Carneades User Manual

Tom Gordon

October 21, 2012

Contents

1 Getting Started

The Carneades argumentation system is open source software, freely available for downloading at http://carneades.github.com.

Carneades provides web-based, collaborative software tools for:

- summarizing 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

In this manual, we distinguish three kinds of users:

Public Users Users interested in understanding arguments and policies, whom we assume have had *no specialist training* in argumentation or policy modeling.

Analysts Users who have had *specialist training* in argumentation or rule-based policy modeling.

System Administrators Users with sufficient computer skills to install, configure and administer the Carneades software.

These are only roles. Some people may be able to use the system in more than one of these roles. In particular, we aim to make the software simple to install and administer, so that ultimately anyone with basic computer skills will be able to serve as "system administrator".

The rest of this manual is organized in the following chapters:

- The Home Page

- Argument Graphs

- Browsing, Visualizing and Evaluating Arguments

- Forming, Polling and Comparing Opinions

- Analyzing and Comparing Policies

- Editing Argument Graphs

- Modeling Policies and Argumentation Schemes

- System Administration

- Application Scenarios

- Built-in Argumentation Schemes

The first two of these chapters, about the Carneades home page and argument graphs, are recommended to be read first by all users. The home page chapter explains how to access and navigate among the various tools provided by the system. The argument graph chapter provides a concise overview of the underlying data model used by all the tools. The next three chapters are for public users and explain how to use the tools for argument browsing, opinion polls and policy analysis, respectively. The chapters on editing argument graphs and modeling policies and argumentation schemes are for analysts with some prior knowledge of argument reconstruction methods and rule-based systems. The system administration chapter is recommended reading for anyone wishing to install, configure and administer the Carneades software. The chapter on application scenarios may be of interest to managers and public administrators looking for ways to improve the quality of decision-making processes in their organizations. The final chapter presents the initial set of argumentation schemes distributed with the system, which you can modify to suit your needs.

2 The Home Page

The home page provides an overview of the features and tools of the system along with the main menu for accessing these tools. The main menu is always available, on every page of the Carneades system, to make it easy to navigate directly from one tool to another at any time.

The items in the main menu are:

- Home

- Argument Graph

- Poll

- Policy Analysis

- System Administration

The "Home" menu item takes you back the home page, described here.

The "Argument Graph" menu item takes you a description of the debate and a hypertext outline of the argument graph model of the debate. The argument graph page includes a submenu of commands for generating argument maps, evaluating arguments, viewing argumentation schemes, exporting argument maps to XML and, if you are logged in as an analyst, for editing the argument graph.

The "Poll" menu item takes you to the opinion formation, polling and comparison tool, where you can moderate a virtual debate of the issues. In return, you will receive a guided tour of the issues and arguments, to help you to formulate your own opinion. The tool also provides you with a way anonymously share your opinion with other users. The polling results are aggregated by the system and used to help you to evaluate the acceptability of claims. Finally, the tool compares your opinions with the opinions expressed in the published documents used to create the argument graph, for example the opinions of political parties, and ranks the authors by the degree to which their opinions agree with yours.

The "Policy Analysis" menu item takes you to a tool for analyzing and comparing the effects of various policy proposals on example cases. In a dialog with the system, you will be asked a serious of questions to gather information about a real or hypothetical case. An argument graph will be generated, by applying the rules of the policy proposals to the facts of the case, showing how the various policies would work in the given case. The tool can help you to find the policies which best serve your interests. Here too your privacy is protected. Care is taken to not request or store information that could be used to reveal your identity.

Finally, the "System Adminstration" menu item takes you to a page for managing and configuring the Carneades server. You will need to log in with a user name and password. Here you can create, edit, archive and delete projects, where each project consists of an argument graph, a set of argumentation schemes, models of proposed policies, the test cases and possibly local copies of source documents.

The current project, the one being accessed and used from the user interface, is identified by a parameter in the Web address (URL) used to access the system. Only system administrators have access to the list of all projects served by a particular installation of the Carneades system.

3 Argument Graphs

Argument graphs model relationships among arguments, statements and source documents. To build the model, source documents need to be interpreted to find the arguments, in a process called "argument reconstruction".

Consider the following simple legal argument: Johnny violated the law by riding his skateboard in the park.

This same argument can be expressed in many different ways. Here are some examples:

1. Because Johnny rode his skateboard in the park he violated the law.

2. Vehicles are prohibited in the park. Someone who does something which is prohibited violates the law. Johnny rode his skateboard in the park. A skateboard is a vehicle. Therefore Johnny violated the law.

3. Johnny hat gegen das Gesetz verstoßen, weil er sein Skateboard im Park benutzt hat.

The first example just reordered the premise and the conclusion of the argument, putting the premise first. The second example reveals some implicit premises of the original formulation of the argument. The third example is a German translation of the original formulation of the argument.

All four of these texts, including the original formulation, express the *same* argument, but in different ways. In a large-scale debate, for example about European policy issues, the same argument might be expressed in *thousands* of different ways in many different languages. (The European Union has 23 official languages.)

One of the purposes of argument graphs is to provide a way to summarize complex debates with exactly one node in the graph for each argument put forward in the debate. A single argument graph is used to represent all the

arguments put forward in a debate, from all participants. The nodes can quote one or more source documents, and include links to these source documents, so no information need be lost and all voices can still be heard, inclusively. Grouping the different formulations of an argument together into a single node in the graph, abstracting away details, makes it possible to quickly obtain an overview of the arguments and to obtain a clearer picture of relationships among arguments. A hypertext or map of the source documents directly, without an argument graph, would make it difficult to "see the forest for the trees".

3.1 Data Model

The entity-relationship diagram above shows the elements of argument graphs and their connections. (The figure does not visualize a particular argument graph, but rather relationships between the elements of argument graphs in general.)

The two main elements of argument graphs are statements and arguments. Statements represent propositions, claims and assertions. Arguments represent simple inferences from one or more premises to a single conclusion. Again, there should be only one statement or argument in the graph for each statement and argument in the source documents, no matter how many different ways the statement or argument has been expressed in source documents. Some or all formulations of the statement or argument can be quoted or referenced in the metadata of the statement or argument node. See the discussion of metadata below for further information.

As can be seen in the entity-relationship diagram, arguments are linked to statements in two ways in argument graphs. Each argument has exactly one conclusion, which is a statement, and zero or more premises, where each premise has exactly one statement node. A statement may the conclusion or premise of more than one argument.

A statement may be both a conclusion and a premise, resulting in complex argument graphs, representing chains or trees of reasoning. Argument graphs may contain cycles. A simple cycle would result if a statement and a premise of the same argument. There are methods for resolving these cycles when evaluating argument graphs.

A statement in an argument graph represents both a proposition and the negation of the proposition. To continue with our example, the sentences "Johnny rode his skateboard in the park" and "Johnny did not ride his skateboard in the park" would be represented by a single statement in an argument graph. Conclusions and premises of argument can negated using con arguments and negative premises, respectively. That is, there are two kinds of arguments, pro and con. An argument is pro if its conclusion claims the statement is true and con if it claims the statement is false. Similarly, there are two kinds of premises, positive and negative. A positive premise holds if its statement is presumably

true ("in"). Conversely, a negative premise holds only if its statement is presumably false ("out").

Prior models of argument graphs do not distinguish pro and con arguments or positive and negative premises. Rather, in these prior approaches all argument nodes are pro and all premises are positive. Our approach has the advantage of reducing the number of statements up to 50%, resulting in more compact summaries of the arguments.

3.2   Statement Properties

id A Uniform Resource Name (URN) serving as a unique identifier for the statement, world-wide.

text A concise formulation of the statement, written by the analyst who reconstructed the arguments from the source documents. Paraphrases the various formulations of the statement in the sources. Translations of the text in several languages may be included in the model. Compare with the "description" property of the metadata of the statement, which can be used to quote some or all of the formulations of the statement in the sources and provide translations in several languages.

weight A real number in range of 0.0-1.0 representing the degree to which the statement is accepted as true by the users, where 0.0 means the statement is *rejected* (believed to be false by the users) and 1.0 means the statement is *accepted* (believed to be true). This information is collected via polls.

proof standard The method used to combine pro and con arguments. Several proof standards are supported by the system. For most purposes, the "preponderance of the evidence" standard should suffice. See the Editing Argument Graphs Chapter for details.

value A real number in the range 0.0-1.0, storing the output of the argument evaluation process, where 0.0 means the statement is *out* (presumably false), 1.0 means the statement is *in* (presumably true) and all other values mean the arguments are insufficient for making any presumptions about the truth or falsity of the statement.

atom An optional formal representation of the statement in predicate logic. (This feature is for analysts and need not interest public users.)

main A Boolean value (true or false) used to indicate whether the statement is one of the main issues of the debate modeled by the argument graph.

3.3   Argument Properties

id A Uniform Resource Name (URN) serving as a unique identifier for the argument, world-wide.

direction Pro or con.

strict A Boolean value (true or false) expressing whether the conclusion of the argument is necessarily true when its premises are true (strict arguments) or only presumably true. Nonstrict arguments are called "defeasible" arguments.

scheme The name of the argumentation scheme applied, if any. Optional. Examples: "argument from credible source", "argument from practical reasoning".

weight A real number in range of 0.0-1.0, representing the relative weight of the argument, compared to other arguments pro and con the conclusion of the argument. This information is collected via polls.

value A real number in in the range 0.0-1.0, used to record the output of the argument evaluation process, where 0.0 means the argument is *out* (not acceptable), 1.0 means the statement is *in* (acceptable) and all other values mean the arguments in the graph, taken together, are insufficient for determining the acceptability of this argument.

3.4 Premise Properties

polarity Positive or negative.

role The role of the premise in the argumentation scheme applied. Examples: "minor", "major".

implicit A Boolean value (true or false). Can be used to note that the premise was not explicit in the source documents from which the argument node was reconstructed.

3.5 Metadata

The argument graph as as whole, as well as each of its statements and arguments, can be annotated with metadata, using the Dublin Core. There are 15 elements in the Dublin Core. Each element may zero or more values. Here is a list of the Dublin Core elements:

1. Title

2. Creator

3. Subject

4. Description

5. Publisher

6. Contributor

7. Date

8. Type

9. Format

10. Identifier

11. Source

12. Language

13. Relation

14. Coverage

15. Rights

See Dublin Core for a detailed description and usage guidelines for each element. The Dublin Core is intended to be useful for describing a wide range of "resources" on the World-Wide Web. Not all of the elements may be applicable for argument graphs.

In addition, Carneades allows each metadata record to be assigned an optional "key", a string which can be used as a label to refer to the metadata record, such as "BenchCapon:2008", similar to the way citation keys are used in bibliographic databases such as BibTeX.[1] At most one key should be provided.

Carneades provides special support for providing description elements of the Dublin Core in multiple languages (English, German, French, . . .) and for formatting these descriptions using the Markdown language. This feature can be used to include quotations from and links to source documents in the descriptions of both statements and arguments.

4 Browsing, Visualizing and Evaluating Arguments

This chapter of the Carneades user manual explains how to:

- Access the argument graph page for viewing and editing argument graphs on the World-Wide Web.

- Use hypertext in web pages to browse an argument graph.

---

[1]

- Visualizing argument graphs in diagrams, called "argument maps", and using these maps to navigate to more detailed views of statements and arguments.

- Evaluate arguments to reveal missing premises, check the form of arguments, ask critical questions and assess the acceptability of statements.

- Export an argument graph to XML, to archive the graph or process it using other software.

- Generate outlines of the arguments in a graph, for further editing using text editors or word processors.

4.1 The Argument Graph Page

The user interface of Carneades is a web application. You access the pages and views of the user interface with web addresses, called Uniform Resource Locators (URL), just like you access any resource on the World-Wide Web. Most of the time you will access the application by clicking on a link embedded in some page on the Web, for example in a news article, blog entry or e-participation web site. If you are using Carneades as a stand-alone, desktop application, these URLs will be local addresses, from the "localhost" domain, pointing to web pages served by the application on your personal computer.

An argument graph page consists of the following parts:

- The *title* of the argument graph. This title usually includes the topic of the discussion or debate.

- A *menu bar* of commands. The commands shown depend on the role of the user. Public users, who need not login to the system, are shown the commands "export" and "map", for exporting the argument graph to XML and viewing an argument map visualization of the graph, respectively. Analysts, who must login to the system with a password, are also shown "new statement" and "new argument" commands. Only analysts may modify the argument graph.

- A *description* of the topic of the discussion modeled in the argument graph. The description may be available in several languages. The user interface provides a way to select and change your preferred language during the session (*Note: not yet implemented*). The description can be arbitrarily long and include multiple sections, paragraphs, images, hyperlinks, lists and other content.

- A list of the *main issues* of the discussion. Each item in the list is linked to a page providing detailed information about the statement in the argument graph at issue.

Reconstruction of Comments on the EU Green Paper "Copyright in the Knowledge Economy"

Export   Evaluate   Map   New Statement   New Argument

IMPACT Project. 2011. Reconstruction of Comments on the EU Green Paper "Copyright in the Knowledge Economy" .

Description

The purpose of the Green Paper on "Copyright in the Knowledge Economy" is to "foster a debate on how knowledge for research, science and education can best be disseminated in the online environment." [@GreenPaper, p. 3].

The Green Paper has two parts. The first deals with general issues and the second deals with "specific issues related to the exceptions and limitations which are most relevant for the dissemination of knowledge and whether these exceptions should evolve in the era of digital dissemination." [@GreenPaper, p. 3].

Here, we present a reconstruction of some of the policies and arguments put forward in the comments submitted in reponse to the Green Paper. Our aim is not to comprehensively model all the policies and arguments submitted, but rather to model a sufficient number of representative arguments and policies for the purpose of illustrating features of the IMPACT argument toolbox.

The Corpus Selection Working Group of the IMPACT project has chosen 4 of the 25 questions raised in the Green Paper, as well as 12 of the 323 comments submitted, representing a wide range of stakeholders, to be used for the research and development purposes of the project.

The four questions covered by this model are listed below. Click on a question for further information.

Main Issues

1   □ Q24. Should there be more precise rules regarding what acts end users can or cannot do when making use of materials protected by copyright?

2   □ Q9. Should the law be clarified with respect to whether the scanning of works held in libraries for the purpose of making their content searchable on the Internet goes beyond the scope of current exceptions to copyright?

Figure 1: An Argument Graph Page

- An *outline* of the top five levels of the arguments in the argument graph.The first level of the outline lists the main issues (again). The second level lists the arguments pro and con each issue. The third level lists the premises of each of these arguments. The fourth level lists the argument pro and con each premise. Finally, the fifth level lists the premises of these arguments. Deeper levels of the argument graph can be navigated to by first clicking on a statement or argument in the outline and then following the links on the next page. Since argument graphs may contain cycles and are not restricted to trees, some items may appear multiple times in the outline.

- A list of *references* to the source documents used to construct the argument graph. For documents available on the Web, the reference will include a hyperlink to the source document.

4.2   Using Hypertext to Browse an Argument Graph

There is a web page for each statement and argument in the argument graph providing detailed information about the element along with links to related statements and arguments in the graph. You can use these pages to navigate from node to node in the argument graph, by simply clicking on the links in the usual way. To go back to previous pages, use the back button of your web browser.

4.2.1   Statement Pages

The top of the statement page displays the properties of the statement: its id, atom, whether or not it is a main issue, its proof standard, usually "pe" (preponderance of the evidence), its weight and value. The other proof standards available are "dv" (dialectical validity), "cce" (clear and convincing evidence), and "brd" (beyond reasonable doubt). See the section on Evaluating Arguments for further details about proof standards.

The next section displays the *text* of the statement. This formulation of the statement is written by the analyst or analysts who reconstructed the arguments to build the argument graph.

If metadata had been provided for the statement, it would be displayed next. Descriptions may be entered, by analysts, in multiple languages. The description, if available, will be displayed using the language chosen by the user. If no description has been entered manually by analysts for the selected language of the user but a description is available in some other language, a translation service will be used to generate a description in the selected language (*Note: not yet implemented*).

Statement

Export | Evaluate | Map | Edit | Delete | New Argument

id	urn:uuid:6d4ab5c5-1511-42a5-a3c2-f017d2c1db2d
atom	
main issue	false
standard	pe
weight	
value	0.5

Text

Yes. The exceptions should be clarified to allow works held in libraries to be scanned for the purpose of making their content searchable on the Internet.

Pro Arguments

- Argument #1
  - Circumstances □ Not all the material digitised by publishers is scanned with OCR (Optical Character Recognition) with the purpose of making the resulting content searchable.
  - Action □ Clarifying the law to allow works held in libraries for the purpose of making the resulting content searchable on the Internet would have a transformative effect on research, learning and teaching.
  - Goal □ Realizing a transformative effect on research, learning and teaching is an important social goal.

Con Arguments

- Argument #1
  - □ If the rules regulating the scanning of works held in libraries were unclear, this would be known.
  - □ It is not known that the rules regulating the scanning of works in libraries are unclear.

Premise of

- Argument #1

Figure 2: A Statement Page

Finally, the statement pages lists pro and con arguments about the statement, i.e. arguments having this statement, or its negation, as a conclusion, as well as arguments which have this statement, or its negation, as a premise. The premises of the pro and con arguments are also listed. This makes it possible to navigate to nearby arguments and statements in the argument graph, by simply clicking on the links in these lists. Use the back button of your web browser to return to this statement page.

4.2.2 Argument Pages

Argument

Export | Evaluate | Map | Edit | Delete

id	urn:uuid:4294d969-94c3-4d09-b663-12d4b4d214e7
scheme	
strict	false
weight	0.5
value	0

Premises

- Goal ☐ Performing the action of harmonizing the exceptions and giving precedence to community law over contracts would achieve a state in which it easier for researchers and students to work in more than one Member State.
- Action ☑ Harmonizing the copyright exceptions would make it easier for researchers and students to work in more than one Member State.
- Values Promoted ☐ Achieving the goal of making it easier for researchers and students to work in more than one Member State would promote the values of efficiency, legal certainty, scientific research and education.
- Circumstances ☐ In the circumstances: Researchers and students increasingly work in more than one Member State. The patchy availability of exceptions makes their work difficult, because what is lawful in one country is probably unlawful in another. The situation is made worse by the provision of most Member States that contracts, governing the use of digital material, automatically overrides statute law.

Conclusion

pro ☐ The permitted exceptions should be harmonised so that they are available in all Member States.

Counterarguments

- Argument #1
  - Goal ☐ It is essential that the basic principle of freedom of contract be recognized and preserved by any copyright legislation.
  - Action ☐ Harmonizing copyright exceptions would impair the freedom of contract.
  - Values Demoted ☐ Impairing the freedom of contract would demote the values of innovation and the dissemination of knowledge and information.
  - Circumstances ☐ Currently, the lack of harmonization of copyright exceptions faciliates the freedom of contract.
- Argument #1
  - ☐ There are better ways to promote efficiency, legal certainty, research and education than making it easier for researchers and students to work in more than one Member State.

Figure 3: An Argument Page

16

Argument pages are quite similar to statement pages. The top of an argument page displays the properties of the argument: its id, the argumentation scheme applied (if any), whether it is a strict or defeasible argument, its weight and value. The argumentation scheme contains a hyperlink (*Note: not yet implemented*). Click on the link to view a description of the scheme.

If metadata had been provided for the argument, it would be displayed next. Descriptions can include quotations of one or more source texts expressing the argument, along with hyperlinks to the sources on the Web. The description, if available, will be displayed using the language chosen by the user. If no description has been entered manually by analysts for the selected language of the user but a description is available in some other language, a translation service will be used to generate a description in the selected language (*Note: not yet implemented*).

Next, the premises of the argument are listed. If available, the role of each premise in the argumentation scheme applied is shown (e.g. "major" or "minor"). The check boxes to the left of each premise are used to indicate whether the statement is current *in* (checked box, meaning presumably true), *out* (crossed out box, meaning presumably false) or neither (empty box, not enough information to presume either truth or falsity), given the arguments in the graph and the opinions of users from polls about the acceptability of statements and relative weights of pro and con arguments.

After the premises, the conclusion of the argument is shown, preceded by "pro" or "con", showing the direction of the argument, and a check box showing the acceptability of the conclusion, as for the premises.

Finally, a list of counterarguments is shown.[2] The premises of the counterarguments arguments are also listed. This makes it possible to navigate to nearby arguments and statements in the argument graph, by simply clicking on the links in these lists. You can use the back button of your web browser to return to this argument page.

4.3   Visualizing Argument Graphs in Argument Maps

The menus of the argument graph page, statement pages and argument maps include a "map" button. Clicking on the "map" button generates a diagram, called an "argument map", which visualizes the argument graph as a network (directed graph) of statement nodes and argument nodes connected by links. Statement nodes are shown as boxes; argument nodes with circles and boxes with rounded corners.

---

[2]By counterarguments here we mean *rebuttals* (arguments with the opposite conclusion) and *undercutters* (arguments which deny the applicability of this argument). Arguments which attack a premise of this argument ("undermining" arguments), are not listed. To navigate to undermining arguments, click on the premise of the argument of interest. The undermining arguments will be listed on its statement page.
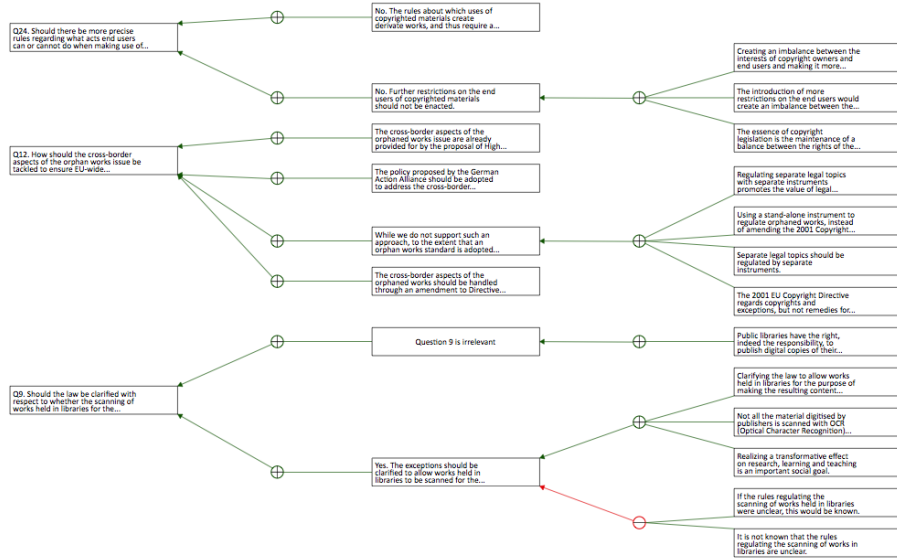
Figure 4: An Argument Map

For statement nodes, the text of the statement is shown inside the box, possibly truncated if the text is too long. In argument nodes, the circle is filled with a plus sign, if the argument is a pro argument, or a minus sign, for con arguments. The edges (links) between argument nodes and statement nodes show the premises and conclusion of the argument. The conclusion of the argument is the statement node pointed to by the edge with the normal arrowhead. The other statement nodes linked to the argument, without arrowheads, are its premises. Negative premises are displayed with a circular (dot) arrowhead on the statement side of the edge.

The statement and argument nodes in argument maps contain hyperlinks. Clicking on a statement or argument node displays the details of the node in a statement or argument page, respectively.

In argument maps, argument nodes whose conclusion is another argument node, rather than a statement node, visualize "undercutting" arguments. These are arguments which question the applicability of another argument. This is the only case where two nodes of the same time are directly connected in the map.

Argument maps are represented using structured vector graphics (SVG) not bitmaps. You can zoom the map in or out, to any scale, without loss of resolution. How this zooming is done depends on your device and web browser.

When argument graphs have been evaluated, the status of the argument and statement nodes is visualized in argument maps using both color and icons. Nodes which are "in" are filled with a green background and contain a checked

box. Nodes which are "out" are shown with a red background and contain a crossed box (a box filled with an X). Nodes which are neither in nor out are filled with white background color and contain an empty checkbox. The colors are redundant to accommodate black and white printing and color-blind users.

Argument graphs may contain cycles. However, currently the algorithm used to layout the argument and statement nodes in the map is not able to handle cycles. (*Note: This limitation will be removed in a later version of the system.*)

Argument graphs can be very large. Currently the *entire* argument graph is displayed in the argument maps. In the future only a partial view of the argument graph in maps will be shown, at least for larger graphs. The part of the graph shown will depend on the context. The map generated from the argument graph page of the argument graph will show the arguments and statements near the main issues of the map. The maps generated from statement and argument pages will show the part of the argument graph near the selected statement or argument. A method for scrolling the maps, to bring other parts of the graph into view, will be provided.

4.4 Searching for Arguments

The argument graph page will provide access to a command for searching for arguments.

(*Note: Not yet implemented*)

4.5 Evaluating Arguments

By argument "evaluation" we mean the process of critically assessing arguments by

- revealing implicit premises

- validating whether the arguments are formally correct, by instantiating accepted argumentation schemes

- asking appropriate critical questions, depending on the schemes applied

- and determining which claims are acceptable, taking into consideration the assumptions of the users and their collective assessment of the relative weights of conflicting pro and con arguments.

The first three of these tasks can be accomplished by comparing the argument with its argumentation scheme. On the argument page, click on the argumentation scheme to view a description of the scheme (*Note: not yet implemented*). Most web browsers provide some way to open the link in a new tab, so that you

can easily switch back and forth between the argument page and the description of the scheme. Now you can check whether any of the premises listed in the scheme are missing from the argument. The argument is formally valid if all the premises of the scheme are explicitly provided by matching premises of the argument and the conclusion of the argument matches the conclusion of the scheme.

Argumentation schemes define exceptions and assumptions which can be used to ask critical questions. The exceptions provide reasons for not applying the argument, undercutting it. If an exception is true, this doesn't mean that the conclusion of the argument is false, but only that the argument does not provide a good reason to presume the conclusion to be true. The assumptions of the scheme are implicit premises which need to be proven only if they are called into question. So, if you think an assumption does not hold, you should consider making an issue of it using whatever channels are available for you for participating in the discussion. (The Carneades system does not provide this service. Its function is to provide a tool for summarising and understanding arguments, wherever they take place.)

The assumptions of the users and their collective assessment of the relative weights of conflicting pro and con arguments are collected and aggregated using the opinion polling tool described in the opinion polling chapter of this manual. At any time, an analyst can execute the "evaluate" command, shown in the menu bars of the argument graph page, and the statement and argument pages, to compute the acceptability of the arguments and statements in the argument graph, on the basis of the information gathered from users via the polls. A statement is considered acceptable if it is *presumably true* given only the arguments modelled in the argument graph and their own assumptions and assessment of the relative weights of these arguments.[3]

If you do not agree with the result of the evaluation, there are at least three reasons why you may be right and the system's evaluation wrong:

1. Not all relevant arguments have been included in the model, or put forward in the debate. If you are aware of some missing argument, consider contributing it to the discussion yourself.

2. The collective, averaged opinion of the users who participated in the poll may be incorrect. Minority views can be correct. If you haven't yet participated in the poll, you may want to do so now.

---

[3]A state-of-the-art computational model of argument is used to compute the acceptability of statements. This model maps argument graphs to a so-called "Dung abstract argumentation framework" (Dung 1995), using a method based on the ASPIC+ model of structured argument (Prakken 2010), but adapted to preserve the features of the Carneades model of argument graphs (Gordon, Prakken, and Walton 2007). This ASPIC+ verison of Carneades can handle cyclic argument graphs, removing the main limitation of the original version of Carneades. Several semantics are available for Dung Abstract Argumentation Frameworks. Carneades uses grounded semantics, which is the most cautious (skeptical) about accepting claims.

3. The formal model of argument we are we are using to compute acceptability, based on the state of the art of the field of computational models of argument, may be incorrect. Of course, specialist knowledge is required to assess the correctness of the model. If you, like most people, do not have this knowledge, then we recommend a skeptical but respectful attitude. If you agree with the results of the model, then the model gives you a reason to have more confidence in your opinion. If you do not agree with the results of the model, then you may want to take pause to reconsider your views, even if in the end you do not change your mind.

4.6   Exporting Argument Graphs to XML

The menu bar of the argument graph page and the statement and argument pages includes an "export" button. Click on this button to generate an XML file containing all the data and metadata in the argument graph, including quotations of and links to source documents.

The XML files use a schema called the "Carneades Argument Format" (CAF), documented elsewhere. (*Note: write the CAF documentation*)

These XML files can be used to transfer argument graphs from one installation of the Carneades system to another, to merge argument graphs from several sources, to archive argument graphs, or to translate argument graphs into other formats, such as the Argument Interchange Format (AIF), or the generate reports or other kinds of visualisations..

4.7   Generating Outlines

(*Note: Not yet implemented*)

5   Formulating, Polling and Comparing Opinions

The opinion formation and polling tool of the Carneades system serves two purposes:

- It guides you step by step through the arguments on all sides of a complex policy debate, in a kind of simulated debate, providing you with an overview of the issues, positions and arguments in a systematic way. The tool can help you to form your own opinion, if you don't yet have one, or to critically evaluate and reconsider your current opinion, if you do. The tool also enables you to compare your responses with the published positions of some stakeholders, such political parties, helping you to find persons and organizations which best represent or share your views and interests.

21

- At the same time the tool conducts a poll to collect your views and opinions on the issues, taking care to protect your privacy. You will not be asked or required to enter any personal data, such as your name or email address, which could be used to identify you or associate your answers to poll questions. The anonymous and aggregated results of the survey can provide valuable feedback to you as well as policy makers, going beyond the information provided by traditional surveys. 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.

5.1 Accessing the Opinion Formation and Polling Tool

Every page of the Carneades system includes the main menu. Click on the "poll" button of the main menu to access the opinion formation and polling tool.

The first page of the opinion formation and 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 your privacy. You can login again using this name to change your responses at any time. Choose a cryptic name, like a secure password, to make it difficult for others to access or change your answers. (The system will generate a username for you, which you can use or change, as you prefer.)

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 you whether you agree or disagree with claims made in arguments. To help you to formulate or reconsider your opinion, you have the option to first view quotations of formulations of the arguments from the source documents. No information about the authors of the quotations will be shown during this phase, to allow the arguments to speak for themselves and avoid prejudicing your answers. (The authors will be revealed at the end, when comparing your opinions to theirs.) The system will inform you about your progress and estimate the remaining number of questions. The precise number of questions may vary, as the order and number of questions depends on your answers to prior questions. You decide how deep to delve into the issues and can control how much time to spend answering poll questions.

4. View a summary of the questions and your responses. You are provided with an opportunity to change your answers.

5. Compare your opinions with those of the authors of the source documents used to create the argument graph. Rank the authors of the source documents, such as political parties, by how much they agree or disagree with your opinions. You can click on the titles of the source documents to download and view their full text and check for yourself the extend to which your agree or disagree with the claims made.

This procedure is flexible and you are in control. You are the moderator of the virtual debate. You can stop at any time, and continue later if you'd like, or jump backwards or forwards to any step in the procedure.

5.2 Question Types

Three types of questions are asked during the poll. The questions are generated automatically by traversing, in a systematic way[4], the nodes of the argument graph. The first time you are asked for your opinion about some statement, you will be asked whether you

- agree with the statement

- disagree with the statement

- want to first see the arguments pro and con the statement before expressing an opinion, or finally

- want to skip this issue and go on the next.

Figure 5: First Time Question About a Claim

If you first want to see the arguments before answering, and thus choose the third alternative, then the question will be put aside and you will be shown questions about the arguments pro an con this statement. This second type of question shows you the argument, quoting the formulations of the argument in the source texts, and asks you whether you agree or disagree with its premsies.

---

[4]Depth-first.

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 6: Questions About the Premises of an Argument

24

After you have seen the arguments, to the depth and level of detail you choose, you will be asked again for your opinion of the statement. This second time, however, the question is formulated somewhat differently. You will first be asked to weigh the arguments pro and con the statement which you have agreed are good arguments, that is arguments whose premises, in your opinion, are true. You can easily adjust the relative weights of these arguments, using sliders, as shown in the figure.

Claim

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.

Now that you have seen the arguments of this claim, how would you evaluate the following arguments?

Pro Arguments

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.

Weak ▭────○────▭ Strong

Do you agree with the claim?

◉ Agree
◯ Disagree
◯ Skip this question
[Next]

Figure 7: Second Time Question About a Claim

After you have weighed the arguments, you will be asked, at the bottom of the same page, whether you

- agree with the statement

- disagree with the statement, or

- want to skip this issue and not express and opinion

Please be careful to take your own assessment of the relative weights of the pro and arguments into consideration when answering the question, remembering that these are all arguments you agree with, even if they happen to conflict. Weigh the arguments to resolve the conflict and justify or explain your opinion.

25

5.3 Checking and Changing Your Answers

To check or change your answers go to the "summary" page, by clicking on the "summary" menu item of the menu bar.

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 copryight 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 8: A Summary Page

The summary page lists all the statements with which you have agreed or disagreed, showing your opinion.[5] To change your opinion about some statement, click on the "change" button next to the statement. This will reveal, inline on the same page, a summary of the pro and con arguments you agree with about the statement and give you an opportunity to adjust the weights of these arguments and change your opinion of the statement. Try to take care to make sure that your new opinion remains consistent with the relative weights you assign to the pro and con arguments.

After you have changed your opinion, or cancelled the dialog, the arguments will be folded away, out of view, and you can continue checking your other answers where you left off.

---

[5]The statements you skipped, without expressing an opinion are not listed. (*Note: Not yet implemented: some way for the user to review and answer the skipped questions, including taking a tour of the arguments about the skipped questions.*).

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 copryight 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.

    Pro Arguments

    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.

    Weak ▭▭▭○▭▭▭ Strong

    ⦿ Agree
    ◯ Disagree
    Save

Figure 9: Changing Your Answer

5.4 Comparing Your Opinions with Others

To compare your opinions with those expressed in the source documents used by the analysts to build the argument graph, click on the "compare" button in the menu bar of the opinion formation and polling tool. This will take you to a page showing how much you agree or disagree with the opinions expressed in these documents. The documents are grouped into five categories:

- Very Much in Common

- Much in Common

- Some in Common

- Little in Common

- Very Little in Common

(*Note: add a screen shot*)

In each category, full references to the documents are provided (author, title, etc). The title includes a hyperlink to the source of the document on the Web. You can click on the title to download and read the original document, to judge for yourself how much you agree or disagree with the opinions expressed in the document.

The rest of this section explains briefly 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.[6] These documents do not merely cite or quote the argument. They 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 your opinions, based on your answers to the poll questions.ˆ[The similarity of opinions is calculated using the "Euclidean distance" metric (Segaran 2007, 9–15)

6 Analysing and Comparing Policies

The Carneades argumentation system provides a tool to analyze the legal effects of specific policies on the facts of cases. The tool is useful for evaluating and

---

[6]These keys are stored in the "source" property of the metadata of arguments. The keys are the identifiers of the metadata records for documents in the references section of the argument graph.

comparing policies which formulate rules at a level of detail comparable to legislation or regulations.

To access the tool, click on the "policy analysis" item in the main menu of the system, shown on every page. This will take you to introduction page of the tool, which briefly explains the purpose of the tool and the following steps for using it.

1. Select an issue of interest.

2. Answer a series of questions to enter the facts of a case.

3. View an argument graph automatically constructed by the system by applying the alternative policy models to the facts of the case.

4. Select a policy to evaluate. This will take you back to the argument graph, with the effects of the selected policy highlighted.

5. Repeat the last two steps to try another policy.

Policy Modeling Tool

Introduction | Issues | Facts | Arguments | Schemes | Policies

This is the policy modeling tool of the IMPACT argumentation toolbox. The policy modeling tool provides a way to simulate the legal effects of specific policy proposals on the facts of test cases. The tool is useful for evaluating policies which formulate rules at a level of detail comparable to legislation or regulations.

The procedure for using the tool consists of the following steps:

1. Select an issue from the Green Paper of interest, among the issues for which sufficiently concrete policies have been proposed.
2. Answer a series of questions to enter the facts of a test case. Only relevant questions will be asked. To protect your privacy, no questions about real persons or events will be asked, but rather only about hypothetical test cases. As the questions are answered, the policy modeling tool is applying the rules of the different polices to construct legal arguments about the effects of the policies, given the facts of the test case.
3. Browse a map of the arguments constructed during the previous step, to see how inferences have been drawn by applying the rules of the various policies to the facts of the test case.
4. Select a policy to simulate and highlight in the argument map its legal effects on the test case. This will take you back to the argument map, with the effects highlighted.
5. Repeat the last two steps to evaluate and visualize the effects of another policy. Or return to the "Facts" page to modify the facts of your test case or enter facts for another case.

The tool can now guide you, step by step, through this procedure, beginning when you click on the "start" button below.
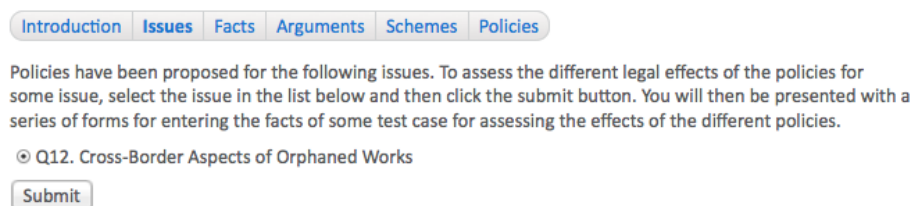
Start

Figure 10: Policy Analysis Introduction

After reading the instructions on the first page, click on the "start" button at the bottom of the page to begin. This will take you to the "issues" page.

6.1 Selecting an Issue to Analyse

Figure 11: Selecting Issues to Analyze

In the figure, only a single issue is shown but in general they may be several issues to choose form. Select an issue, by clicking on a radio button, and then click on the "submit" button to continue. This will take you to the "facts" page for entering the facts of a case.

6.2 Entering Case Facts

The "facts" page asks a series of questions to gather the facts of the case. The process of asking questions is driven, in a goal-directed way, by the process of applying the rules of the model of the various policies to try to construct pro and con arguments about the chosen issue, using an inference engine. Only relevant questions will be asked, depending on the rules and your prior answers. Related questions will be grouped together, to faciliate a more coherent dialogue.

Be careful not to enter any information which would enable you or or another person to be identified. Do not use real names, but rather pseudonyms or artificial identiers, such a P1, for persons and organizations.

After sufficient facts have been entered to apply the rules of the policies, you will be taken to the "analysis" page of the tool. (You can return to the facts page at any time to check or modify the facts of the case. (*Note: Not yet implemented.*) The analysis page displays the argument graph constructed when applying the rules of the policies to the facts entered for the case. The user interface is exactly the same as the one described in the chapter entitled "Browsing, Visualizing and Evaluating Arguments". The only difference is the content of the argument graph being displayed. Here we are viewing arguments constructed by an inference engine when applying policy models to the facts of a case, rather than manual reconstructions by analysts of arguments in source documents.

Policy Modeling Tool

Introduction | Issues | **Facts** | Arguments | Schemes | Policies

Identifiers

Please provide an identifier for the person interested in publishing the work, such as P1.

[p1] is a person.

Please provide an identifier for the orphaned work, such as W1.

[w1] is a work.

[submit]

Purpose

Will the work be used for commercial or non-commercial purposes?

Does p1 use w1 for commercial purposes? ⊙ Yes ○ No ○ Maybe

[submit]

Figure 12: Entering Case Facts

6.3   Viewing the Arguments Constructed from the Policies

Figure 13: Arguments Constructed from the Policies

The argument graph can be viewed using both a hypertext interface as well as graphically, in an argument map. Click on the "map" item in the menu bar of the analysis page to generate and view the argument map.

The next step is to select one or more policies to evaluate. Click on the "Policies" menu item in the menu bar of the policy analysis tool to view the available policies.

6.4   Viewing the Policy Models

All of the proposed policies which have been modeled are displayed. These can be alternative policies under consideration in a policy deliberation process. Alongside the model of each policy, using a rule language, there is a natural language description of the policy, possibly quoting and linking to the text of some document in which the policy was proposed. The descriptions will be shown in your preferred language, which can be selected on the home page of the Carneades system (*Note: not yet implemented*). The rules of the model are shown in an easy-to-read format, generated automatically from the source code of the model, not the source code itself.

Select Orphaned Works Policy Proposed by the Aktionsbündnisses "Urheberrecht für Bildung und Wissenschaft"

The German "Action Alliance" on copyright for education and science proposes the following policies for handling orphaned works [@Aktionsbündnis, pp. 6-7].

> Licensing copyrights. (1) The author can grant a simple or exclusive license to others to use the work ...
> Exclusive rights to use a work give the licensee the sole right to use the work in the ways stated by the
> *along with the right to grant licenses to third parties* [§ 31 UrhG, emphasis added] ...

§ UrhG-31
conclusion
   P may publish W.
conditions
   • P is a person.
   • W is a work.
   • P has a license to publish W.

§ AB-52c-1-a
conclusion
   P may publish W.
conditions
   • P is a person.
   • W is a work.
   • P uses W for non-commercial purposes.
   • P conducted a standard search for the copyright owner of W.

Figure 14: Viewing the Policy Models

6.5   Evaluating and Comparing Policies

To evaluate a policy to see how it would work given the facts of the case you have entered, click on the "select" button to the left of the title of the policy. This will take you back to the "analysis" page, again showing the argument graph, but this time visualizing what the legal consequences of the selected policy would be in the given case, if the policy were enacted and put in force. You can also view the results of the chosen policy in an argument map, by clicking on the "map" button in the menu bar.

To analyze the effects of other policies, go back to the policies page, by clicking on the "policies" button in the menu bar, and select another policy. Repeat this process to compare several policies.

6.6   Finding Policies with Desired Effects

(*Note: Not yet implemented.*)

Rather than manually trying out a number of policies, one at a time, it is possible to automatically search for the policies with the desired result in your case, as follows:
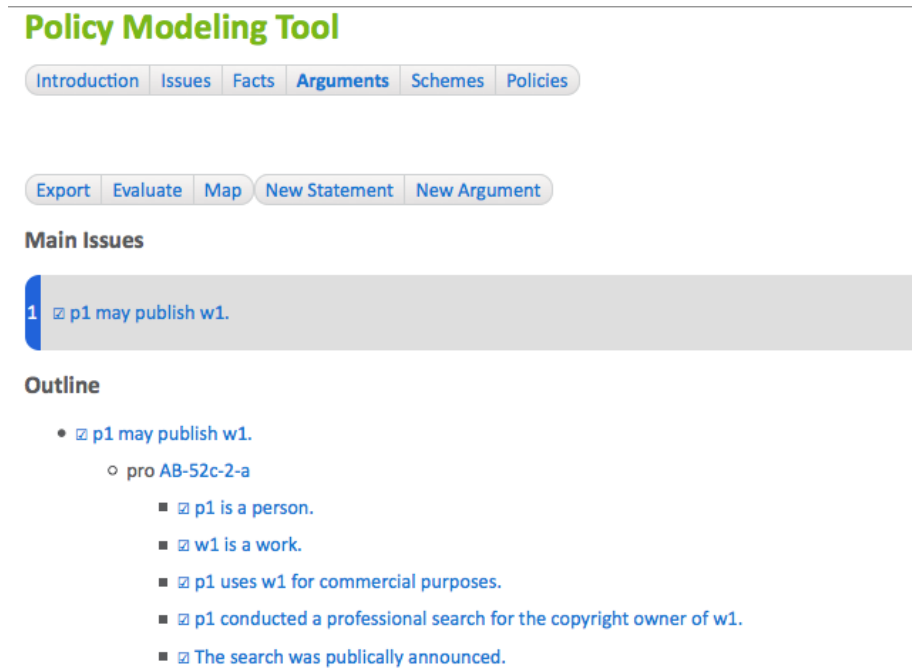
Figure 15: Viewing the Effects of a Selected Policy

1. In the menu bar of the "policies" page, click on the "find" button.

2. You will be asked whether the main issue of the case should be "in" (true), "out" (false) or "undecided". Select your preferred alternative.

3. The system will then present a list of the policies, if any, which produce the preferred result in this case. This list has the same form and user interface as the policies page, but showing only the subset of the policies which achieve the desired result. You can now select from these policies, as before, to view the argument graph evaluated using the selected policy, to understand just why the policy leads to the desired result in this case.

6.7 Sharing Cases

You can share the cases you have entered with others, to allow anyone to see how the various policies work in this case and make it unnecessary for others to have to reenter the facts of the case.

Every case is stored in its own database on the server and assigned web address (URL) by the Carneades system. This URL can be used, by anyone, to view

and browse the argument graph constructed for the case and to compare the effects of the different policies on the case.

The URL to use for sharing cases is the URL of the argument graph view of the case. To copy the URL:

1. Click on the "analysis" button in the menu bar of the policy analysis tool.

2. Copy the URL shown in the address bar of your Web browser, where you type in URLs to visit web sites.

To share the case, paste this URL into an email, a blog or discussion forum article, or indeed anywhere else text can be entered.

(*Note: Not yet implemented:*) The case will be read-only, not modifiable, by other users. (The case is editable only during the session in which it was created.) However, an editable copy of the case, with its own URL, can be created, as described in the next session.

6.8   Making Editable Copies of Shared Cases

(*Note: not yet implemented*)

To make it easier to create variations of test cases, without having to reenter the facts they have in common, it is possible to creating a copy and then edit the facts of this copy. The copy will have its own URL, which can be shared with others on the Web, just like the original. The facts of the copy can be modified during the session in which the copy was created.

To copy a shared case:

1. Enter the URL of the case in the address bar of your web browser, or just click on some link with this URL. This will take you to the analysis page for viewing the argument graph of the case.

2. Click on the "copy" button in the menu bar of the analysis page. This will create the copy and then take you to the analysis page for viewing the argument graph of this copy. When you first visit this page, it will display a clear warning, so that you can be sure that the copy is being displayed, and not the original case.

You can now edit the facts of this copy during this session, and also share this copy with others in the usual way, as described in the Sharing Cases section of this manual.

6.9 Policy Opinion Polls

(*Note: Not yet implemented.*)

Opinions can differ of course about what the correct or best result is for a particular case. The policy analysis tool provides a way to poll users to gather their opinions about the claims of the case (its main issues in the argument graph). Any user can take part in the poll, not just the user who entered the facts of the case.

To take part in the poll, click on the "vote" button in the menu bar of the policy analysis tool. The main issues of the case will be displayed, for example "The work may be copied.", along with the following question for each claim:

> Given the facts of this case, should claim "The work may be copied."
> be: 1) accepted, 2) rejected, or 3) undecided?

It is not necessary to login to the system or provide a username to take part in the poll.

After answering the questions, the current tally of the poll results will be displayed in a table, like this:

Claim	Accepted	Rejected	Undecided
The work may be copied.	55%	25%	20%

Table 1: Results of the Case Poll

Notice that users are not asked to directly express their preferences regarding the polices under consideration, just what they think the correct result should be in particular cases. See the section of this manual entitled Generating Policy Reports for further information.

6.10 Generating Policy Reports

(*Note: Not yet implemented.*)

The policy analysis tool can generate a report ranking the policy proposals by the degree to which they achieve the results preferred by the majority of the users who took part in the poll.

To generate the report, click on the "report" button in the menu bar of the policy analysis tool. This will cause the system to apply every modeled policy to all the cases and then compare the results with those preferred by the majority of the users who participated in the poll.

The report will be presented in the form of a table, similar to the following:

Policy	Agreement
Org 1 Proposal	55%
Org 2 Proposal	35%
Org 3 Proposal	15%

The policy names, such as "Org 1 Proposal" in the example, will be linked to the descriptions and models of the policies. The policies are ranked by the percentage of issues in all the cases (there may be more than one issue per case) which have the result preferred by the majority of poll participants. In the example report, the "Org 1 Proposal" obtains the results preferred by the majority for 55% of the issues.

7 Editing Argument Graphs

Normally, argument graphs can be edited only after logging into the system with a password. (*Note: the login form has not yet been implemented. Currently all users are logged in automatically, sidestepping this protection.*) Interpreting source texts to reconstruct arguments using argumentation schemes requires some training. The editing functions of the system are intended for use by "analysts" with the required skills, not "public users". See the Getting Started chapter for a discussion of the different user roles supported by the system. System administrators can make the editing functions available to the public, if this is desired for some application. See the System Administration chapter for further information.

To edit an argument graph, first go to the argument graph page, as described in the Chapter entitled Browsing, Visualizing and Evaluating Arguments.

You can add nodes to the graph by clicking on the "New Statement" or "New Argument" buttons in the menu bar. In both cases you will be presented with a form to enter the required information. The form will be inserted and displayed at the top of the current argument graph page, so that you can scroll down to view information about the argument graph, without having to toggle back and forth between two tabs or pages in your web browser. See the Entering New Statements or Entering New Arguments sections for further information.

To add a new argument pro or con some existing statement, go to the statement page and click on the "New Argument" button in its menu bar. The conclusion of the new argument will be set to the existing statement. Then complete the rest of the form as described in the Entering New Arguments section.

Reconstruction of Comments on the EU Green Paper "Copyright in the Knowledge Economy"

Export   Evaluate   Map   New Statement   New Argument

IMPACT Project. 2011. Reconstruction of Comments on the EU Green Paper "Copyright in the Knowledge Economy".

Description

The purpose of the Green Paper on "Copyright in the Knowledge Economy" is to "foster a debate on how knowledge for research, science and education can best be disseminated in the online environment." [@GreenPaper, p. 3].

The Green Paper has two parts. The first deals with general issues and the second deals with "specific issues related to the exceptions and limitations which are most relevant for the dissemination of knowledge and whether these exceptions should evolve in the era of digital dissemination." [@GreenPaper, p. 3].

Here, we present a reconstruction of some of the policies and arguments put forward in the comments submitted in reponse to the Green Paper. Our aim is not to comprehensively model all the policies and arguments submitted, but rather to model a sufficient number of representative arguments and policies for the purpose of illustrating features of the IMPACT argument toolbox.

The Corpus Selection Working Group of the IMPACT project has chosen 4 of the 25 questions raised in the Green Paper, as well as 12 of the 323 comments submitted, representing a wide range of stakeholders, to be used for the research and development purposes of the project.

The four questions covered by this model are listed below. Click on a question for further information.

Main Issues

> **1** □ Q24. Should there be more precise rules regarding what acts end users can or cannot do when making use of materials protected by copyright?

> **2** □ Q9. Should the law be clarified with respect to whether the scanning of works held in libraries for the purpose of making their content searchable on the Internet goes beyond the scope of current exceptions to copyright?

Figure 16: An Argument Graph Page

To edit or delete existing statements or arguments, first go to the page of the statement or argument and then click on the "Edit" or "Delete" button of its menu bar. Editing statements and arguments is done using the same forms used to create new statements and arguments. Deleting a statement will also delete the arguments pro or con the statement. Deleting an argument does not delete the conclusion or premises of the argument. This can leave some statements in the argument graph being unused in any argument.

Warning: There is no undo function, so all editing and delete operations are permanent. However, you will be asked to confirm all delete operations and have the option of cancelling or saving editing operations.

7.1   Reconstructing Arguments

To reconstruct a *new* argument in some source text, one which is not already in the argument graph, follow the procedure below. If instead the source text is *another formulation* of an argument already in the argument graph, you can modify the description of the existing argument to also quote this source.

1. Optional: If the source text is not already in the list of references of the argument graph, you can add it by following the instructions in the Editing the Metadata and Reference List of an Argument Graph section of this manual. When doing so you will assign the source document a "key" (label). Remember or copy this key.

2. Click on the "New Argument" button in menu bar of the argument graph page.

3. Copy the text of the argument from the source document and paste it into the description field of the new argument form. You can quote the text of the argument, using Markdown, by preceding each line with a ">" symbol. Be careful not the reveal the authors of the quoted source texts if you are planning to use the opinion formation and polling tool, to avoid prejudicing the opinions of the users taking part in the poll. At the end of the quotation, cite the source document, using the key you assigned it when adding it to the list of references, using the Pandoc extension of Markdown for citations. For example, if the key is "BenchCapon:2011" the cite would have the form [@BenchCapon:2011, pg. 5]. The page number or numbers are optional. See the Pandoc documentation for details.

4. Optional: You can choose an argumentation scheme to apply, from the pull down list of schemes in the argument editor. If you choose a scheme, its documentation will be shown and the form will be customized, with fields intialized for each of the premises, exceptions and assumptions of the scheme. You can modify the argument however you want, unconstrained by the chosen scheme, for example by deleting or adding premises, or renaming premise roles. The schemes are there to help you, not constrain

you. You can however check whether the argument complies with the given scheme. See the Validating Arguments section for details. See the Argumentation Schemes chapter for documentation of the schemes included in the Carneades. These schemes may be modified or extended, or replaced entirely, as described in the Modeling Policies and Argumentation Schemes chapter.

5. Enter the conclusion of the argument, either by choosing a statement already in the graph with the same meaning as the conclusion of the argument in the source text or by creating a new statement, by clicking on the plus (+) button to the right of the "select a statement" message. (Move the move over the "select a statement" message to see the plus sign, which is otherwise hidden.) If you create a new statement, you can either quote the conclusion of the argument in the "text" field of this new statement, or reformulate the conclusion in your own words.

6. Similarly, add the premises of the argument, either by choosing existing statements in the argument graph or by adding new statements.

7. At the bottom of the form, click the "Save Argument" or "Cancel" button. **Warning:** Any changes you make using the form will not be saved to the argument graph until you execute the "Save Argument" command by clicking on this button.

7.2 Editing the Metadata and Reference List of an Argument Graph

(*Note: Not yet implemented.*)

7.3 Entering New Statements

The form for entering new statements is shown when you click on the "New Statement" button in the menu bar of the argument graph page or on a plus sign (+) next to the conclusion or premise pull-down selection boxes for statements. (The plus sign allows you to enter a new statement if the statement you need is not listed in the pull-down menu.)

A new statement should not be entered into the argument graph if the negation of the statement is already in the argument graph. That is, for every proposition P, a single statement node should be entered into the graph to represent *both* P and ¬P. It doesn't matter whether the positive or negative form of the statement is included explictly in the argument graph. Typically the positive form is used, but in some cases you may prefer the negative form. For example, you may prefer to include the sentence "The payment was illegal" explicitly in the argument graph and then represent arguments pro the legality of the payment

New Statement

Header

Description:

| En | De | Fr | It | Nl | Sp |

| H1 H2 H3 H4 H5 H6 | **B** *I* | ≔ ≔ | ▣ ⊕ | 🖉 🗗 | ✔ |

MARKDOWN

Add a [Key ⇅] to the header [Add]

Atom (optional): [_____]

Text:

| H1 H2 H3 H4 H5 H6 | **B** *I* | ≔ ≔ | ▣ ⊕ | 🖉 🗗 | ✔ |

MARKDOWN

Proof standard: [Preponderance of Evidence ⇅]

Main: ○ Yes ◉ No

Weight (optional): Low ════○════ High [____]

Cancel   Save statement   41

Figure 17: New Statement Form

as argument con the claim the payment was illegal. The best choice may depend on the procedural context of the debate.

Only one form of the statement is required, since arguments pro the negation of the statement are equivalent to arguments con the statement. And premises of arguments can be explicitly negated in argument graphs, and in the forms for entering and editing arguments. This approach has the advantage of reducing the number of statements in the graph up to 50%.

7.3.1 Statement Description

You can use the *description* field to provide whatever background information you want about the statement. Unlike the mandatory "text" field (see below), this information is optional. You can structure and format the description, including headers, lists, quotations, hypertext links and other elements, using the Markdown wiki language. The form includes a Markdown editor to make this easier for you. You can enter translations of the description. There is a tab of the description form for each available language. (The set of languages can be configured by system administrators when installing the system. *Note: not yet implemented.*) The system is preconfigured for the following languages:

En English

De German

Fr French

It Italian

Nl Dutch

Sp Spanish

7.3.2 Statement Metadata

Statements can be annotated with metadata properties, using the Dublin Core metadata elements. The Dublin Core elements are briefly summarized in the Metadata section of this manual.

To add a metadata element to a statement, select the element from the pull-down menu and click on the "Add" button. A text box for entering the value of the element will be added to the form. Multiple values of an element can be added by clicking on the "Add" button as often as needed. A value can be deleted by clicking on the "x" icon to the right of the value.

7.3.3 Atoms: Formalizing Statements using Predicate Logic

Optionally, statements can be formalized in predicate logic, by providing a value for the "atom" property in the form. This is an advanced feature that may be needed only for more specialized application scenarios.

Atoms are formalized in Carneades using the prefix notation of s-expressions ("symbolic expressions"), as in the Lisp family of programming languages.

For example, Socrates is a man could be formalized as `(man Socrates)`. Here, `man` is a unary predicate symbol. The fact that Socrates died in 399 B.C. could be represented using a binary predicate, for example as `(died Socrates -399)`.

The language for Atoms is quite expressive:

- The arity of atoms can be 0 or more, as in predicate logic, not restricted to unary and binary predicates, as in description logic, RDF and OWL. Examples: `(good)`, `(between 0 1 2)`.

- Compound terms may be used. For example `(= 2 (+ 1 1))`. The is, the atom language is not restricted to Datalog.

- Atoms can be higher-order. Example: `(believes Gerald (and (made-of Moon green-cheese) (really-exists Yeti)))`. Notice that compound propositions can be reified as terms, as in this example.

If you choose an argumentation schemes to use as a template when editing arguments, the statements of the argument will be initialized with atom schemas. These are atoms with schema variables, represented by strings starting with a question mark, such as `?x`. If your application does not make use of atoms, you can simply ignore the atoms, or delete them, as you prefer. If your application does use atoms, you need to substitute the variables in the schemas with constants from your application domain, where constants are represented by symbols not beginning with a question mark, such as `work1`, or numbers.

7.3.4 Statement Text

The "text" property of a statement node of an argument graph is for expressing the statement concisely in natural language. You should always provide such a text. As for descriptions, translations of the text in several languages may be included in the model and the statment may be structured using Markdown. (*Note: the tabs for entering the text is more than one language is not yet implemented. Update the screenshots when this has been fixed.*) This is the text that will appear in statement boxes in argument maps and in hypertext views of the argument graph. Whereas descriptions are optional and part of the *metadata* about the statement, this text is the *data* representing the statement itself.

7.3.5 Proof Standard

The proof standard of a statement determines how much proof is required for the statement to be deemed acceptable (presumably true). The proof standard is used by the computational model to argument to compute the acceptability of the statement. Several proof standards are available:

Preponderance of Evidence (pe) This standard is meet if at least one pro argument is *in* that weighs more than any *in* con argument. (Only arguments pro or con this statement are compared.)

Clear and Convincing Evidence (ce) This standard is satisfied if the preponderance of evidence standard is meet and, in additional, the difference between the strongest *in* pro argument and the strongest *in* con argument is above a certain threshold.

Beyond Reasonable Doubt (brd) This standard is meet if the clear and convincing evidence standard is meet and, in addition, the weight of the weakest *in* con argument is below a certain threshold.

Dialectical Validity (dv) This standard is the only one that does not make use of arugment weights. It is satisfied if at least one pro argument is *in* and no con argument is *in.*

The default proof standard is preponderance of the evidence, and for most applications this proof standard should be sufficient. Note that the preponderance of evidence standard is meet whenever the dialectical validity standard is met. If arguments are not weighed, the dialectical validity and preponderance of evidence standards will give the same results. The preponderance of evidence, clear and convincing evidence and beyond reasonable doubt standards are ordered by the amount of proof required, with beyond reasonable doubt requiring the most proof. Whenever one of these standards is meet, any weaker standards are also meet.

7.3.6 Main Issue

An argument graph should have at least one main issue. These are the issues which are central to the debate. All the other issues are subsidiary issues only important to the extent that they are relevant for resolving one of the main issues.

The main issues are offered as entry points on the argument graph page and by the "guided tours" of the opinion formation and polling tool.

The make the statement a main issue, click on the "Yes" radio button next to the "Main" label in the statement editor.

7.3.7 Statement Weight

The weight of a statement encodes the extent to which users agree or disagree with the statement, on a scale of 0.0 to 1.0, where 0.0 represents disagree (reject), 0.5 means no opinion and to 1.0 is denotes agree (accept).

The weight can be entered directly in the statement editor, using a slider. This can be useful for exploring interactions among the arguments given various assumptions. Alternatively the public users can be polled for their opinions and the results of the poll can be averaged to compute the weights. See the Using Polls to Weight Statements and Arguments section of this chapter for further information.

7.4 Editing Statements

To edit an existing statement in an argument graph, first navigate to the statement, using either the hypertext view of the graph, beginning on the argument graph page, or the argument map. In the map view, click on the box containing the statement you want to edit.

Either way you will now be viewing the statement page. Click of the "Edit" button in the menu bar of the page. This will reveal, on the same page, a form for modifying the statement. This is the same form used to enter new statements, described in the Entering New Statements section of this chapter, but with the fields of the form filled in with the current values of the properties of the statement. After making changes, click on the "Save Statement" button at the bottom of the form, to have the changes stored in the argument graph database, or the "Cancel" button to abort the editing process and retain the prior values of the properties of the statement.

Warning: The changes cannot be undone after saving them.

7.5 Deleting Statements

To delete a statement from an argument graph, first navigate to the statement, using either the hypertext view of the graph, beginning on the argument graph page, or the argument map. In the map view, click on the box containing the statement you want to edit.

Either way you will now be viewing the statement page. Click of the "Delete" button in the menu bar of the page. You will be asked to confirm or cancel the deletion.

Warning: The deletion cannot be undone after it has been confirmed.

Warning: Deleting a statement also deletes all arguments pro or con this statement, i.e. with this statement as the conclusion of the argument. The statements for the premises of these arguments are not deleted.

7.6 Entering New Arguments

The form for entering new arguments is shown when you click on the "New Argument" button in the menu bar of the argument graph page or a statement page. When used on a statement page, the form for entering the new argument will be initialized with the statement as the conclusion of the argument.

See the section on Reconstructing Arguments for tips about how to proceed when interpreting and modeling arguments in source texts.

7.6.1 Argument Description

You can use the *description* field of the argument to provide whatever background information you want about the argument, including quotations of formulations of the argument in source texts, along with hyperlinks to the sources.

You have the same possibilities for structuring the text of the description, using the Markdown wiki language, and entering translations of the description as you have for statement descriptions. See the Statement Description section of this chapter for further information.

If you plan to use the argument graph with the opinion formation and polling tool, be careful to formulate the descriptions in a way which does not risk prejudicing the answers by revealing the authors of quotations. Reference the source of the quotation using using a citation key. See the section on Reconstructing Arguments for further information.

7.6.2 Argument Metadata

Arguments can be annotated with metadata properties, using the Dublin Core metadata elements, in the same way as statements. See the Statement Metadata section for further instructions.

7.6.3 Argument Direction

Arguments can be *pro* or *con* their conclusion. An argument con a statement is semantically equivalent to an argument pro the negation of the statement. Select the direction of the argument in the form using the radio buttons.

7.6.4 Strict or Defeasible Arguments

An argument may be *strict* or *defeasible* (not strict). An argument is strict if and only if its conclusion must be true, with no exceptions, if its premises are true, whether or not its premises are in fact true. For example the following arguments are strict:

Figure 18: New Argument Form

- The moon is made of green cheese since the moon is made of green cheese and unicorns have horns.

- The figure is a triangle since it has three sides.

The first example argument is of course cyclic, not to mention fanciful. But it is strict nonetheless.

Arguments are defeasible if they are not strict. With defeasible arguments, the conclusion is only *presumably* true when the premises are true. The argument gives us a reason to accept the conclusion, but there may be exceptions or other reasons leading to the opposite conclusion. Here are a couple of famous examples from the computational models of argument literature:

- The object is red since it looks red.

- Tweety flies sinces Tweety is a bird.

The object may look red because, for example, if is being illuminated by a red light. And Tweety the bird may be a penguin, or have a broken wing, and so on.

The conclusion of a defeasible argument need not be *probably* true, in some statistical sense. It is not necessary to have good empirical data allowing us to draw conclusions about what is probably the case in order to make a defeasible argument. The argument only needs to give us some good reason to believe the conclusion, at least until we have heard arguments giving us reasons to the contrary.

Select whether the argument is strict in the form using the radio buttons. By default, arguments are defeasible.

7.6.5 Argument Weight

Argument weights only become important when two or more arguments with contradictory conclusions, that is arguments pro and con some conclusion, come into play. Such arguments are called "rebuttals". The conflict between the rebuttals is resolved by weighing the arguments and applying proof standards. See the Proof Standard of this chapter for further information.

The weight can be entered directly in the argument editor, using a slider. This can be useful for exploring interactions among the arguments and experimenting with the effects of different proof standards. Alternatively the public users can be polled for their opinions and the results of the poll can be averaged to compute the weights. See the Using Polls to Weight Statements and Arguments section of this chapter for further information.

7.6.6   Using Argumentation Schemes

Arguments can be entered into the system, without using argumentation schemes. Their use is entirely optional. That said, argumentation schemes can be helpful for interpreting source texts when trying to reconstruct arguments. They serve as templates which can guide you in your task of understanding the source text. For example, if the argument looks like it might be an argument from expert opinion, using the expert opinion scheme will help you to remember the conventional form of this kind of argument. Can the text reasonably be understood as including all the premises of the expert witness scheme? Were any missing premises left implicit by the author, or would this be reading too much between the lines? Does the argument perhaps better fit some other scheme?

See the section on Reconstructing Arguments for tips about how to proceed when interpreting and modeling arguments in source texts.

You can view a list of the available argumentation schemes using the pull-down menu. Type in a string, such as "expert", to display the schemes containing the string in the title. Then select the desired scheme from the list. This will display a description of the scheme, perhaps with example arguments, along with references to publications about the scheme. You can go back and select another scheme at any time.

Whenever a scheme is selected, the form will customized to include premises and exceptions fields for the chosen scheme. The roles of the premises and exceptions will be modified to match the selected scheme. The premise roles can be edited. You are not constrained by the scheme.

If you change your mind and select another scheme, any statements you have selected for the conclusion, premises and exceptions of the argument will remain selected, but you will need to check whether they are still appropriate and select or add other statements if necessary.

Again, the argumentation schemes are meant to be helpful, not get in your way. They do not constrain you to enter arguments matching the schemes. You can, however, check whether arguments correctly instantiate the schemes they have been assigned. See the Validating Arguments section for further information.

7.6.7   Entering Premises and Exceptions

In the premise and exception fields of the form, you can choose an existing statement in the argument graph or click on the + icon shown on the right-hand side of the field, when you move the mouse cursor over the field, to open up a form for entering a new statement. When you add a new statement, its "atom" property will be initialized with a predicate logic formula. If you are interested in formalizing the statements, you can use this formula as a template, replacing its schema variables with constants, or edit the formula to meet your

own requirements. If you don't need to formalize statements, you can either delete the atoms or just ignore them.

You can additional premises or exception by clicking on the "Add a premise" and "Add an exception" buttons. Premises and exceptions can be deleted by clicking on the X icon shown on the right-hand side of the premise or exception field, when you move the mouse cursor over the field.

7.7 Editing Arguments

START HERE

7.8 Validating Arguments

7.9 Deleting Arguments

7.10 Using Polls to Weigh Statements and Arguments

Public users can be polled for their opinions and the results of the poll can be averaged to compute the weights of statements and arguments. See the Formulating, Polling and Comparing Opinions chapter for further information about polls.

To compute the weights of statements and arguments from the poll results, go to the argument graph page and click on the "Weigh" button in the menu bar. (*Note: not yet implemented.*) Afterwards you may also want to re-evaluate the argument graph to update the acceptability labels of the statement and argument nodes, as explained in the Evaluating Argument Graphs section.

7.11 Evaluating Argument Graphs

Computing Acceptability and Labeling

7.12 Importing XML

(*Note: XML import, unlike XML export, is not yet implemented.*)

The Carneades Argument Format (CAF) is an XML schema for storing argument graphs in files, in a structured but plain text form which can be read or edited by humans, using any text editor, that is also suitable for machine processing.

To facilitate moving statements and arguments between argument graphs, across installations of the Carneades system, every statement and argument is assigned

an [Uniform Resource Name](#) (URN). URNs are identifiers which are extremely likely to be unique, world-wide.

(*Note: Write a reference manual describing the CAF format.*)

For information about how to *export* argument graphs to XML, see the Exporting Argument Graphs to XML section of this user manual.

To import a CAF file, navigate to the argument graph page and click on the "Import" button in the menu bar. You be asked to select a file from your file system. The selected filed will be validated to make sure that it is a valid CAF file. It if it is valid, it will be used to update the argument graph, as follows.

1. First, you will be asked if you want to *merge* or *overwrite* the data in the argument graph in the database with the data in the CAF file.

2. If you choose to *merge* the data:

   - The properties of statements or arguments in the argument graph will be *replaced* by the values of the same properties in the CAF file but the values of properties undefined in the CAF file will retain their current values in the database. (**Note**: all values of a multivalued metadata property will be *replaced* by the values of the property in the CAF file. The values of individual properties are not merged.)

   - Any new statements or arguments in the CAF file will be added to the database, without deleting statements or arguments in the database which are not in the CAF file.

   - The metadata of the argument graph as a whole will be updated using the metadata in the CAF file. Properties of the argument graph will be replaced by the values of the same properties in the CAF file. (**Note**: If a property has multipled values, they will all be replaced, not merged.) Properties not in the CAF file will retain their current values in the database.

   - References of the argument graph in the database will be replaced by references with the same keys in the CAF file, but references in the database which are not in the CAF file will be retained, not deleted.

3. If instead you choose to *overwrite* the data:

   - The properties of statements or arguments in the argument graph will be *replaced* by the values of the same properties in the CAF file and properties undefined in the CAF file will be deleted in the database. (**Note**: all values of a multivalued metadata property will be *replaced* by the values of the property in the CAF file. The values of individual properties are not merged.)

- Any new statements or arguments in the CAF file will be added to the database, without deleting statements or arguments in the database which are not in the CAF file.

- The metadata of the argument graph as a whole will be updated using the metadata in the CAF file. Properties of the argument graph will be replaced by the values of the same properties in the CAF file. (**Note**: If a property has multipled values, they will all be replaced, not merged.) Properties not in the CAF file will be deleted in the database.

- References of the argument graph in the database will be replaced by references with the same keys in the CAF file and references in the database which are not in the CAF file will be deleted.

(*Note: Look for a simpler solution.*)

8 Modeling Policies and Argumentation Schemes

9 System Administration

10 Applications Scenarios

11 Argumentation Schemes

This chapter presents the argumentation schemes included in the distribution of the Carneades system. The system is pre-configured to use these schemes, but system administrators can configure the system to use other schemes.

11.1 Argument from Position to Know

(*Note: Include both Doug's version and Liverpool's "argument from credible source".*)

11.2   Argument from Witness Testimony

11.3   Argument from Expert Opinion

11.4   Argument from Analogy

11.5   Argument from Precedent

11.6   Argument from Verbal Classification

11.7   Argument from Definition to Verbal Classification

11.8   Defeasible Modus Ponens

11.9   Argument from an Established Rule

11.10   Argument from Value Promotion

11.11   Argument from Value Demotion

11.12   Argument from Positive Consequences

11.13   Argument from Negative Consequences

11.14   Argument from Practical Reasoning

(*Note: Include both Doug's and Liverpool's versions.*)

11.15 Argument from Cause to Effect.

11.16 Argument from Correlation to Cause

11.17 Argument from Sunk Costs

11.18 Argument from Appearance

11.19 Argument from Ignorance

11.20 Argument from Abduction

11.21 Ethotic Argument

11.22 Slippery Slope Argument

Transitivity of Causality, Inductive Step

12 Credits

- Conception and Design

  - Tom Gordon
  - Douglas Walton

- Programming

  - Pierre Allix
  - Stefan Ballnat
  - Tom Gordon
  - Matthias Grabmair

- Funding (EU Projects)

  - Standardized Transparent Representations in order to Extend Legal Accessibility (Estrella, FP6-IST-027655), 2006-2008.
  - Quality Platform for Open Source Software (Qualipso, FP6-IST-034763), 2006-2010.
  - Integrated Method for Policy making using Argument modelling and Computer assisted Text analysis (IMPACT, FP7-IST-247228), 2010-2012.
  - The MARKet for Open Source – An Intelligent Virtual Open Source Marketplace (MARKOS, FP7-ICT-317743), 2012-2015.

13 References

Dung, Phan Minh. 1995. "On The Acceptability Of Arguments And Its Fundamental Role In Nonmonotonic Reasoning , Logic Programming And N-Persons Games." *Artificial Intelligence* 77: 321–357.

Gordon, Thomas F., Henry Prakken, and Douglas Walton. 2007. "The Carneades Model of Argument and Burden of Proof." *Artificial Intelligence* 171: 875–896.

Prakken, Henry. 2010. "An abstract framework for argumentation with structured arguments." *Argument & Computation* 1: 93–124.

Segaran, Toby. 2007. *Programming Collective Intelligence.* O'Reilly.