Structured Consultation with Argument Graphs

Thomas F. Gordon
Fraunhofer FOKUS, Berlin
thomas.gordon@fokus.fraunhofer.de

June 19, 2013

Abstract

This article presents the Carneades opinion formation and polling tool, which was inspired by work by Katie Atkinson, Trevor Bench-Capon and Adam Wyner at the University of Liverpool on the Structured Consultation Tool (SCT) they developed in the European IMPACT project. The Carneades polling tool generalises and extends their results by using argument graphs to support consultations about any argument, independent of the argumentation schemes used to reconstruct the arguments, by collecting feedback on the arguments put forward on all sides of a debate, rather than only the arguments of a single position of one party, such as the government agency proposing some policy, and by providing a convenient way for respondents to rank stakeholders by the extent to which they share opinions. Argument graphs abstract away details of the argumentation schemes used to construct or reconstruct the arguments but not needed for the purposes of conducting the poll. Moreover, Carneades provides a high-level declarative language for argumentation schemes, enabling humanities scholars, such as lawyers or argumentation experts, to define and configure the set of argumentation schemes to be used to construct the argumentation graphs, without requiring technical computer-science skills or modifications to the implementation of the polling tool.

1 Introduction

It is a great personal pleasure for me to be able to contribute to this Festschrift in honor of Trevor Bench-Capon. We have known each other for many years in the field of Artificial Intelligence and Law, having both participated in many, perhaps most, ICAIL and Jurix conferences, and have worked together rather intensively since 2006, in a couple of European research projects, ESTRELLA and IMPACT. I believe we first met 26 years ago at the first ICAIL conference, which took place in Boston in 1987. Trevor and his colleagues presented *two* papers on modeling legislation using logic programming [6, 7], an approach I had been constructively criticizing for its insufficient support for isomorphic modeling of exceptions [14, 16] and made an early attempt to rectify with the

Oblog system [15] I presented at the same conference. Trevor's work on modeling dialogue games [8, 9] directly influenced my research on the procedural aspects of legal argumentation in dialogues, for my doctoral thesis on the Pleadings Game, first presented at the 1993 ICAIL in Amsterdam [17]. I have always enjoyed Trevor's company and good humor and consider him a good friend. And I don't want to neglect to take this opportunity to thank Trevor for his generous support over the years, in particular for his positive and helpful review of my habilitation thesis [18], for which I am extremely grateful and will always be in his debt.

My contribution here, however, was inspired by more recent work by Trevor and his colleagues Katie Atkinson and Adam Wyner on the Structured Consultation Tool (SCT) they developed in the European IMPACT project [1, 26]. I present a fully implemented polling tool, based on the Carneades argumentation system, which generalizes and extends their results by using argument graphs to support consultations about any argument, independent of the argumentation schemes used to reconstruct the arguments, by collecting feedback on the arguments put forward on all sides of a debate, rather than only the arguments of a single position of one party, such as the government agency proposing some policy, and by providing a convenient way for respondents to rank stakeholders by the extent to which they share opinions. The polling tool advances the state-of-the-art by using argument graphs to abstract away details of the argumentation schemes used to construct or reconstruct the arguments but not needed for the purposes of conducting the poll. This enables the tool to be used to conduct polls about all arguments in an argument graph, no matter which argumentation schemes have been applied to construct the graph. The process of applying argumentation schemes to construct the argument graph is cleanly separated from the process of using the argument graph to generate poll questions. Moreover, Carneades provides a high-level declarative language for argumentation schemes, enabling humanities scholars, such as lawyers or argumentation experts, to define and configure the set of argumentation schemes to be used to construct the argumentation graphs, without requiring technical computer-science skills or modifications to the implementation of the polling tool.

The Carneades argumentation system provides web-based, collaborative software tools for:

- reconstructing the arguments of a debate in an argument graph
- visualizing, browsing and navigating argument graphs
- critically evaluating arguments
- forming opinions, participating in polls and ranking stakeholders by the degree to which they share your views
- obtaining clear explanations, using argument graphs, of the different effects of alternative policies in particular cases

Carneades is open source software, available for downloading from `http://carneades.github.com`.

The focus of this paper is the polling tool of the Carneades system, which serves two main purposes:

1. It guides users step by step through the arguments on all sides of a complex policy debate, in a kind of simulated debate, providing an overview of the issues, positions and arguments in a systematic way. The tool can help users to form an opinion, if they do not yet have one, or to critically evaluate and reconsider their current opinion, if they do. The tool also enables users to compare their opinions with the published opinions of stakeholders, such as political parties, which can be useful for finding persons and organizations which best represent or share their views and interests.

2. At the same time the tool conducts a poll to collect and aggregate views and opinions on the issues of a debate, taking care to protect privacy. The anonymous and aggregated results of the poll can provide valuable feedback, to the respondents and policy makers, going beyond the information provided by typical surveys and polls. It enables users to discover not only how much support policies enjoy, but also to learn precisely why particular aspects of the policies, or their underlying assumptions, are supported or not.

The rest of this paper presents an analysis of requirements for the polling tool, the algebraic model of argument graphs underlying all tools of the Carneades system, an overview of the design and implementation of the polling tool, a tour of its user interface, and a discussion comparing the results with prior, related work, in particular work in the European IMPACT project by Trevor, Katie and Adam at the University of Liverpool on the Structured Consultation Tool.

2 Requirements Analysis

Following an *agile* methodology [21], the functional requirements of the polling tool are defined here via *user stories*. User stories are brief, high-level statements describing users in particular roles (who) that would like to be able to use the system to perform some task (what) in order to achieve some benefit or value (why). They are typically specified by filling in templates, such as "As a *role*, I want to *action*, in order to *value*." There are various versions of these templates, but their differences are minor and not significant for our purposes.

Let us focus on the application scenario of the IMPACT project: supporting public policy deliberations. In this scenario, first a government agency publishes a green or white paper on the Web regarding some policy topic, such as "copyright in the knowledge economy".[1] Whereas green papers ask questions about some policy topic, without proposing a specific policy, white papers do

[1] See `http://ec.europa.eu/internal_market/copyright/docs/copyright-infso/greenpaper_en.pdf`.

propose a specific policy.[2] For both types of papers, at the time of publication, the agency invites interested parties to submit comments, proposals and arguments, by uploading documents in PDF format to the agency website. At the end of the commenting period, the agency analyses the comments and produces a report summarizing the arguments contained in the comments, along with any decisions taken by the agency as a result of the consultation process, which is then published on its website.

The question which interests us is how to use argumentation technology to support and improve this consultation process. Here we will focus mainly on the requirements for the polling tool, which would provide interested persons with an opportunity to learn about and evaluate claims and arguments put forward during the consultation process, both by the government agency and by the parties who submitted comments. One goal is to systematically generate polling questions from an argument graph containing reconstructions of arguments in the paper and comments, taking care to assure that all relevant critical questions are asked. The polls are conducted only after some arguments have been reconstructed in an argument graph, using argumentation schemes to guide the reconstruction process. If the initial paper contains arguments, for example arguments in a white paper used to justify the proposed policy, a poll could be conducted soon after publication of the paper, before any comments have been submitted. Another alternative would be to wait until the commenting period has expired to conduct the poll, to also collect feedback about the arguments put forward in the comments. A further alternative would be to conduct several polls during the consultation process, using the arguments which have been put forward thus far in the process at each stage.

Several roles can be identified in this scenario:

Agency. The government agency which published the green or white paper and manages the consultation process.

Analyst. The persons who have the job of using argumentation schemes to reconstruct the arguments in the paper and comments, to build the argument graph. Analysts are presumed to have had some training in how to reconstruct arguments, over a period of weeks, including how to use argumentation software tools designed to support this tasks.

Commentator. A person or organization who submits a comment, putting forward arguments in response to the green or white paper. Commentators are presumed to have some knowledge about the policy issue being debated, but no specialist knowledge about argumentation theory or information technology.

Respondent. Persons who take part in the polls, to express their opinions about the claims and arguments exchanged by the agency and the commentators. It is presumed that these persons have no specialist knowledge

[2] For a description of the distinction between green and white papers, see `http://en.wikipedia.org/wiki/Green_paper`.

in argumentation, information technology, the policy domain being discussed, or any other field, and are unwilling to invest any time in learning how to use the polling software.

Humanities Scholar. Philosophers and others with the specialist knowledge required to formalize argumentation schemes.

Given this consultation scenario, with these roles, functional requirements for the polling tool can be formulated in the following user stories:

1. As an agency, I want to obtain feedback from the public with their opinions on the claims and arguments put forward in a green or white paper, as well as in the comments submitted during the consultation, in order to understand which policy proposals are acceptable or not by the public, along with the reasons for their opinions.
2. As an analyst, I want to be able to easily and quickly reconstruct the arguments in the paper and comments, in order to produce a report summarizing the arguments for the agency and the public.
3. As a poll respondent, I want to be able to participate in the poll in order to learn more about the policy issues, influence the policy-making process to protect my interests and discover which stakeholder organizations, such as political parties, share my views and represent my interests.
4. As a humanities scholar, I want to be able represent and model argumentation schemes using a high-level declarative language, in order to be able to customize or extend the schemes used by analysts to reconstruct arguments, without the help of IT experts or the need to modify the implementation of the argument reconstruction tool.

3 Argument Graphs

All tools of the Carneades system, including the polling tool which is the focus of this paper, are interoperable and tightly integrated due to their all being based on the same underlying model of argument graphs. Currently, the following tools are provided:

- A tool for creating and editing argument graphs using argumentation schemes to reconstruct arguments in source text;
- An argument visualization tool for interactively viewing maps of argument graphs containing links to source documents;
- The polling tool, for guiding users through argument graphs and collecting and aggregating opinions about the claims and arguments represented in the graphs;
- And a tool for analyzing the effects of rule-based models of policies in particular cases, via dialogues with a kind of legal expert system, which uses argument graphs to visualize and explain the results of the analysis.

Argument graphs play the role in argumentation of proof trees in classical logic. They are structures representing chains of reasoning and more general, nonlinear, relationships among inference steps. Whereas in proof trees the inference steps are applications of the strict inference rules of some calculus for classical logic, in argument graphs the steps are applications of more general argumentation schemes, which may be defeasible as well as strict.

An individual inference step in an argument graph is called an *argument node*. They are often also called "arguments", but this terminology is less precise, since the term "argument" has other uses. In particular, argument nodes are not arguments in the sense of Dung argumentation frameworks [12]. Another term for argument nodes, suggested by Trevor in a personal correspondence, would be "single-step argument". With this caveat, in contexts where there is little risk of confusion, we will use the term "argument node" and "argument" interchangeably.

A single argument graph can be used to represent all the arguments put forward in a debate, from all participants.

Argument graphs are abstract structures, which can be represented in various concrete ways in software systems. The Carneades system currently represents argument graphs in three different, but isomorphic ways:

1. In XML, using an XML schema called the Carneades Argument Format (CAF).
2. In relational databases, using a database schema defined in SQL.
3. Using data structures defined in the Clojure language, the Lisp dialect used to implement Carneades.

The rest of this section provides a formal, algebraic specification of argument graphs.

Definition (Argument Graph) An argument graph is a bipartite, directed, labelled graph, consisting of statement nodes and argument nodes connected by premise and conclusion edges. Formally, an argument graph is a structure $\langle S, A, P, C \rangle$, where:

- S is a set of *statement nodes*,
- A is a set of *argument nodes*,
- P is a set of *premises*, and
- C is a set of *conclusions*.

Argument graphs are bipartite, because they consist of two kinds of nodes, argument nodes and statement nodes, and all edges (premises and conclusions) link argument nodes to statement nodes, i.e. to nodes of different kinds.

Let L be a predicate logic *language*, containing a unary predicate symbol `applicable`. Each statement node in S is labelled with a well-formed formula of the language L.

Each argument node in A is a structure $\langle i, s, d \rangle$, where

- i is a term in L identifying the argument node (no two argument nodes in an argument graph have the same identifier),
- s is a Boolean value which is true if the argument node is *strict* and false if it is *defeasible*.
- d is a Boolean value, representing the *direction* of the argument, which is true if the argument is *pro* its conclusion and false if it is *con* its conclusion.

An atomic formula in L of the form `applicable(X)` is intended to denote that the argument node identified by the term `X` is applicable. This enables the applicability of argument nodes to be an issue in argument graphs. An argument node `a1` is undercut by an argument node `a2` if `a2` is an argument node con the conclusion `applicable(a1)`.

The premises and conclusions of an argument graph represent the edges of the graph, connecting the statement and argument nodes.

Each premise in P is a structure $\langle s, a, p \rangle$, where

1. $s \in S$,
2. $a \in A$,
3. p is a Boolean value denoting the *polarity* of the premise, i.e. positive or negative. If p is true, then the premise is positive, otherwise it is negative.

Each conclusion in C is a structure $\langle a, s \rangle$, where

1. $a \in A$, and
2. $s \in S$

Every argument node has exactly one conclusion. That is, for every argument $a \in A$ there exists exactly one $\langle a, \_ \rangle \in C$.

An argument node may have zero or more premises. That is, it need not be the case that for every $a \in A$ there exists a premise $\langle \_, a, \_ \rangle \in P$.

Figure 1 shows a visualization of an example argument graph, instantiating an argumentation scheme for value-based practical reasoning [5]. Argument nodes and statements nodes are represented by circles and boxes, respectively. Statements nodes are labeled, for readability, with a natural language representation of their formula. Argument nodes are labeled with their id and a plus or minus sign, indicating whether the argument is pro or con, respectively. The conclusion of the con argument, `a2`, is shown in the visualization with a link to the other argument node, `a2`. Thus it may appear that the argument graph is not actually bipartite. However, this is just a more readable visualization of an undercutting argument node. In the underlying argument graph, the conclusion of `a2` is actually a statement node containing the formula `applicable(a1)`, where by convention `applicable` is a standard predicate in every language L of argument graphs. All the argument nodes in the example are defeasible. The example does not illustrate negative premises.

Prior conceptualizations of argument graphs, such as Beardsley/Freeman argument diagrams [4, 13] and the Argument Interchange Format (AIF) [11], do

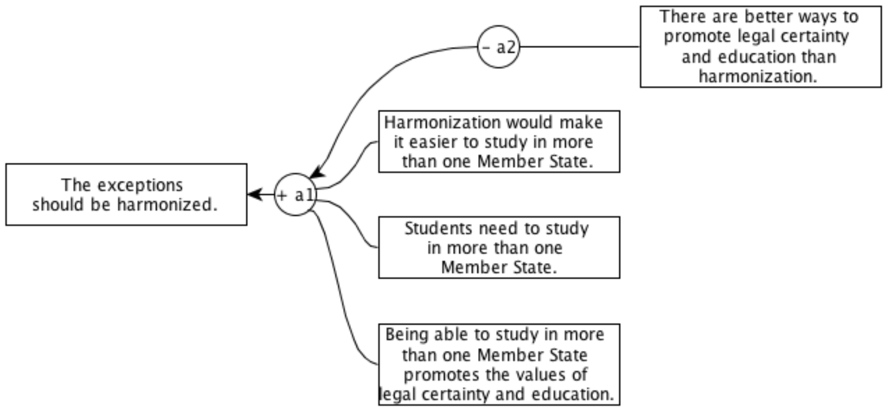

Figure 1: Visualization of a Simple Argument Graph

not distinguish pro and con argument nodes or positive and negative premises. Rather, in these prior approaches all argument nodes are pro and all premises are positive. An argument con a statement node P is represented in these prior approaches by an argument node having the conclusion $\neg P$. Similarly, if one argument node has a premise P and another has a premise $\neg P$, then two statement nodes are needed in the argument graph, using these prior approaches, one for P and one for $\neg P$. Explicit "refutation" or "conflict" links are used to express the information that P and $\neg P$ are complementary statements. Our approach has the advantage of reducing the number of statement nodes required by up to 50%. There is no room here for a deep or scholarly comparison of models of argument graphs, but a thorough comparison of Carneades argument graphs and the Argument Interchange Format has been published [10]. For the purpose of comparing the Carneades polling tool with the work of Trevor and his colleagues on the Structured Consultation Tool, it should be sufficient to note that the Structured Consultation Tool makes no use of any kind of argument graph for modeling structured arguments.

The above formalization of argument graphs defines the basic, abstract data model. In the implementation of the data model in the Carneades software, argument graphs have additional properties, omitted here, for associating weights with the argument and statement nodes, recording the results of formally evaluating the argument graph and for annotating the elements of the graph with metadata, quotations of and links to source documents, among other information.

The weights associated with the statement and argument nodes of an argument graph represent the aggregated opinion of an *audience* [22] about the truth or significance of the statement or argument, respectively. These weights can be computed from the data collected using the polling tool.

The weights can then be used to formally evaluate the argument graph using a computational model, based on an instantiation of the ASPIC+ framework [23] that maps the argument graph to an abstract argumentation framework [12]. As in earlier versions of Carneades [19, 20] proof standards are used to resolve conflicts between rebuttals, but by mapping argument graphs to Dung abstract argumentation frameworks, cyclic argument graphs can now be handled. Proof standards enable the risks of errors to be balanced against the costs of further argumentation, such as the costs of collecting evidence. This is important for practical reasoning in most domains, not just for legal reasoning.

4 Design and Implementation

The Carneades argumentation system, including its polling tool, is a three-tiered Web application, with a relational database backend, an application logic layer, and a Web client user interface.[3]

The relational database schema is a very simple and direct implementation of the algebraic model of argument graphs presented in Section 3. It consists of tables for statement nodes, arguments nodes, premises, and metadata. There are additional tables for storing poll responses and managing translations of text, to support multilingual application scenarios.

The database schema is independent of the argumentation schemes used to reconstruct arguments. A high-level declarative language for representing argumentation schemes is provided. A selection of about 20 of Doug Walton's schemes [25] have been modeled using the language, in collaboration with him. We have also used the language to represent versions of the schemes for value-based practical reasoning and arguments from a credible source developed by Trevor, Katie and Adam during the course of the IMPACT project [3]. The Carneades system is preconfigured to use these schemes, but they can be modified, extended or replaced with others. Restarting the web application is sufficient to reconfigure the system to use the new schemes. Existing argument graphs are not invalidated when the schemes are modified. Carneades includes a web application for editing argument graphs, which uses the argumentation schemes to generate forms for entering and modifying arguments. The editor can be used to update existing arguments to correctly instantiate modified schemes. A tool is planned, but not yet implemented, for checking arguments against the schemes and reporting errors.

The premises and conclusions of argumentation schemes are represented at a fine-level of granularity, at the level of a higher-order predicate logic. Premises are labelled by their roles in the scheme, e.g. "major" or "minor". All of this detail, including the identifiers of the schemes applied, is preserved in the relational database representation. Statements are represented in the database both in natural language and formally, in predicate logic. A logic-based query

[3] Carneades is open source software, freely available for downloading from `http://carneades.github.com`.

language is provided for retrieving statement nodes from the database which unify with (match) the query.

The application layer is implemented in a functional programming language, Clojure, which is compiled to byte codes for the Java Virtual Machine. A purely functional, declarative style has been used, with no mutable state and side effects only for input/output. The application layer is packaged as a Web service which can be accessed via HTTP. An inference engine is provided. It can be used to automatically generate argument graphs by applying argumentation schemes to sets of predicate logic formulas ("semantic models"). We have demonstrated this feature by reconstructing Liverpool's traffic law example [1]. Finally, the application layer also includes an argument graph evaluator, which uses a mapping to Dung abstract argument frameworks to label (in, out or undecided) the statement and argument nodes. Grounded semantics is currently used, but the system has been designed in a modular way to allow future implementations of other semantics to be selected at run-time.

The user interface, illustrated in the next section, is a Rich Internet Application (RIA), implemented in JavaScript and, more recently, ClojureScript. The client communicates with the Carneaedes Web service via HTTP and exchanges data using JSON. The style of the user interface was designed and implemented by the company User Interface Design (UID), a partner in the IMPACT project.

The entire code of the server-side of the system, including a relational database engine (H2) and a web server (Jetty), is packaged as a single, double-clickable JAR file. The system is very simple to install, requires no configuration or administration, and can be used stand-alone, on a personal computer, without an Internet connection.

5 User Interface

This section presents the user interface of the polling tool and illustrates how it is used.

The first page of the polling tool provides an overview of the features of the tool and explains the following procedure for using the tool:

1. Log in using a pseudonym to protect the respondent's privacy.
2. Read an introduction to the topic of the debate and select an issue of interest.
3. Answer a series of survey multiple-choice questions about the selected issue, asking whether the respondent agrees or disagrees with claims made in arguments.
4. View a summary of the questions and responses. The respondent is provided with an opportunity to change his or her answers.
5. Compare the respondent's opinions with those of the authors of the source documents reconstructed by analysts in the argument graph.

The procedure is flexible and the respondent is in control. The procedure can be stopped at any time, and continued later if desired. Moreover, the respondent can jump backwards or forwards to any step in the procedure.

5.1 Question Types

Three types of questions are asked during the poll. The questions are generated automatically by traversing, depth-first, the nodes of the argument graph. Figure 2 shows the form displayed the first time the respondent is asked for his opinion about some statement.

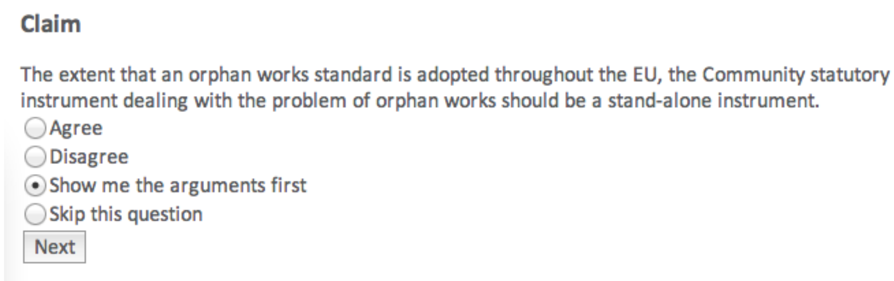

Figure 2: First Time Question About a Claim

If the respondent first wants to see the arguments before answering, and thus chooses the third alternative, then the question will be put aside and he will be shown questions about the arguments pro and con this statement. As illustrated in Figure 3, this second type of question shows the argument, quoting the formulations of the argument in the source texts, and asks for each premise whether the respondent agrees or disagrees with the premise, or, if there are arguments in the graph about the premise, whether he would like to first see the arguments (not shown in this example, since there are no arguments in the graph for these premises). Note also that the default answer, “Skip this question”, has not been changed in this example.

After the respondent has seen the arguments, to the depth and level of detail chosen by his answers to the questions, he will be asked again for his opinion of the statement. (Recall that users can control the depth of the survey by skipping questions or asking to see the arguments before answering questions.) This second time, however, the question is formulated somewhat differently. He will first be asked to weigh arguments pro and con the statement. The respondent can easily adjust the relative weights of these arguments, using sliders, as shown in the figure. The user is asked to weigh an argument only if he has agreed with all of its premises, since we expect that it would be too confusing for most users to ask them to weigh arguments as if they accepted them, when they do not. In the example there is only one argument. Weighing the argument can be useful nonetheless, since the weights entered by all respondents are averaged to resolve

Argument

> While we do not support such an approach, to the extent that an orphan works standard is adopted throughout the EU, we recommend that a Community statutory instrument dealing with the problem of orphan works should be a stand-alone instrument. As noted above, an orphan works defense would not be an exception to copyright infringement. The orphan works defense is a rights clearance mechanism that would merely serve to limit the legal remedies that a user would be subject to if that user was found liable for copyright infringement. Accordingly, a user of an orphan works owner is still deemed to be an infringer. Because the 2001 Copyright Directive relates to rights and exceptions, but not remedies, it would be inappropriate for the Directive to be amended to include a provision relating to orphan works.

Premises

The 2001 EU Copyright Directive regards copyrights and exceptions, but not remedies for violations of copyrights.

- ○ Agree
- ○ Disagree
- ◉ Skip this question

Using a stand-alone instrument to regulate orphaned works, instead of amending the 2001 Copyright Directive, would cause the separate topic of remedies for copyright violations to be regulated by a separate instrument.

- ○ Agree
- ○ Disagree
- ◉ Skip this question

Separate legal topics should be regulated by separate instruments.

- ○ Agree
- ○ Disagree
- ◉ Skip this question

Regulating separate legal topics with separate instruments promotes the value of legal clarity.

- ○ Agree
- ○ Disagree
- ◉ Skip this question

[Next]

Figure 3: Questions About the Premises of an Argument

conflicts among rebuttals when evaluating the argument graph, using weights and proof standards [19, 20].

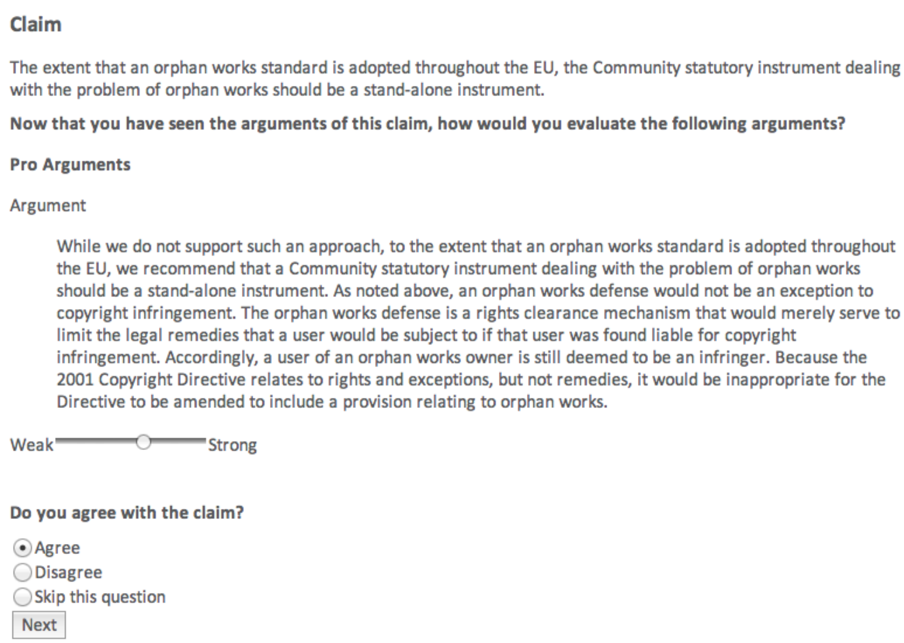

Figure 4: Second Time Question About a Claim

After the arguments have been weighed, the respondent is asked, at the bottom of the same page, whether he now agrees or disagrees with the claim.

5.2 Checking and Changing Answers

To check or change answers the respondent can go to the "summary" page, shown in Figure 5, listing all the claims with which the respondent has agreed or disagreed, showing the opinion entered and providing an opportunity to make changes. The sixth item in the list shows the user's position on the main claim, that a stand-alone statutory instrument should be used for standardizing the handling orphaned works. The user has agreed with this claim.

5.3 Comparing Opinions

Finally, the respondent can compare his opinions with those reconstructed from the source documents. The comparison page (Figure 6) shows the source documents grouped into several categories, ordered by how much the opinions expressed in the documents have in common with the opinions expressed by the respondent in his answers to the poll questions. In each category, full references

Summary

Thank you for having participated in this consultation process! Here's a list of your responses to the survey questions. Click on any item in the list to change your answer. Or compare your answers with the positions of other stakeholders by clicking the button below.

[Compare]

Your responses

- The cross-border aspects of the orphaned works issue are already provided for by the proposal of High Level Expert group, which recommends mutual recognition by Member States of each other's copyright exceptions. **Agree**. Change
- Regulating separate legal topics with separate instruments promotes the value of legal clarity. **Agree**. Change
- The policy proposed by the German Action Alliance should be adopted to address the cross-border aspects of orphaned works. **Agree**. Change
- The 2001 EU Copyright Directive regards copyrights and exceptions, but not remedies for violations of copyrights. **Agree**. Change
- Separate legal topics should be regulated by separate instruments. **Agree**. Change
- The extent that an orphan works standard is adopted throughout the EU, the Community statutory instrument dealing with the problem of orphan works should be a stand-alone instrument. **Agree**. Change
- The cross-border aspects of the orphaned works should be handled through an amendment to Directive 2011/29/EC. **Agree**. Change
- Using a stand-alone instrument to regulate orphaned works, instead of amending the 2001 Copyright Directive, would cause the separate topic of remedies for copyright violations to be regulated by a separate instrument. **Agree**. Change

Figure 5: A Summary Page

to the documents are provided (author, title, etc). The title includes a hyperlink to the source of the document on the Web.

Figure 6: Opinion Comparison Page

Here is brief explanation of how the comparison is computed. All of the arguments modeled in the argument graph are tagged with the keys of source documents in which the argument has been made, from the corpus of source documents used by the analysts to construct the graph. These documents do not merely cite or quote the argument but rather express agreement with the argument, by claiming that the premises and the conclusion of the argument are true. Since the arguments are linked to their conclusion and premises in the argument graph, it is easy to compute from the source metadata of arguments the set of claims, i.e. statements claimed to be true or false, in each source document. These claims are then compared to the respondent's opinions. The similarity of opinions is currently measured by the percentage of claims in the document with which the respondent has expressed agreement, but other metrics are possible, such as "Euclidean distance" [24, pp. 9–15]. For every claim, the opinion of the user matches the position of a comment only if they both agree, disagree or have expressed no opinion about the claim. For example, if an argument graph contains 100 claims (statements) and the opinion of the user matches 20 of the opinions of the comment, then the comment is assigned a score of 20%. The comments are grouped into five qualitative categories: very little in common (< 20%), little in common (20-39%), some in common (40-59%), much in common (60-79%), and very much in common (80-100%).

6 Discussion

The Carneades polling tool presented here, which is fully implemented, has been inspired by the Structured Consultation Tool Trevor Bench-Capon and his Liverpool colleagues Katie Atkinson and Adam Wyner developed in the

European IMPACT project [1, 26]. The SCT, in turn, builds on prior work at Liverpool by Katie, Trevor and Peter McBurney on the Parmenides system [2]. Our aim in developing and implementing the Carneades polling tool was not specifically for the purpose of this Festscrift, to allow comparison with Trevor's work, but rather with the aim to develop a practical tool meeting identified user requirements. This work was mostly completed before receiving the invitation to make a contribution to this Festschrift.

We have aimed to preserve all of the features of Parmenides and the SCT, but with a more flexible design supporting further use cases and enabling a tighter integration with tools for argument reconstruction, visualization and evaluation. These additional use cases include the provision of support for consultations about any argument, independent of the argumentation schemes used to reconstruct the arguments, the collection of feedback on the arguments put forward on all sides of a debate, rather than only the arguments of a single position of one party, and the ranking of stakeholders by the extent to which they share opinions. We believe these goals have been achieved with the polling tool presented here.

Argument graphs provided the key for this increased flexibility, by enabling the code for generating and conducting polls to be decoupled from the code for using argumentation schemes to (re)construct arguments. The SCT represents every argumentation scheme with a separate table in a relational database. Modifying the schemes, or extending the system to support further schemes, requires modifications to the database schema, the middleware (application layer) and the user interfaces, invalidating existing databases using prior versions of the schemes. All of these modifications are labor intensive and require specialist computer programming skills. Our approach, on the other hand, provides a high-level declarative programming language for specifying argumentation schemes, facilitating experimentation with various formulations of argumentation schemes, by humanities scholars as well as computer scientists, without requiring technical computer-science skills or modifications to the implementation of the polling tool. We find this feature especially useful, because we consider argumentation schemes to be an active field of research, with many schemes not yet well understood and waiting to be adequately formalized.

7 Acknowledgments

This work was partially funded by the European IMPACT project (FP7-IST-247228), 2010-2012. Carneades was conceived and designed by the author, with contributions by Douglas Walton and Henry Prakken. The current version of Carneades was programmed by Pierre Allix, Stefan Ballnat and the author.

References

[1] K. Atkinson, T. Bench-Capon, D. Cartwright, and A. Wyner. Semantic Models for Policy Deliberation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Law (ICAIL 2011)*, pages 81–90, New York, NY, USA, 2011. ACM Press.

[2] K. Atkinson, T. Bench-Capon, and P. McBurney. PARMENIDES: facilitating deliberation in democracies. *Artificial Intelligence and Law*, 14(4):261–275, 2006.

[3] K. Atkinson, A. Wyner, and T. Bench-Capon. Report on the structured consultation tool (sct). IMPACT Deliverable D5.2, University of Liverpool, 2012.

[4] M. C. Beardsley. *Practical Logic*. Prentice Hall, New York, 1950.

[5] T. Bench-Capon. Persuasion in Practical Argument Using Value-Based Argumentation Frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.

[6] T. J. Bench-Capon. Support for Policy Makers: Formulating Legislation with the Aid of Logical Models. In *Proceedings of the First International Conference on Artificial Intelligence and Law*, pages 181–189, Boston, 1987.

[7] T. J. Bench-Capon, G. O. Robinson, T. Routen, and M. Sergot. Logic Programming for Large Scale Applications in Law: A Formalisation of Supplementary Benefit Legislation. In *Proceedings of the First International Conference on Artificial Intelligence and Law*, pages 190–198, Boston, 1987.

[8] T. J. M. Bench-Capon, P. E. S. Dunne, and P. H. Leng. Interacting with Knowledge Systems Through Dialogue Games. In *Proceedings of the 11th Annual Conference on Expert Systems and their Applications (vol. 1)*, pages 123–130, Avignon, 1991.

[9] T. J. M. Bench-Capon, P. E. S. Dunne, and P. H. Leng. A Dialogue Game for Dialectical Interaction with Expert Systems. In J. C. Rault, editor, *Proceedings of AVIGNON-92 (vol. 1)*, Nanterre, 1992.

[10] F. Bex, T. Gordon, J. Lawrence, and C. Reed. Interchanging arguments between Carneades and AIF. In B. Verheij, S. Szeider, and S. Woltran, editors, *Computational Models of Argument – Proceedings of COMMA 2012*, pages 390–397, Amsterdam, 2012. IOS Press.

[11] C. Chesnevar, J. McGinnis, S. Modgil, I. Rahwan, C. Reed, G. Simari, M. South, G. Vreeswijk, and S. Willmott. Towards an argument interchange format. *Knowledge Engineering Review*, 21(4):293–316, 2006.

[12] P. M. Dung. On The Acceptability Of Arguments And Its Fundamental Role In Nonmonotonic Reasoning , Logic Programming And N-Persons Games. *Artificial Intelligence*, 77:321–357, 1995.

[13] J. B. Freeman. *Dialectics and the Macrostructure of Arguments: A Theory of Argument Structure*. Walter de Gruyter, Berlin / New York, 1991.

[14] T. F. Gordon. The Role of Exceptions in Models of the Law. In H. Fiedler and R. Traunmüller, editors, *Formalisierung im Recht und Ans{ä}tze juristischer Expertensysteme*, pages 52–59. J. Schweitzer Verlag, Munich, 1986.

[15] T. F. Gordon. Oblog-2: A Hybrid Knowledge Representation System for Defeasible Reasoning. In *Proceedings of the First International Conference on Artificial Intelligence and Law*, pages 231–239, Boston, 1987.

[16] T. F. Gordon. Some Problems with Prolog as a Knowledge Representation Language for Legal Expert Systems. In C. Arnold, editor, *Yearbook of Law, Computers and Technology*, pages 52–67. Leicester Polytechnic Press, Leicester, England, 1987.

[17] T. F. Gordon. The Pleadings Game; Formalizing Procedural Justice. In *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*, pages 10–19. ACM Press, New York, 1993.

[18] T. F. Gordon. *Foundations of Argumentation Technology – Summary of Habilitation Thesis*. PhD thesis, Technical University of Berlin, 2009.

[19] T. F. Gordon, H. Prakken, and D. Walton. The Carneades Model of Argument and Burden of Proof. *Artificial Intelligence*, 171(10-11):875–896, 2007.

[20] T. F. Gordon and D. Walton. Proof Burdens and Standards. In I. Rahwan and G. Simari, editors, *Argumentation in Artificial Intelligence*, pages 239–260. Springer-Verlag, Berlin, Germany, 2009.

[21] D. Leffingwell. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs and the Enterprise*. Addison-Wesley, 2010.

[22] C. Perelman and L. Olbrechts-Tyteca. *The New Rhetoric*. University of Notre Dame Press, Notre Dame, 1969.

[23] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument & Computation*, 1:93–124, 2010.

[24] T. Segaran. *Programming Collective Intelligence*. O'Reilly, 2007.

[25] D. Walton, C. Reed, and F. Macagno. *Argumentation Schemes*. Cambridge University Press, 2008.

[26] A. Wyner, K. Atkinson, and T. Bench-Capon. Towards a Structured Online Consultation Tool. In *Electronic Participation: Proceedings of Third IFIP WG 8.5 International Conference (ePart 2011)*, Lecture Notes in Computer Science (LNCS), pages 286–297, Berlin, 2011. Springer.