Araucaria: Software for Puzzles in Argument Diagramming and XML

CHRIS REED & GLENN ROWE

Department of Applied Computing, University of Dundee, Dundee DD1 4HN Scotland

[chris, growe]@computing.dundee.ac.uk

Abstract. Argumentation theory involves the analysis of naturally occurring argument, and one key tool employed to this end both in the academic community and in teaching critical thinking skills to undergraduates is argument diagramming. By identifying the structure of an argument in terms of its constituents and the relationships between them, it becomes easier to critically evaluate each part of an argument in turn. The task of analysis and diagramming, however, is labor intensive and often idiosyncratic, which can make academic exchange difficult. The Araucaria system provides an interface which supports the diagramming process, and then saves the result using AML, an open standard, designed in XML, for describing argument structure. Araucaria aims to be of use not only in pedagogical situations, but also in support of research activity. As a result, it has been designed from the outset to handle more advanced argumentation theoretic concepts such as schemes, which capture stereotypical patterns of reasoning. The software is also designed to be compatible with a number of applications under development, including dialogic interaction and online corpus provision. Together, these features, combined with its platform independence and ease of use, have the potential to make Araucaria a valuable resource to the academic community.

Keywords: Argumentation schemes, argumentation theory, critical thinking, diagramming, informal logic, XML.

1.Introduction and Motivation

Argumentation theory aims to analyse, describe, and evaluate arguments that occur in the real world, and occurs as a topic in many undergraduate syllabi, where it aims to teach students both to think critically about the arguments of others, and to create better, more measured arguments of their own. One of the key tools available to the discipline is diagramming. The claims and their associated reasons within a given argument are identified, and the relationships between them drawn up as trees. This diagram then serves as a basis for criticism and reflection.

Computer software might well be anticipated to be highly suited to the task of visualisation, and particularly to the sort of diagrams used in argumentation theory. Similarly, it is also well suited to the task of aiding an analyst in constructing the diagrammatic representation of an argument. And yet, there are very few computer systems which support argument diagramming for the student, and none at all which support the diversity and sophistication of analyses formed within the research community. It is this dearth of computer support for a labor intensive but crucial activity, which is addressed in this paper.

2.Background

The development of informal logic and argumentation theory within philosophy has represented a backlash against post-Fregean formal logic, which though immensely powerful and widely applicable, is a poor choice for representing and characterising natural – i.e. real-world – language and argument, despite its Aristotelian heritage aimed at just that. The inception of informal logic marked by Toulmin (1958), Perelman and Olbrechts-Tyteca (1969), and others saw a return to an empirically driven logic.

Within argumentation theory, systems of diagramming argument have played an important practical role, in two distinct areas. The first is in pedagogy: employing diagrams in support of the teaching of critical thinking skills. Though opinion is divided as to the degree to which diagramming is useful for all students (see, for example, (Argthry, 2001)) there is clear evidence that the technique is of great benefit for some (van Gelder & Rizzo, 2001). These results concur with more general results in the psychology of reasoning (see (Rips, 1994: 347-349) for an introductory discussion of this issue).

The driving force provided by the need to teach critical thinking, and the rise in utility and subsequent popularity of computer assisted learning packages, has led to the appearance of a number of software systems for argument diagramming which are intended for pedagogical use. One of the most sophisticated and polished examples is the Reason!Able system (van Gelder & Rizzo, 2001). This system aids students – particularly those learning informal reasoning skills at an introductory level in schools and universities (and, they report, even kindergarten) – in constructing and analysing argument maps. These maps employ arrows and colours to indicate support and rebuttal relationships and are manipulated through a straightforward interface that is appealing to children. The preliminary results reported in (van Gelder & Rizzo, 2001) suggest that the software has, at the very least, the potential to substantially improve students' critical thinking skills. For van Gelder, as with the work described below, the focus is on the software itself; there are other examples where the focus is on other resources (typically a textbook) and the software plays more of a supporting role. A good example of this is LeBlanc's Lemur software which accompanies her text (LeBlanc, 1998), and supports dynamic student interaction with the exercises with which each chapter is brought to a close. Some of these exercises involve basic diagramming, but as the software is encompassing the range of techniques introduced in the text, the diagramming components themselves are rather limited and constraining, and unlike Reason!Able, are restricted to only those examples provided in the software.

The second role of diagramming is in the construction and implementation of theories of argument evaluation within the research community. One of the earliest methods, now the textbook favourite, is that proposed by Beardsley (1950), and enhanced with nomenclature by Thomas (1986). More recently, inadequacies and problems with this *standard treatment* have been identified, leading several authors to propose alternatives, e.g. (Freeman, 1991) which extends the standard treatment to deal with structures described by Toulmin (1958); (Reed, 1999) which explores an alternative method of handling linked structures; and (Wilson, 1986) which emphasises the evaluative aspect of argument analysis by including it explicitly in the diagram. The various approaches to the issue of diagramming, of which these are just a few, represent scholarly approaches to problems that lie at the heart of argumentation theory and informal logic.

Although there has not to date been software specifically designed to support research into argumentation and diagramming, there have been a few systems which impact upon that research. Foremost amongst these is the ambitious and far-reaching *Archelogos* project under development at the University of Edinburgh (Scaltsas, 1997), which aims to analyse and mark-up substantial portions of the argumentation in the oeuvres of both Plato and Aristotle, and provide an interface that allows online navigation of the structure of the reasoning in the works. The *Archelogos* project does not, however, focus upon diagramming the structure that is produced through analysis. In contrast, work in linguistics, and in particular in pragmatics, aiming to analyse interclausal relations, uses software tools to build diagrammatic analyses of textual structure. RSTtool (O'Donnell, 1997) is a good example of such software, and it has been argued (Mann and Thompson, 1988) that the approach can be applied to argumentative text (just as it can to any other genre). These linguistic research projects make no use, however, of the rich analytical structures and techniques of argumentation theory.

Finally, argumentation itself has found many applications within computer science, and various branches of artificial intelligence in particular. A review of many of these systems can be found in (Reed, 1997), whilst a more recent analysis of interdisciplinary work between argumentation and each of multi-agent systems, legal reasoning decision support, computational linguistics, and contextual reasoning, can be found in (Reed *et al.*, 2002).

The focus here, however, is squarely upon software to support both teaching and research in argumentation theory. To the authors' knowledge, there is currently no system which provides such support, and it is this gap that the *Araucaria* system fills.

3.Araucaria

The Araucaria system does not attempt to tackle fundamental restrictions of the diagramming process. As with other methods of analysing textual structure – such as Rhetorical Structure Theory, RST (Mann and Thompson, 1988), for example – any given analysis is potentially disputable. RST offers a means of specifying the relation that holds between spans of text – though both the judgements concerning the delimitation of text structure phrases, and the identification of relationships between those phrases, can be challenged. Mann and Thompson suggest that in marking up the rhetorical structure of a text, the analyst makes *plausibility judgements* (rather than absolute analytical decisions) and that there can be more than one reasonable analysis. The assumptions behind Araucaria follow the same pattern: a single text might be analysed in several different ways, depending upon a variety of analytical choices.

As in RST, the judgements concerning the delimitation of argument components can vary, depending upon the aims of the analyst and the clarity of the text itself. Example (Ex1) is an excerpt from an extended argument taken from a US Supreme Court case, and used in a current textbook (Johnson, 1998: 328).

> *(Ex1)* *It [the Court's decision] permits the state's abstract, undifferentiated interest in the preservation of life to overwhelm the best interests of Nancy Beth Cruzan, interest which would, according to an undisputed finding, be served by allowing her guardians to exercise her constitutional right to discontinue medical treatment. ... Because Nancy Beth Cruzan did not have the foresight to preserve her constitutional right in a living will, or some comparable "clear and convincing" alternative, her right is gone forever and her fate is in the hands of the state Legislature instead of in those of her family, her independent neutral guardian ad litem, and an impartial judge – all of whom agree on the course of action that is in her best interests.*

One defensible analysis would include four components in the first sentence, and three in the second. Johnson's analysis, in contrast, considers each of the two sentences to be indivisible units. In the context of the analysis, the reasons for his choice are clear, namely, to ensure that the analysis is at an appropriate level of abstraction (the text of the example runs over several pages), and to provide a pedagogically sound tutorial.

Also analogously to RST, decisions about the relationships between components may also vary. Freeman (1991) points out, for example, that a given text might be analysed using the Toulmin (1958) schema, and the data and warrant be quite interchangeable. Furthermore, recent work examining *argumentation schemes* – stereotypical patterns of nondeductive reasoning – has demonstrated that a single text might be regarded as instantiating several different schemes, depending on the focus of the analysis (Walton, 1996).

Again by analogy to RST, there is also freedom in analytic resources. Mann and Thompson emphasise that the set of relations they put forward is simply one possible set that has been found to have utility in the analysis of a particular corpus. They claim neither exhaustiveness nor accuracy of their proposed set, instead describing the process by which researchers can produce their own sets of relations. A similar solution is adopted in the provision of *schemesets* of argumentation schemes. Many scholars and teachers of critical thinking and related fields find that argumentation schemes are a useful tool for describing the relationships between argument components. Determining a single, exhaustive, consistent set of schemes has proved difficult – though existing sets such as (Grennan, 1996), (Kienpointner, 1992) and (Walton, 1996) are nevertheless rich and extensive. The importance of argumentation schemes is also growing within various computational applications of argument (Reed and Walton, 2001), so one aim in developing the Araucaria software was to ensure that argumentation schemes were coherently integrated. The choice of which - if any - argument set to use is left to the user, with the standard distribution including not only schemeset definitions corresponding to the Grennan, Kienpointner and Walton lists mentioned, but also software to design custom schemesets in a straightforward manner.

Finally, one analytic freedom with which RST does not have to contend is the process of reconstruction, and in particular, of supplying missing premises. (This is one of the reasons that RST is often an inappropriate tool for analysing argumentation, (Reed and Long, 1998; Reed, *to appear*)). In argument analysis, however, argument reconstruction forms a critically important phase (van Eemeren *et al.*, 1993), because arguments are typically heavily abridged. Enthymemes – arguments (or, specifically, syllogisms) with one or more components left implicit – are extremely common, to the extent that the natural language expression of a Modus Ponens argument (A, A implies B, therefore B) is so frequently contracted through the omission of the major premise (leaving just: A, so B) that it has led linguists to regard it as a separate form of argument altogether – the Modus Brevis (Sadock, 1977). Though there are various patterns to these contractions (such as those described by Sadock and in (Reed, 1999b)) the software itself needs only to support an analyst's work at reconstruction. This support is provided in allowing new premises not explicitly included in the text to be added to the structure of an argument.

The emphasis upon comparison with Rhetorical Structure Theory is quite deliberate. By accepting the diversity not only of language, but also of the interpretation and analysis of language, RST has become a powerful and widely used tool in discourse analysis and computational linguistics, and has played a key role in making common resources available to the research community. By equipping argument analysis tools with a similar flexibility and tolerance of analytic diversity, the rich variety of approaches in teaching, learning, and research can be preserved whilst at once providing a common interlingua and environment for carrying out those activities.

As part of the commitment to supporting diversity, Araucaria has been developed in Java, to support execution on many platforms. The software has been tested under various versions of Microsoft Windows, Solaris, Linux and MacOS.

System Overview

The system can load either a text file or an existing, marked up, argument. In either case, the left-hand pane shows the original text of an argument. Selecting part of this text and dropping the selection on to the right-hand, diagram, pane creates a node which corresponds to that text. There are then a number of actions which structure and combine the nodes in the diagram:

- Dragging a line from one node to another creates a support relationship, indicated by an arrow from premise to conclusion (supporter to supported)
- Multiple lines of support can be selected and 'linked' together (and then subsequently unlinked back to convergent support – Araucaria adopts the terminology of Freeman (1991), *inter alia*)
- Selecting either multiple lines of support or multiple nodes allows the selection of an argumentation scheme to be associated with those nodes. Schemes are identified by coloured areas on the graph, and can be overlapping (that is, a single text span can play a role in several argumentation schemes – typically as a premise in one and a conclusion in another)

In addition, the menus also support a range of further functionality:

- The addition of missing premises to the diagram
- The creation, modification, loading and saving of argumentation scheme sets
- The deletion of components from the diagram

File manipulation supports the saving not only of the marked up argument using XML, but also of the diagram itself as a JPEG image for incorporation into documents and online material.

Finally, as opinion seems divided in both the research and pedagogic communities, the entire diagram can be inverted at any time with a single key-press.

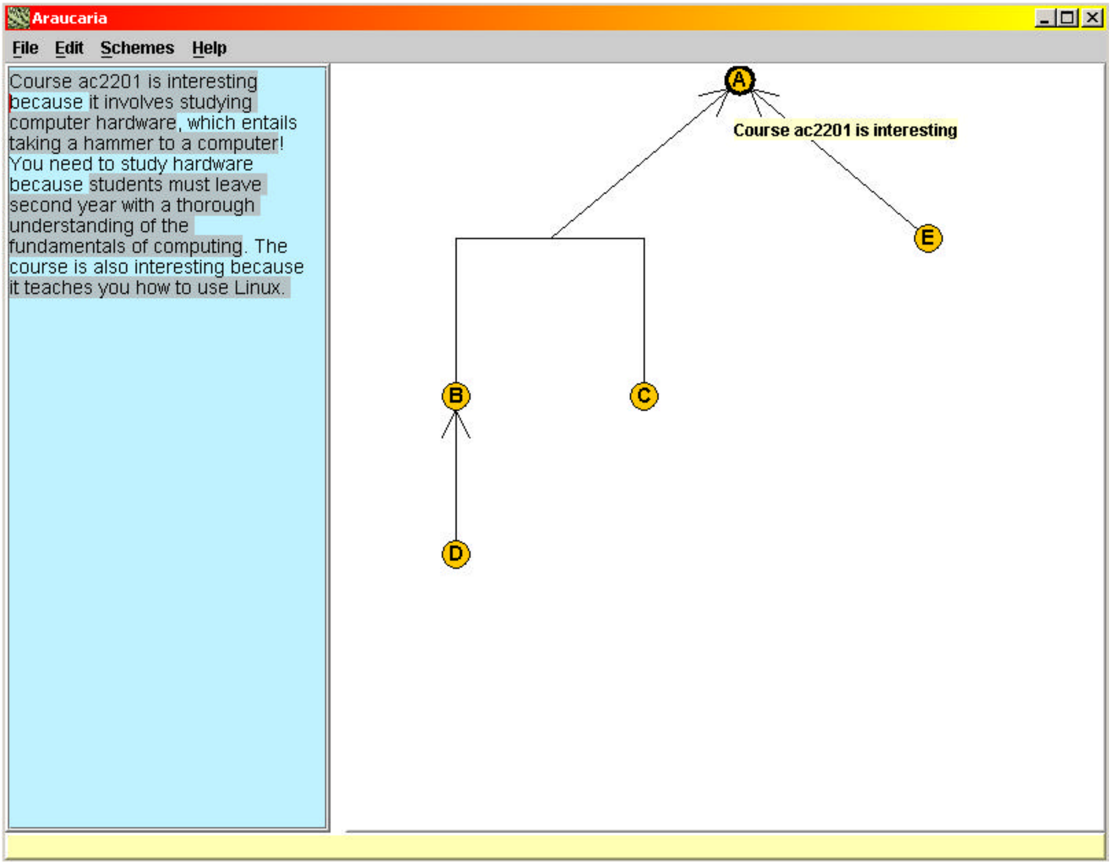

Figure 1. Araucaria

4.AML

The environment provided by Araucaria is one suited to analysis. That is, it is assumed that a sample text is available and that this text is to be analysed to produce a diagram. It is important during this process that links between the original text and the corresponding components of the diagram be maintained: deletions of components of the diagram, for example, rely upon these links. Further, once the analysis is complete, it is useful to save not just the text and diagram, but also the relationship between them, to allow future modification and manipulation.

These factors suggest that marking up the original text is an appropriate approach. By adding to the original text *tags* that indicate the evolving structure, the relationship between text and diagram is preserved. The *argument markup language* (AML) defines a set of tags that indicate delimitation of argument components (loosely, propositions), tags that indicate support relationships between those components, and tags that indicate the extent of instances of argumentation schemes.

Both Araucaria, and the markup language in which analyses are saved, exploit the typical tree structure of argumentation[1]. This means that the markup language can be defined quite succinctly, by characterising an argument recursively as a proposition supported by one or more arguments. Implementation of the markup language employs XML (eXtensible Markup Language) which carries with it a range of benefits. Firstly, XML is a well-defined, well-understood, widely used industrial standard. This means that there are a range of standard tools which can be employed to manipulate the data: one simple example is a sample application available with many XML parsing suites[2]. The application provides a conventional tree view (such as is employed by Microsoft Windows Explorer) that, in the context of the files produced by Araucaria, allows a conclusion to be expanded to show its (immediate) supporting premises, and each of those premises to be expanded to show their supports, and so on. There are similarly tools for creating summaries, diagrams, for verifying content, etc.

Secondly, as a generic data representation language XML files are also easily translatable into other formats through the application of *stylesheets*. A good example of the possible uses of

1Argument analysis diagrams are not always trees (i.e. a node is not always restricted to having a single parent). Setting aside circular reasoning, which can be handled as an argumentation scheme, non-tree structures are required where a single premise supports multiple conclusions – Freeman's (1991) *divergent* structures. These structures are unusual from a formal perspective (see (Shoesmith and Smiley, 1978) for a formal investigation), and are also rare in argumentation practice, with many textbook authors ignoring them altogether (see, e.g., (Groarke *et al*., 1997)). Given the overwhelming advantages of adopting tree rather than graph structures, this is an acknowledged restriction of the Araucaria system. A straightforward workaround in the situation where premise P1 supports conclusions C1 and C2, is to replicate P1, once in its role of supporting C1, and once in its role of supporting C2. There is, in general, no restriction on the uniqueness of premises. The underlying markup language is also restricted in respect of its handling of divergent argumentation, again for pragmatic reasons (parsing the data is quicker and more efficient if a tree structure can be exploited), though it is less of a problem here. Application software – such as Araucaria – that reads or writes to the format could be designed in such a way that divergent structures are created and displayed in the conventional manner, with P1 supporting C1 and C2, but then saved with the replication mentioned above. The development of application software which can handle divergent structures in this way is left to future work.

2The suite referred to here is that provided by Apache at http://xml.apache.org including, amongst other components, the Xerces parser. The application mentioned in the text is called "Treewalker" in the Xerces software.

stylesheets is in the creation of tailored HTML web files. Thus arguments can be automatically summarised or made navigable for online provision. The discussion returns to various such applications in sections 5 and 6, below.

Thirdly, XML has recently been recognised as having the potential to play a significant role in corpus resources (Ide, 2000). Since this is one of the areas of application planned for Araucaria and AML as described in section 6, the adoption of XML leads to several advantages in the development of, publication of, and access to, corpus resources.

Fourthly, the acceptance of XML as a de facto industry standard facilitates data sharing: with a single common format or interlingua, separate applications can share data in the tasks of input, manipulation and output. With the structure of a particular XML markup language defined independently in a document type definition (DTD) file, the definition of AML can quickly and simply be made open to the community through the publication of its DTD .

Finally, the definition of AML in its DTD is independent of the applications in which it is employed. This independence facilitates maintenance and development, whilst permitting backward compatibility. AML is defined in `argument.dtd`, a full specification of the components from which arguments – and argument diagrams – can be constructed. (The file `argument.dtd` can be downloaded from the project homepage at `http://www.computing.dundee.ac.uk/staff/creed/research/araucaria`).
Work is under way to extend the DTD in several specific directions, as described in section 7, but these extensions have been planned during the design of the current version of `argument.dtd`. Arguments marked up according to later formats will thus be compatible with earlier data definitions: that is, newer formats will only add to and extend – and not otherwise modify – earlier formats.

The Araucaria software is thus uncoupled from the AML; the latter can evolve (monotonically) without requiring changes to the former[3]. Furthermore, the stability of the AML also facilitates the development of a suite of related applications clustered around the AML hub.

3The current released version of Araucaria is still loosely coupled to the AML definition as it carries out AML parsing statically. The code is currently being migrated to exploit a SAX parsing module which dynamically tailors the parsing process to the specified DTD.

5.The Broader Context

The argument markup language AML has been designed to form a flexible data representation scheme that can be employed in a number of applications. In this section, the roles and relationships between these applications is sketched.

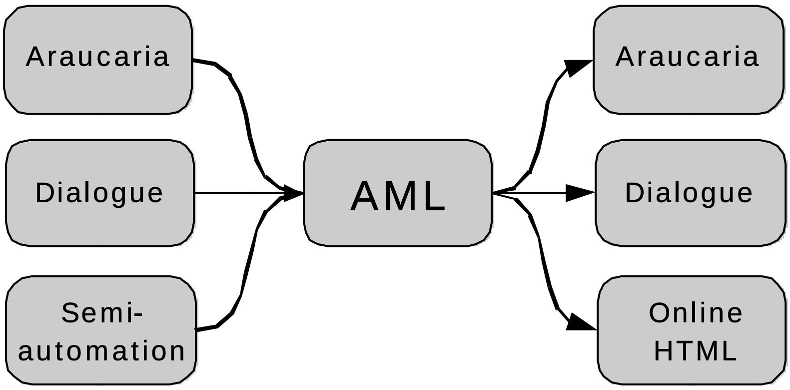

Figure 2. AML at the hub of a suite of modular applications

AML forms the hub, acting as a common data representation format that each application can write to, or read from: applications on the left are involved in the creation of AML content; those on the right in the manipulation and display of existing AML material.

Creating AML content

Currently, the primary means of creating AML content is Araucaria: that is, providing a simple application to support the manual markup of text. Though perhaps the most obvious approach, a range of alternative methods of supporting content creation are also under investigation.

One ambitious means of creating AML material is to automate the process of marking up. Work on this task is currently at an early stage, primarily because the broader research community has only recently started to investigate systematically the relationship between a variety of cues (such as discourse markers and punctuation) and argument structure (Snoeck Henkemanns, 1997). The complexity of the task is also compounded by the problems posed by enthymemes: automatically reconstructing implicit premises is a task which skilled analysts find both taxing and hard to explain. Automation of this task thus represents a significant challenge. A more reasonable short-term goal is to link the human analyst and computer tools, such that the mistakes of one might be flagged by the processing of the other, with their respective skills working in a complementary fashion.

A second project is investigating the role of dialogue in creating argumentation structures. In building applications which rely upon domain knowledge (expert systems, decision support systems, etc.), one of the most challenging tasks is knowledge elicitation: acquiring appropriate information from experts. Recent work has suggested that dialogue logics can be employed as a means of structuring the problem. Freeman (1991) proposes a dialogic interpretation of argumentation structures that is essentially a simple, stylised dialogue logic. He suggests that it is possible to view the move in a monologue from claim to data, or from conclusion to premise, by conceiving of an imaginary interlocutor who asks the question, 'Why?' - the ground adequacy question. Moving from one premise to the next in a convergent arrangement (i.e. with independent support) is the result of the implicit question, 'Can you give me another reason?', and moving from one premise to the next in a linked arrangement, the result of 'Why is that relevant?'. The 'dialogue logic' that results is thus rather impoverished by comparison to, say, Mackenzie's (1979) DC or Walton and Krabbe's (1995) PPD and RPD, but nonetheless it shares the core features of sets of rules specifying what can be said when, and what the interlocutors are rationally committed to. Of course, like RPD *inter alia*, the game is very much one-sided, with one player naively probing with one of three questions, and the other player providing all the information. The prototype system developed implements the dialogue logic by having the machine take on the role of question-asking player, leaving the user to respond appropriately. Thus to any claim that the user enters, the machine will probe for supporting premises and claims of relevance for those premises; if the user has no premise available she can enter a blank and the machine will halt probing in that particular direction. The following excerpt (Ex2) demonstrates a typical exchange:

(Ex2)	User (A):	Course ac2201 is interesting
	System:	Why is it that course ac2201 is interesting?
	User (B):	ac2201 involves studying computer hardware
	System:	Why is "ac2201 involves studying computer hardware" relevant to "Course ac2201 is interesting"?

User (C):	Looking at hardware means taking a hammer to a computer
System:	Why is it that looking at hardware means taking a hammer to a computer?
User:	-
System:	Why is it that ac2201 involves studying computer hardware?
User (D):	Students leaving second year must have a thorough understanding of the fundamentals of computing
	...
System:	Can you give me another reason for why course ac2201 is interesting?
User (E):	ac2201 teaches students how to use Linux
System	...

By carrying out the exchange in (Ex2), the system builds up an argument diagram (that is, an AML description of an argument) shown in Figure 3:

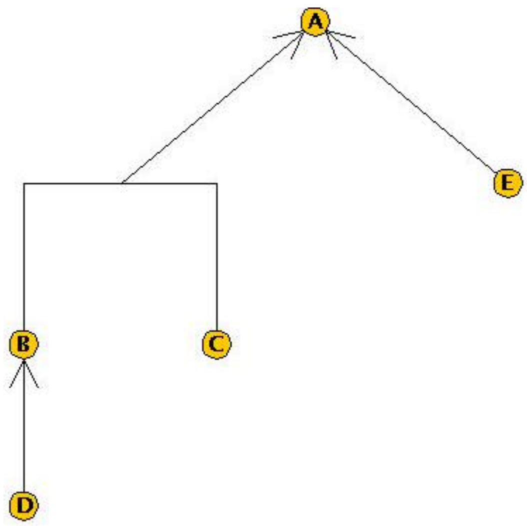

Figure 3. Argument diagram for *ac2201* argument

Many issues of discourse coherence, structure navigation, and linguistic manipulation remain to be addressed, but the prototype clearly demonstrates that an argumentation structure can be built up through a reasonably natural dialogue. The key benefit of the AML approach is that the exercise is not simply an end in itself. The argument created by this dialogue (perhaps an exchange with a member of faculty) could then be employed by a separate application to offer advice to students as to which courses to sign up for.

Manipulating AML content

Currently the primary means of viewing and manipulating AML material is once again Araucaria. There are, however, two distinct ways in which Araucaria can be employed in this regard. The first is as described above, with an interactive application which loads an AML file and allows the user to view and modify the argumentation structure. In addition, Araucaria can be employed in a stripped down version simply to load and view an argument diagram. More than just enforcing 'read-only' access, this method is useful in providing an online applet: work is underway to provide a database of AML files which can be accessed over the WWW, and displayed using this cut down applet version of Araucaria. This forms an implemented platform upon which to develop corpus resources, as described in the next section.

Again, there are also several other applications providing alternative means of access to AML content. The first exploits the capabilities of stylesheets in dynamically producing HTML content from AML data. Work is under way to produce a library of such stylesheets, supporting a range of different views of arguments. Some of these views act as summaries (showing only part of a whole argument), and others as typographical improvements (indenting and formatting to highlight various structural aspects). Yet others in this library perform more sophisticated transformations, producing versions of source arguments in which individual premises and conclusions appear as hyperlinks, from which supporting data can be linked, thus providing a means of navigating a large or complex argument. In this way a single component of this stylesheet library is currently capable of producing HTML output which attempts to mimic the format (though certainly not the content) of Scaltsas' extensive online *Archelogos* project (Scaltsas, 1997).

Finally, dialogue can also be exploited as a means of content provision, albeit in a restricted form. In much the same way as data can be acquired by the machine through implementation of a dialogue logic with the machine as the questioner, so too can data be provided by having the user take on the role of questioner. Clearly, to conform to a specific dialogue logic, it is necessary to restrict the user's input, so that the bounds on questioning are clearly identified. Prototype work in implementing Walton and Krabbe's RPD has demonstrated that supporting user participation in tightly constrained dialogue logics is both feasible and an acceptable model for interaction, and work is now under way to provide a user with the ability to probe an existing AML structure with simple questions following the pattern of Ex2.

6.Application Areas

There is a very wide range of potential applications of the various components described in the previous section; in this section, the focus is squarely upon the current and potential applications of the currently implemented system functionality.

The foremost application domain is pedagogy. Teaching critical thinking skills, particularly in North America, forms an important part of the curriculum in providing generic, transferable skills. Syllabi for the topic, such as those provided in popular text books such as (Johnson & Blair, 1993; Govier, 1997; Groarke *et al.*, 1997) typically introduce some method for diagramming arguments fairly early on, to provide students with the practical scaffolding around which to erect a battery of analytic techniques. Though the various techniques may differ somewhat, and the presentation of them differ significantly in these works, the diagramming tools are substantially the same. Thus the diagramming itself is uncoupled from the subsequent presentation of critical thinking skills and techniques, which suggests that a ‘theory-neutral’ software tool such as Araucaria might be successfully employed as a component of teaching support in a broad range of argumentation and critical thinking courses. The current version of the software is being trialled on undergraduate courses at several Canadian and US universities in the fall of 2001.

One problem that Araucaria does not yet address is that of computer based assessment. In some situations, the burden of marking on academic staff can be drastically reduced through the use of automated marking software. Typically, such software relies upon a very narrowly bounded range of possible answers, and although some of the problems associated with traditional “multiple choice” papers can be avoided (Scriven, 2001), such assessment techniques nevertheless severely restrict the freedom of the student. The approach offered by

Araucaria, in contrast, offers the possibility for automatic marking of students' unfettered argument analysis by exploiting techniques for graph matching (Bunke & Messmer, 1997). This avenue is left to future work, for it involves one key challenge: that the comparison between student and model answers should be sufficiently flexible to handle the wide variation in argument structures which might be considered 'correct' or 'nearly correct'.

With domain information structured as arguments in AML, there is also a rich potential for supporting the teaching of other topics. Thus, for example, a small corpus of AML arguments, perhaps constructed using Araucaria, could capture the material for part of an introductory paleontology course, giving arguments for and against conclusions to be drawn from various aspects of the fossil record[4]. Student interaction with this resource could then form part of a computer assisted learning environment such as those described in (Jackson, 1998). Employing applications such as those described in section 5, students could review arguments and summaries of them, engage in dialogic exchanges, extend the existing arguments with their own additions, and so on.

In a similar vein, such applications could also have a role to play outside the classroom, in providing one resource for topics in the Public Understanding of Science and Technology (PUST). With online provision of the same tools (Araucaria, dialogic interaction, dynamic generation of summaries, etc.) and a set of AML resources in topical areas such as the genetic modification of food, conflicting viewpoints could be presented to the public in a coherent and measured way. It would be simple to provide for public interaction with the arguments, supporting the contribution of new arguments to the online database. Using arguments to structure online debate has been found to be a good means of involving people in public policy decision making processes (Gordon & Karacapilidis, 1997), and it might be expected that similar advantages might accrue in PUST.

Araucaria and the underlying data representation format also has the potential to serve the academic community in several respects. First, from a practical point of view, the output of Araucaria, both graphical and textual, simplifies the task of preparing material for dissemination. Secondly, and equally practically, having a common format in which to

4 Paleontology as a discipline offers particularly rich examples of texts in which dialectical structure and chains of argumentation are extremely clear. It is for this reason that many introductory texts employ a chronological basis for exposition, following the various turns of the academic dialogue. See, e.g. (Edey & Johanson, 1990) and chapters 15-16 of (Wilford, 1985)

express the structure of an argument simplifies the task of exchanging and dissecting analyses. (Argthry, 2001) has many good examples of academic discussion on structural analyses of problematic examples; each suggestion and counter-suggestion is phrased in idiosyncratic and lengthy analyses which increase the chances of misunderstanding and error. A common language may not make the analyses themselves any easier, but it will at least make the subsequent exchange and discussion of those analyses more open, and less prone to confusion.

Much more substantial than these, however, is the provision of a corpus of argumentation, analysed and marked up in AML, and made available online. Work is starting at Dundee to construct such a corpus, the contents of which will be accessible from the WWW, with Aracuaria as an applet for viewing individual arguments. Because the data is stored in a highly structured form using AML, sophisticated access and manipulation becomes possible: a visitor might search for arguments with a particular structure, or for examples of a particular argumentation scheme, or for arguments in a particular domain, or for arguments with a particular degree of complexity, and so on. Furthermore, if Araucaria is used by both academics and students in argumentation to mark up new examples of arguments, then these analyses can be submitted to the corpus to extend the resource for others.

7.Conclusions

The Araucaria system performs a range of functions which are unique, and the software has the potential to play a significant role in both academic and educational domains. Perhaps the most similar software is van Gelder's (2001) Reason!Able system. Like Reason!Able, Araucaria employs a tree structure for mapping out the relationships between components in an argument, and allows the user to manipulate that structure. Unlike Reason!Able, however, Araucaria is driven primarily by research concerns rather than educational concerns, and although pedagogy is a significant application area for Araucaria, it is not the only such area. As a result, the current version utilises, for example, recent research in argumentation schemes to provide support for analyses based upon such schemes. In addition, Araucaria also starts with the assumption that the task at hand is one of analysis of existing argument, rather than the construction of a new argument; for Reason!Able the focus is squarely upon argument synthesis. Finally, Araucaria differs fundamentally from Reason!Able and all other argumentation software in its provision of AML, an open standard for argument description defined in XML, which has the potential to have a pervasive effect in both teaching and research.

One feature that is offered by Reason!Able but is not currently available in Araucaria is the notion of rebuttal, namely, that a particular premise might function in detracting from, rather than supporting, a conclusion[5]. Future work aims to address this issue in a principled way in two steps: first by extending AML, and then by developing a version of Araucaria which exploits the AML extensions. The fundamental restriction that is currently imposed on Araucaria is that the design of AML assumes that an argument is monologic, such that no alternative or dissensual views can be represented. The first step, then, is to extend AML (in such a way that it is backwardly compatible with the current version, as described in section 4 above), so that it can represent multiple points of view, thereby characterising dialogic argument. The second step is to determine how best dialogic argument can be represented graphically by a tool such as Araucaria. Both tasks are currently under investigation.

Though many of the applications described in section 6 are currently under development, they are described here to show the role that the implemented Araucaria system and its underlying data representation format, AML, will play in a variety of domains. Though Araucaria on its own represents a significant tool for those working in argumentation, when coupled with the applications in the domains suggested, it has the potential not only to play a key role in the development of a range of systems of real utility in academic, pedagogical, and public arenas, but also to support and encourage the further development of aspects of argumentation theory and the application of that theory in computer systems.

Araucaria is free, open-source software, released under the GNU General Public License. The software can be downloaded from the project homepage,
`http://www.computing.dundee.ac.uk/staff/creed/araucaria/`

Alternatively, a distribution CD can be obtained by sending an email to
`araucaria@computing.dundee.ac.uk`

5In fact, it would be possible to characterise rebuttal as a particular argumentation scheme, but this is a rather unprincipled solution which would be very cumbersome to extend to truly dialogic

Acknowledgements

The authors would like to clearly acknowledge the work of three of Chris Reed's students, Simon Wells, Yang Li and Kaifeng Li. Simon worked on a final year undergraduate project in 2000/01 focused upon the implementation of RPD, and Yang and Kaifeng worked during 2001 on M.Sc. projects implementing the prototype application for dialogic creation of AML material, and a number of XSL stylesheets for the dynamic generation of HTML content from AML, respectively. It is their work to which reference is made in section 5.

Grateful acknowledgement is also made of comments and feedback on this work offered by Leo Groarke, David Hitchcock, and Doug Walton. Any errors that may remain, however, are of course entirely the responsibility of the authors.

References

Argthry (2001) Discussions on the ARGTHRY list, particularly the archives for July 2001 (available through the process described at `http://www.yorku.ca/gilbert/argthry/`)

Beardsley (1950) *Practical Logic*. Prentice Hall.

Bunke, H & Messmer, BT (1997) "Recent advances in graph matching", *International Journal of Pattern Recognition and Artificial Intelligence*, **11** (1): 169-203.

Edey, M.A. & D.C. Johanson (1990) *Blueprints: Solving the Mystery of Evolution*, Oxford University Press.

Freeman, J.B. (1991) *Dialectics and the MarcoStructure of Arguments*, Foris.

Gordon, T.F. & Karacapilidis, N. (1997) “The Zeno Argumentation Framework” in *Proceedings of the 6th International Conference on AI and Law (ICAIL’97).*

Govier, T. (1997) *A Practical Study of Argument*, Wadsworth.

Grennan, W. (1996) *Informal Logic: Issues and Techniques*, McGill-Queens University Press.

Groarke, L., C.W. Tindale & L. Fisher (1997) *Good Reasoning Matters*, 2nd Edition, Oxford University Press.

Ide, N. (2000) “The XML Framework and Its Implications for Corpus Access and Use” in

arguments.

Proceedings of LREC'2000 MMLR Workshop, Athens.

Jackson, S. (1998) "Beyond Web Course Automation: Designing Online Dialogue" in *Working Notes of NAU/Web'98*

Johnson, R. & J.A. Blair (1993) *Logical Self Defense*, McGraw Hill.

Kienpointner, M. (1992) "How to classify arguments" in F.H.van Eemeren, R. Grootendorst, J.A. Blair & C.A. Willard, *Argumentation Illuminated*.

LeBlanc, J. (1998) *Thinking Clearly*, W.W.Norton & Co.

Mackenzie, J.D. (1979) "Question Begging in Non-Cumulative Systems" *Journal of Philosophical Logic* **8**:117-133.

Mann, W.C. & S.A. Thompson (1988) "Rhetorical Structure Theory: A Theory of Text Organisation" *Text* **8** (3) 243-281.

O'Donnell, M. (1997) "RST-Tool: AN RST Analysis Tool", in W. Hoeppner (ed.) *Proceedings of the 6th European Workshop on Natural Language Generation (EWNLG'97)*, Duisburg.

Perelman, Ch. and L. Olbrechts-Tyteca (1969) *The New Rhetoric*, Notre Dame Press.

Reed, C.A. (1997) "Representing and Applying Knowledge for Argumentation in a Social Context" *AI & Society* **11**:138-154.

Reed, C.A. (1999) "Building Monologue" in *Proceedings of the 3rd OSSA Conference (OSSA'99)*, St. Catharines.

Reed, C.A. (1999b) "The Role of Saliency in Generating Natural Language Arguments", in *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, Morgan Kaufmann, pp876-881.

Reed, C.A. (*to appear*) "Reconciling argumentation theoretic and discourse relation based approaches to natural language argument"

Reed, C.A., Norman, T.J. & Gabbay, D. (2002, to appear) *Argument and Computation*

Reed, C.A. & D.P. Long (1998) "Generating the Structure of Argument" in *Proceedings of the 36th ACL and 17th COLING (ACL-COLING'98)*, pp1091-1097.

Reed, C.A. & Walton, D.N. (2001) "Applications of Argumentation Schemes" in *Proceedings of the 4th OSSA Conference (OSSA'2001)*, Windsor.

Rips, L.J. (1994) *The Psychology of Proof*, MIT Press

Sadock (1977) "Modus Brevis: The Truncated Argument" in *Papers from the 13th Regional Meeting of the Chicago Linguistics Society* pp545-554

Scaltsas, T. (1997) "Representation of Philosophical Argument" in T. Bynum and J. Moor (eds) *The Digital Phoenix: How Computers are Changing Philosophy*, Blackwell.

Scriven, M. (2001) "Testing Critical Thinking" in *Proceedings of the 4th OSSA Conference (OSSA'2001)*, Windsor.

Shoesmith, D.J. & T.J. Smiley (1978) *Multiple Conclusion Logic*, Cambridge University Press.

Snoeck Henkemanns (1997) "Verbal Indicators of Argument and Explanation" in *Proceedings of the 2nd OSSA Conference*, St. Catharines.

Thomas, S.N. (1986) *Practical Reasoning in Natural Language*, 3rd Edition, Prentice Hall.

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

van Eemeren, F.H., R. Grootendorst & F. Snoeck Henkemanns (1993) *Reconstructing Argumentative Discourse*, LEA.

van Gelder & A. Rizzo (2001) "Reason!Able Across the Curriculum" in *Is IT an Odyssey in Learning? Proceedings of the 2001 Conference of the Computing in Education Group of Victoria.*

Walton, D.N. (1996) *Argumentation Schemes for Presumptive Reasoning*, LEA.

Walton, D.N. & E.C.W. Krabbe (1995) *Commitment in Dialogue*, SUNY Press.

Wilford, J.N. (1985) *The Riddle of the Dinosaur*, Faber and Faber.

Wilson (1986) *The Anatomy of Argument*, University Press of America.