Support for Factor-Based Argumentation

Latifa AL-ABDULKARIM, Katie ATKINSON, Trevor BENCH-CAPON
Department of Computer Science, The University of Liverpool, UK

Abstract. In this paper we describe a tool which supports the analysis of arguments in the legal domain for the purpose of building computational models that use factor-based reasoning (FBR).

Keywords. Knowledge engineering, case-based reasoning, legal argumentation.

1. Introduction

In AI and Law, FBR is used to represent a style of argumentation based on legal cases, where precedents provide the reasons governing how a case is decided. Cases are represented as sets of *factors*. A factor represents an abstraction from the *facts* of the case, which allows cases to be compared. Factors may have parents which are at an increasing level of abstraction, and the presence or absence of their children provide reasons for their own presence or absence, this relation is used to form a *factor hierarchy*. The top level in this hierarchy represents the *issues* in the case, while the base level shows the case *facts*. In the U.S. Supreme Court (see Court Process in [1]), the opinion (case decision) uses *factors* to construct the arguments, but, it does not always identify their purpose or the conditions under which they apply. This can be investigated from the analysis of the oral hearing dialogues (in which counsel for the parties present and clarify their arguments) which will typically comprise three dialogues: between the petitioner and the Justices, between the respondent and the Justices and the petitioner's rebuttal.

Here we present a program to support the analysis of these dialogues to establish the components (issues, factors and facts) from which to construct the arguments in the opinion. Previously [2] presented a set of speech acts which can be used to represent the oral hearings dialogues to build up Argument Components Trees (ACTs) showing the argument components that should be used in the case.

2. Factor-Based Argumentation Support Tool

To construct a tool that provides support for analysing factor-based argumentation, we identify the method shown in Figure1. First we need to identify a suitable set of cases to represent a domain, and proceed through the cases in chronological order. Initially we will have no components, but as we build up our knowledge of the domain, each new case will use components from previous cases, introduce

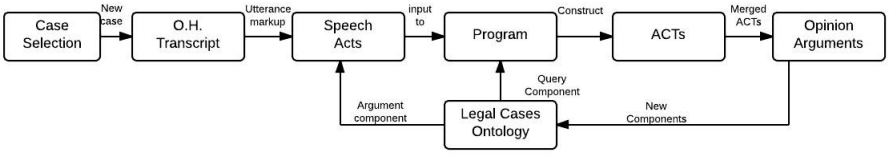

Figure 1. Legal Case Analysis Method

new components and relate them to existing components. After that, for each case the analyst marks each utterance in the transcript of the oral hearing with the appropriate speech act taken from [2], identifying the content of the speech act either from the existing components recorded in the ontology or introducing a new component from the current case. Once the transcript has been marked up, the sequence of speech acts and their contents can be input to a program, to effectively reproduce the dialogues using the restricted vocabulary. The program will process the sequence of speech acts to build the various component trees, corresponding to petitioner, respondent and Justices' perspectives.

Speech acts We defined a number of dialogue moves to enable the proposal of new components (*AssertComponent move*), challenging existing components (*CombineComponent move*), emphasising specific component (*Emphasise move*)and other moves (see [1]). Moves can be made either explicitly by a participant or implicitly, triggered by another move. Each move shows the participant, the content, the running case and dialogue, controlled by pre and post condition(s).

Ontology The ontology is primarily used to record the argument components and the relations between them as the analysis develops, but additionally records some other potentially useful information about the cases and the dialogues so as to enable the provenance of the various components to be traced. The ontology is built in the Web Ontology Language (OWL) using the standard open source ontology editor, Protégé.

Program A Java program using Jena API for ontology management has been produced. At this stage, the program takes a sequence of speech acts (input), and constructs the corresponding ACTs (output). The program presents two ACTs for the petitioner and Justices representing the components from the petitioner dialogue (and petitioner rebuttal dialogue), and two other ACTs, representing the respondent and Justice respectively in the respondent dialogue. These are then passed to a program such as *Graphviz* which produces the ACTs in graphical form.

References

[1] Latifa Al-Abdulkarim, Katie Atkinson, and Trevor J. M. Bench-Capon. Dialogues in US supreme court oral hearings. In *Proceedings of CMNA 2013*, 2013. In press, see: http://www.cmna.info/CMNA13.

[2] Latifa Al-Abdulkarim, Katie Atkinson, and Trevor J. M. Bench-Capon. From oral hearing to opinion in the us supreme court. In *Proceedings of JURIX 2013*, pages 1–10, 2013.