understanding of Locke by the author, and so suggests that, like the first paper, this paper is also of poor quality. Moreover, the claim that inconsistencies "are unavoidable anyway" casts serious doubt as to the author's understanding of the philosophical enterprise generally. In the third paper, the author discusses, *inter alia*, "Locke's failure to clarify the rule of parliament in relation to the community (or state of nature) as a whole," and "problems of the revolutionary allegiance to the king after the colonist break from Great Britain, considering that a state of nature had not been created." Like the second paper, the description of this paper indicates that the author lacks a coherent understanding of Locke. Hence, it seems all three papers are of poor quality.

My suggestion, therefore, is that, at least *prima facie*, paper mill websites of this kind are not a serious threat to teaching philosophy

Endnotes

1. See "Confronting Plagiarism," *Academe* 86:3 (May-June 2000) and Jeffrey R. Young, "The Cat-and-Mouse Game of Plagiarism Detection," *The Chronicle of Higher Education* (July 6, 2001). Young informs us that plagiarism-detection software by Louis A. Bloomfield, a physicist at the University of Virginia, is available at http://plagiarism.phys.virginia.edu

2. Ellen Laird, "Internet Plagiarism: We All Pay the Price," *The Chronicle of Higher Education* (July 13, 2001): B5.

3. For a discussion of writing assignments for undergraduate philosophy courses see [self-identifying reference omitted].

4. The search was conducted on July 14, 2001.

Argument Mapping with Reason!Able

Tim van Gelder
University of Melbourne
tgelder@unimelb.edu.au

Reason!Able is educational software supporting argument mapping. This essay introduces computer-supported argument mapping as an alternative to prose as a medium for reasoning and argumentation, reviews the main features of Reason!Able, and discusses the use of Reason!Able in teaching reasoning skills.

1. Argument and Prose

Reasoning and argumentation are at the very heart of philosophy. A series of classic arguments is a large part of our subject matter, and argumentation is our primary means of making progress. And one of the main benefits of studying philosophy is that it is supposed to enhance reasoning abilities.

As a profession, we have standard practices for handling reasoning and argumentation. One feature of these practices is so familiar and pervasive that it is almost invisible: the *medium* of philosophical argumentation is *prose*. We spend a great deal of time articulating arguments in written prose, and identifying arguments in the writings of others. The dominance of prose goes beyond writing; even when discussing arguments or jousting philosophically, we are using prose, albeit in its spoken form.

Sometimes we do use other methods. Occasionally, for example, we shift from standard natural-language prose into the medium of formal logic. And even when using prose, we add special terminology, strategies and conventions. Yet these idiosyncrasies don't alter the fact that, overwhelmingly, philosophers handle arguments in prose.

Interestingly, in this regard little has changed in thousands of years. That is why we can expect our undergraduate students to engage as productively with the writings of Plato and Aristotle as they can with the latest textbooks and journal articles. We do have new technological supports such as word processors and email. What we do with this new technology, however, is very much the same as would have been done 200 or 2,000 years ago. Descartes hand-wrote letters to Queen Christina; we now send Word documents as email attachments. But these are superficial differences; in both cases, the philosophical work is largely a matter of expressing arguments in lengthy concatenations of words and sentences.

Is this constancy simply due to the fact that philosophical argument is somehow *essentially* prose-based? Not at all. As already noted, philosophers find that certain arguments are best handled by shifting to symbolic logic, though formal techniques are only useful in a narrow range of cases. However there is now emerging another alternative to prose, one which is naturally suited to the vast range of argumentation which is intrinsically informal. That alternative is computer-supported argument mapping.[1]

2. Argument Mapping

Any argument can be understood as a structure of claims standing in inferential or evidential relationships to each other. An argument map is a presentation of an argument in which the inferential structure is made completely explicit, usually by graphical techniques. The typical argument map is a "box and arrows" diagram in which the nodes correspond to claims and the links indicate their evidential relationships.

Argument mapping is the activity of producing (or, more generally, using) argument maps. The activity is thought to have originated with J.H. Wigmore, who early last century used mapping techniques to complex evidential structures in legal cases (Wigmore, 1913).[2] Closer to our time, Stephen Toulmin in *The Uses of Argument* (Toulmin, 1958) used maps to illustrate his theory of the general structure of informal arguments.[3] Most philosophers, however, will be familiar with argument maps mainly as the simple structure diagrams found in many introductory logic or critical thinking textbooks (e.g., Govier, 1988).[4]

A great deal of philosophers' work involves articulating and communicating arguments, and identifying arguments as communicated by others, so you might have thought that a means of presenting arguments in which inferential structure is made completely explicit would be deemed very useful. Yet argument mapping has never really taken off among philosophers. One of the most important factors behind this neglect is that it just hasn't been easy to for your average philosopher to produce, modify and distribute diagrams of any kind, let alone diagrams of complex arguments. Given the range of tools that nature has provided (e.g., voices) and those we have developed (pens, paper, printing presses, etc.) the obvious choice for handling argument has always been prose: ever-available, cheap and easy to produce, and infinitely malleable.

3. Computer-supported Argument Mapping

This is changing. Equipment such as the personal computer, graphics software, colour printers, overhead projectors, email attachments and websites mean that producing, presenting and distributing diagrams of quite professional appearance is now fairly straightforward for all but the most technologically challenged philosophers. Using such tools, pioneers have found that even massively complex philosophical debates can be effectively mapped; the most notable example, of course, being Robert Horn's argument map series *Can Computers Think?*[5] The latest development is the arrival of software designed from the outset to support argument mapping. A number of teams around the world are developing software packages which make it easy to assemble and modify "box and arrow" argument maps. Of those publicly released, the best examples are Reason!Able[6], Araucaria[7], and Athena[8]. With only a small amount of training, philosophers using such tools can produce arbitrarily complex argument maps at least as quickly and easily as they can generate the corresponding prose. Argument-mapping software packages can also provide users with greater power over their arguments (or at least, the presentations thereof): power to view, manipulate, annotate and display in new ways.

In what follows, I will illustrate computer-supported argument mapping using Reason!Able, a package we have been developing over a number of years at the University of Melbourne and Austhink.[9] Reason!Able is educational software, designed to be used in undergraduate critical thinking classes. It has however been picked up and used in many different contexts and at many different levels, both inside and outside the academy.

4. Reason!Able Features

4.1. Building Argument Trees

Reason!Able provides a workspace within which click and drag operations are used to build and modify hierarchical "tree" structures representing the inferential relationships among the various claims which make up argument.

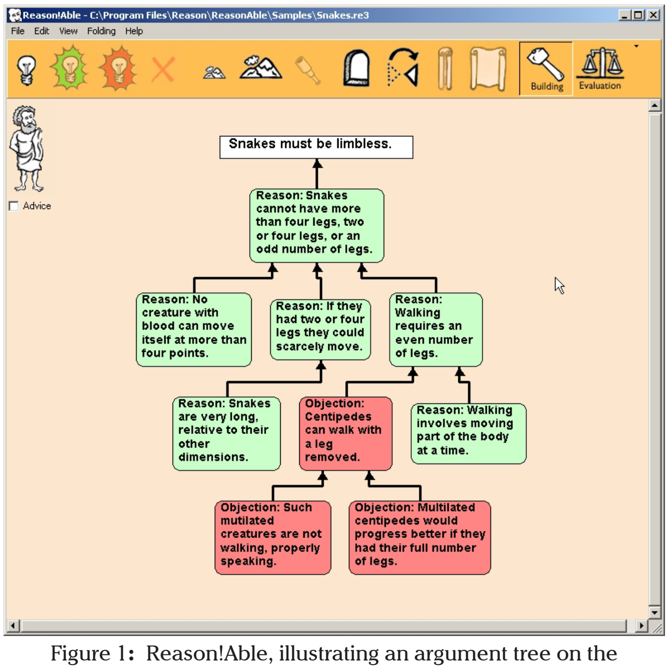

Figure 1: Reason!Able, illustrating an argument tree on the workspace. This argument map presents Aristotle's' reasoning in support of the claim that snakes *must* have no legs, from his *On the Gait of Animals*.

The primary objects in a Reason!Able-style argument tree are claims, reasons and objections. (As will be explained below, reasons and objections are themselves groups of claims.) A claim is represented by a white box; reasons are green boxes and objections are red boxes. Sentences expressing the relevant claims are written in the boxes. In this respect, Reason!Able differs from many other argument mapping schemes and programs, which don't put the full text in the nodes themselves, but hold them in a separate list, thereby creating a heavy cognitive burden for the user who must mentally pair nodes with sentences.

In the argument tree, a "child" is always evidence for or against a "parent." Thus in Figure 1, there is one reason providing evidence for the main conclusion; that reason is supported by three secondary reasons; there is an objection to the third of those primary reasons, to which there are two rebuttals; and so on. Note that because the reasoning is presented in a diagram, you can see all this structure at a glance.

Additional reasons and objections can be added to any node on the tree by selecting that node and then just clicking on the appropriate button on the toolbar. In this way, you can rapidly assemble arbitrarily complex argument trees.

4.2. Viewing Argument Trees

Given the size and resolution of contemporary monitors, with even moderately complex arguments it soon becomes impossible to see both the forest and the trees (i.e., the structure of the whole argument and the contents of the individual nodes) at the same time. Thus Reason!Able provides various mechanisms for changing view on the argument:

- **Zooming**. The user can zoom in or out by increments; can zoom in one click to a size at which the entire argument fills the window; and can select any area on the workspace and zoom in to that area.
- **Panning**. As you would expect, panning across the workspace can be achieved by scrolling. It can also be achieved by dragging a rectangle representing the current view on the workspace within a small overview window.
- **Rotating**. The argument can be viewed in any one of four orientations (top-down, L-R, R-L, bottom-up). Sometimes rotating can make for a more revealing layout.

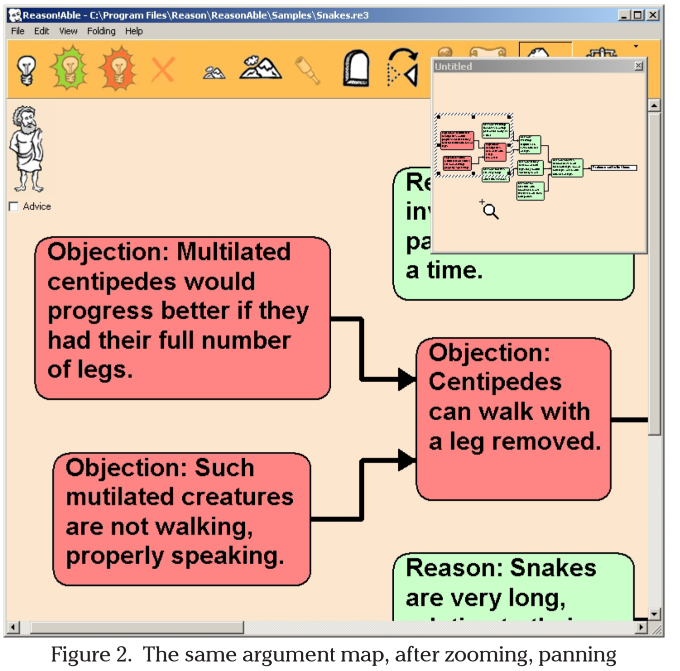

Figure 2. The same argument map, after zooming, panning and rotating. This allows the user to focus on a particular piece of reasoning. One click on the "Fit to Window" button will zoom out so that the structure of the entire argument can be viewed. In the upper right hand corner there is an overview window, which shows the "forest" and provides an easy way to zoom and pan.

4.3. Editing and Modifying

The text inside the nodes can of course be edited in the normal way. More significantly, the argument tree can be reorganized at will by drag and drop operations. Nodes or branches can be torn off the tree, or relocated to new positions.

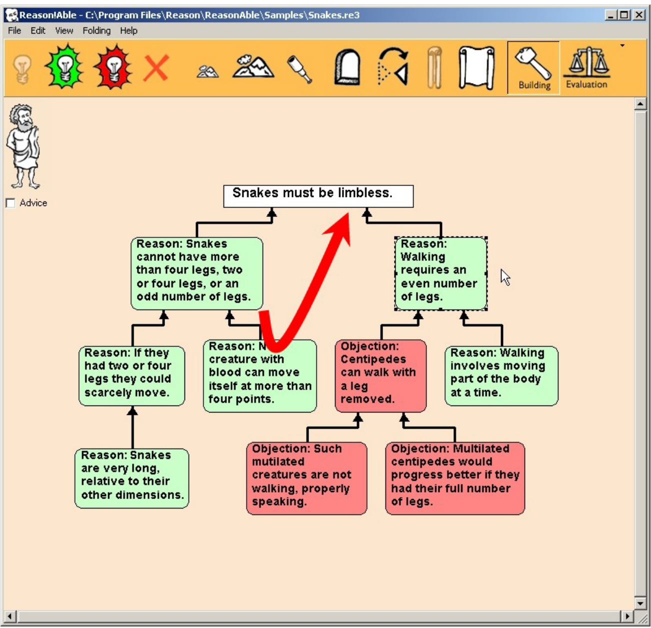

Figure 3. The argument from Figure 1, after a single drag-and-drop operation in which an entire branch of the argument was relocated so as to be attached directly to the main conclusion.

4.4. Premises

A key feature of Reason!Able is that reasons and objections are always complex objects, made up of sets of claims (premises) working together. Consider the classic philosophical argument**:**

P1**:** Socrates is a man.
P2**:** All men are mortal.
C**:** Socrates is mortal.

The argument has two premises, but how many distinct *reasons* have been provided? Only one, and both the premises work together as part of this reason.

In Reason!Able, reasons are initially represented as single green boxes containing the main premise, but they can be "unfolded" to show the full set of premises ("helping premises," or "co-premises").

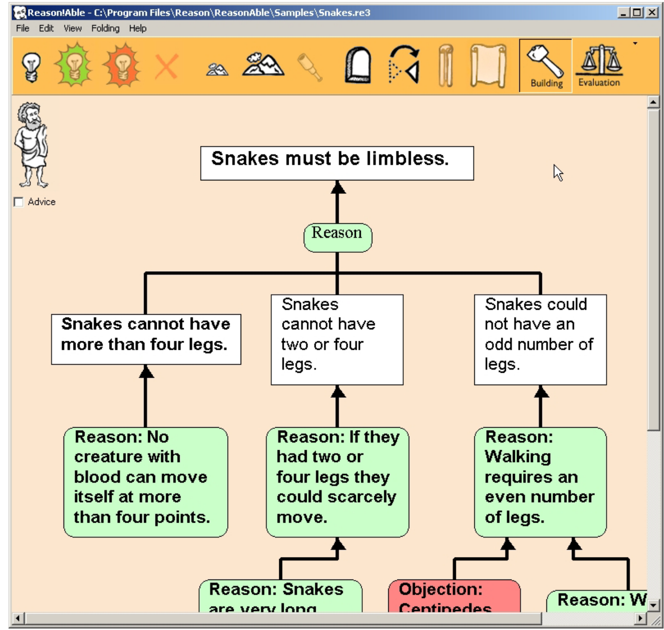

Figure 4**:** The primary reason of Aristotle's snake argument is unfolded to reveal that it has three distinct premises, which work together to provide evidence that the conclusion is true. Distinct premises are separately debateable; in this case, each premise has been provided further supporting evidence.

By default a reason has two premises (a main premise and one co-premise) but additional premises can easily be added by clicking on the "claim" icon on the toolbar. Premises can be moved around by dragging and dropping in much the same way as whole reasons or objections.

Objections, of course, are constructed from claims in the same way as reasons.

4.5. Evaluating Arguments

Thus far we have considering the *structure* of arguments, and how Reason!Able supports assembling, viewing and modifying argument structures. Colour has been used to indicate the *type* of object**:** white for claims, green for reasons and red for objections. In philosophy, however, we are at least as interested in the quality of arguments, and in assessing quality we make various evaluative judgements. The verdicts we reach constitute further information which can be represented on the same argument tree.

Reason!Able has two primary modes, Build and Evaluate. In Evaluate mode, three kinds of evaluations can be represented**:**

Kind of evaluation	Possible evaluations	Represented by
Strength of reasons/objections	Conclusive (deductively valid with definitely true premises), Strong, Weak, None	Intensity of colour of reason/objection
Degree of confidence in the truth of claims	Definitely, Probably, No Verdict, Probably False, Definitely False	Intensity of colour of claim (blue for true)
Independent grounds for accepting a claim as true	Common knowledge, Personal knowledge, Testimony, Expert opinion, Necessary truth, Considered plausibility	An icon attached to the claim

It is certainly possible to be more sophisticated in one's choice of evaluative dimensions and values. For example, the simple range of discrete values for degrees of confidence could be replaced by a numerical scale. The options built into Reason!Able were chosen to meet two dominant criteria: (a) maximizing the utility of the package as an educational tool, and (b) providing a tool which "makes sense" to ordinary people dealing with real-world argumentation.

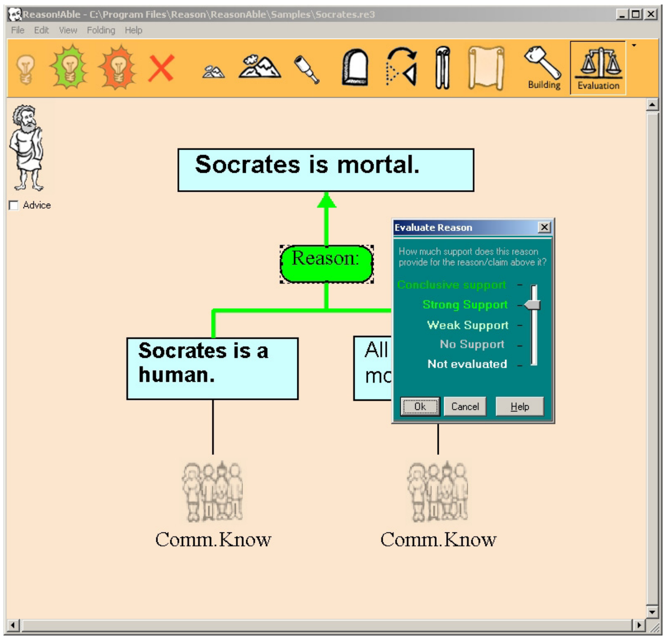

Figure 5: Evaluate mode. This simple example illustrates how evaluative information is represented. Claims have been rated as probably true (light blue); the premises are both accepted as probably true on the grounds of common knowledge; and the whole reason is being evaluated as offering strong support (mid green).

When evaluative information is represented on the trees of complex arguments, strengths and weaknesses (including "fault lines") are immediately visually apparent. The overall effect is akin to having a satellite photo of a region of the country, in which city, farmland, forest and water can be instantly distinguished by vivid colour differences.

4.6. Guidance

As mentioned, Reason!Able was developed as an educational tool. Undergraduate students typically have only the foggiest grasp of the concepts and procedures involved in analyzing and evaluating arguments. In order to help them learn, the software provides guidance in the form of context-sensitive instructions from "Socrates," a character similar to the infamous paper clip in the Office software suite. When he is switched on, clicking anywhere on an argument tree will prompt Socrates to proffer a piece of advice pertinent at that point and at that stage of the process.

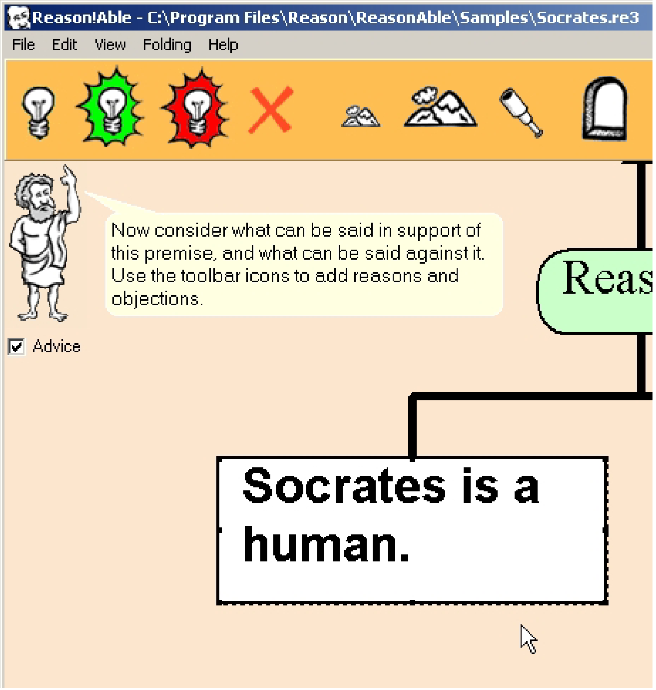

Figure 6: Socrates' context sensitive advice.

Socrates provides two major kinds of advice. One is for critical evaluation; it guides the student through the process of identifying an argument as presented (in prose) by another person, and evaluating that argument. The other is to guide the student in the process of producing their own argument, and evaluating it to ensure that it is a strong one.

We find that students rapidly get the hang of what Socrates is going to say, and prefer to switch him off. This is good; these students have internalized the steps involved in systematically handling an argument.

5. Reason!Able in Critical Thinking Instruction

One domain within which computer-supported argument mapping has already been extensively deployed is in teaching the general skills of reasoning and argument. For three years Reason!Able has been the primary learning vehicle in a large, one-semester undergraduate Critical Thinking subject at the University of Melbourne. The subject has been intensively evaluated to determine the extent to which students actually improve their critical thinking skills. The data gathered so far suggest that an approach based on computer-supported argument mapping is substantially more effective than traditional methods.[11]

The Reason!Able software is a central part of what we call the Reason! approach. The conjecture driving this approach is that critical thinking is a skill, and that skills improve through "quality practice." Quality practice is practice with certain features: it must be motivated, guided, graduated, scaffolded, and feedback-modulated. In addition, for a general skill such as critical thinking, it must be practice-for-transfer – that is, practice in the transferring of skills from one domain or context to another. The fundamental challenge is how to get students doing lots of practice with those features, within the constraints and limited resources of an undergraduate subject.

To help address this challenge, Reason!Able was developed to function as a "quality practice environment," intended to help students engage in better quality practice than they would using traditional methods. In particular, Reason!Able provides guidance and heavy scaffolding, and facilitates more targeted feedback. Students use the software in dozens of exercises which become gradually more challenging as the semester progresses. The two main kinds of exercises are *critical evaluation*, in which they identify and evaluate the reasoning of others as expressed in prose, and *production*, in which they generate and evaluate their own arguments (and perhaps go on to express those arguments in prose).

Does it work? Each batch of students is pre- and post-tested using the California Critical Thinking Skills Test, a 34 question multiple-choice test. Over the past three years we have found that students on average improve their score by almost 4 points, or about 0.8 of a standard deviation. (For two years we also used a written test, which found gains of the same order of magnitude.) This may not sound much, but consider that students would normally be expected to improve by about 0.5 of a standard deviation over three years of college. The Reason! approach thus dramatically accelerates growth in critical thinking skills, relative to undergraduate education. Alternatively, consider that a gain of equivalent magnitude in IQ would be one point per week.

How does this compare with traditional approaches? This is a bit hard to say, since disturbingly little is really known about the effectiveness of traditional one-semester critical thinking or introductory logic. We are currently engaged in an extensive survey of relevant empirical literature. The bad news is that traditional subjects appear to make little if any difference. Our best estimate, at the moment, is students in traditionally taught first-year undergraduate subjects improve by not more than 0.1 of a standard deviation over and above the amount they would have improved anyway due to growing up and being at university, which is about 0.2. If this is correct, then the Reason! approach is around 6 times more effective.

Why is this? We have been taking detailed measurements of the amount of practice students are doing, and so far are *not* finding strong correlations between amount of practice and gain. My hunch is that the other obvious difference between Reason! and traditional approaches – the use of computer-supported argument mapping – is largely responsible.

6. Argument Mapping in Philosophy Instruction

Many philosophers, even if not teaching critical thinking or introductory logic, work hard to help their students improve their general reasoning and argument skills. This is a slow, difficult and often frustrating business. Computer-supported argument mapping, using a package such as Reason!Able, may help instructors be more effective in this respect, no matter what their subject (ethics, philosophy of mind, etc.).

Here are some relatively straightforward pedagogical strategies:

When setting argumentative essay assignments, require students to hand in a map of their main argument along with their essay. Students will find that expressing their reasoning in an argument map requires that they be much more clear and explicit about what that reasoning is, and it gives them a logical backbone on which to hang their essay. When it comes to grading their work and giving feedback, you'll find that having their argument map is like having x-ray vision into their thinking (though this is generally not a pretty sight).

Require students, when doing their reading, to map the author's main line of argument. Tell them that reading properly consists in understanding the text to the point where mapping the argument is a straightforward matter. This will give most students a whole new perspective on what it is to engage seriously with a philosophical text.

When lecturing, display arguments (whether your own, or those you are discussing) in map form. This can be done in a variety of ways. One is to print out the argument map on a transparency and display it using an overhead projector. A better way, for those with both the technical agility and a suitably equipped classroom, is to do "live" argument mapping, projecting from a PC running argument mapping software.

In tutorials, if facilities allow, project an argument map and use it as the basis of discussion. Arguments or debates can be mapped in real time, and you can require students to make their contributions in the form of additions or modifications to the argument tree.

7. Future Directions

From the brief tour of Reason!Able given above, it should already be apparent that handling arguments in computer-supported argument mapping mode can be a very different experience than is had when using the traditional spoken or written prose. Argument maps represent information more densely than prose, and make that information more immediately available to the mind, by using a wider range of representational resources (colour, line, shape). Computer software supports a wider range of interactions with these maps. The abstract complexity of argumentation has become more visual, concrete, and manipulable.

That said, it is also important to realize that these are early days in the development of computer-supported argument mapping. Back in 1962, Douglas Englebart imagined and predicted computer-supported argument mapping as a means of augmenting human intellect.[12] Four decades later, his vision is at last starting to be realized. Reason!Able (and other packages available today) are like Model T Fords compared with the automobiles of today, let alone the "maglevs"[12] of the future. I brashly predict that once the technology becomes sufficiently advanced, those who deal with complex arguments for a living will switch to the new methods just as the accounting profession has switched entirely to computer packages in preference to the old system of ledgers and manual entries and calculations.

References

1. Kirschner, P. Buckingham Shum, S., & Carr (Eds.) (Forthcoming). Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making. London: Springer-Verlag.

2. Wigmore, J. H. (1913). *The principles of judicial proof : as given by logic, psychology, and general experience, and illustrated in judicial trials*. Boston: Little, Brown.

3. Toulmin, S. (1958). *The Uses of Argument*. Cambridge: Cambridge University Press.

4. Govier, T. (1988). *A practical study of argument* (2nd ed.). Belmont, CA: Wadsworth.

5. Horn, R. E. (1999). *Can Computers Think?* : MacroVU, Inc.

6. van Gelder, T. J., & Bulka, A. (2000). Reason!Able (Version 1.1). Melbourne. http://www.philosophy.unimelb.edu.au/reason. Reason!Able is a stand-alone software package for computers running Windows 95 and above. It can be obtained from http://www.goreason.com.

7. Reed, Chris and Rowe, Glenn. University of Dundee Araucaria. http://www.computing.dundee.ac.uk/staff/creed/research/araucaria.html

8. Rolf, B., & Magnusson, C. (2002). Developing the art of argumentation, a software approach, *Proceedings, ISSA 2002*. http://www.athenasoft.org/

9. The Australian Thinking Skills Institute, http://www.austhink.org

10. van Gelder, T. J. (2001). How to improve critical thinking using educational technology. In G. Kennedy & M. Keppell & C. McNaught & T. Petrovic (Eds.), *Meeting at the Crossroads: Proceedings of the 18th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education* (pp. 539-548). Melbourne: Biomedical Multimedia Unit, The University of Melbourne.

11. Engelbart, D. (1962). *Augmenting Human Intellect: A Conceptual Framework*. Menlo Park, CA: Stanford Research Institute

12. Maglevs are vehicles in the recent movie *Minority Report*, set in 2054. The film also portrays a wall-sized information display controlled remotely using data gloves. It will not be long - much less than five decades - before complex argumentation is processed on this kind of interactive display.

Tools and Techniques

Indexing a Book—Fast and Easy

Lawrence M. Hinman
hinman@sandiego.edu

Indexing a book has always been tedious work. Occasionally, it may rise to the level of a work of art—as, for example, the index that Rawls did for the original edition of *A Theory of Justice*. Indeed, years earlier Rawls and his wife Margaret Fox spent their first summer of marriage doing the index for Walter Kaufmann's *Nietzsche*. But for the rest of us mere mortals, indexing remains an onerous task. I would like to offer a way of making it less onerous, even if it will not rise to the level of Rawlsian greatness.

Modern word processing programs such as *Microsoft Word* appear to make indexing easier, but they leave us with a crucial question unanswered: if I use *Word* (or *WordPerfect* or some other comparable program) to compile an index to my book-length manuscript, how can I get the pages in the manuscript to match the pages in the printed version of the book? And if I can't do so, then what use is the index?

There is, in fact, a comparatively easy answer to this question, and it will allow you to prepare indices to books quickly and accurately. Let's presuppose that you now have the final page proofs of your most recent book.

The first step is to assemble all the chapters of your book in a single computer file, if you have not already done so. Also include the preface, forward, introduction, appendices and bibliography, but not the table of contents. Make sure that everything is in the proper order, that is, that it follows the printed version.

Here's the second step. Some introductory material such as the preface may be paginated with lower case Roman numerals; the rest of the book will be standard Arabic numbers. Make sure you insert a section break between that introductory material and the body of the text; then set the body of the text (presumably beginning with Chapter One) to begin with the numeral "1."

Now you are ready for the third step: making the page numbers in your computer file correspond to the page numbers in the printed version of your book. In order to do this, you must replace automatic page breaks with manual page breaks. But before doing that, you must make sure that no automatic page breaks occur accidentally in your file. You can accomplish this quite simply: just change the default length of your page to something fairly long, say twenty inches.

Once you have set the default page length to twenty inches, go to page 1 in the printed book. Look for that same spot in the computer file. Set your page numbering in this section of the computer file to begin with "page 1." In Microsoft Word, you do this by using the following command:

Insert | Page Numbers...| Format | Page numbering start at...

Look at the end of page 1 in your printed version, and then enter a hard or manual page break at that point in your computer file. Now "page 2" begins at the same place in both the computer file and in the print version. Look at the end of page 2 in the printed version, and place a hard page break at that point in your computer file. Simply continue to do this until you reach the end of the manuscript. If there is a break page in the printed book, enter two hard page breaks, etc. You will quickly get the hang of it and be able to do a couple hundred pages in an hour.

Once you have finished paginating the main section of the manuscript, go back to the prefatory material. Let's say you have only a preface to be indexed, and that it begins on page viii. Set your section up in *Word* to paginate beginning with "page viii." Then check your preface against the printed version, again entering hard page breaks to correspond to the printed version.

At this point, you should now have a computer manuscript whose pagination is identical to the pagination in your printed final proofs. You can now compile an index using your word processing program's built-in indexing tools, and the resulting index will have a set of page numbers that corresponds exactly to your printed version. Typically, programs such as *Microsoft Word* offer powerful and easy indexing features. For example, if you mark the word "Kant" as a word to be indexed, *Word* will automatically find all other instances of that same word and list them in the index, so you only have to mark the word once for indexing and the program does the rest. You can also create sub-entries. Say that you had an entry for "emotions," you can then have sub-entries for "Aristotle," "Stoics," and "Kant." You can also have Word index a range of pages on a particular topic and do cross-references as well.

Once you have finished marking all your entries, make sure that all field codes are hidden, go to the end of your document, and then click on the command:

Insert | Reference | Indexes and Tables | Index

and choose the options you prefer for the index. Word will then compile your index in the format you prefer. Copy and paste (or print) the index and send it to you publisher and you're done.