad, E Francesconi, et al. A history of ai and law in 50 papers: 25 years of the international conference on AI and Law. *Artificial Intelligence and Law*, 20(3):215–319, 2012.

[7] G Brewka, S Ellmauthaler, H Strass, J Wallner, and S Woltran. Abstract dialectical frameworks revisited. In *Proceedings of the Twenty-Third IJCAI*, pages 803–809. AAAI Press, 2013.

[8] K Clark. Negation as failure. In *Logic and data bases*, pages 293–322. Springer, 1978.

[9] P Johnson and D Mead. Legislative knowledge base systems for public administration: some practical issues. In *Proceedings of the 3rd ICAIL*, pages 108–117. ACM, 1991.

[10] S Modgil and H Prakken. The ASPIC+ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62, 2014.

[11] H Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2):93–124, 2010.

[12] R Pressman. *Software engineering: a practitioner's approach*. Palgrave Macmillan, 2005.

[8]see https://kirasystems.com/

[9]https://www.neotalogic.com/