Transforming Natural Arguments in Araucaria to Formal Arguments in LMA

Yohsuke Takahashi
Graduate School of Science and Technology
Niigata University
8050, 2-cho, Ikarashi, Niigata, JAPAN
yohsuke@cs.ie.niigata-u.ac.jp

Hajime Sawamura
Institute of Natural Science and Technology
Academic Assembly, Niigata University
8050, 2-cho, Ikarashi, Niigata, JAPAN
sawamura@ie.niigata-u.ac.jp

Jing Zhang*
School of Computer and Control Engineering, Qiqihaer University, CHINA
zhangjing1968_6@sina.com

Abstract

Over the last few years, argumentation has been gaining increasing importance in multi-agent systems research. LMA is a Logic of Multiple-valued Argumentation built on top of EALP (Extended Annotated Logic Programming). Araucaria is a software tool for analyzing and diagramming arguments written in natural language. We integrate these two systems involved in argumentation to an easy to use and more user-friendly argumentation system, in which natural arguments can be transformed into formal arguments, in order to lighten users' knowledge burden forced by formal argumentation frameworks, and attain extensive usability of formal argumentation systems. We illustrate the transformation process by translating natural arguments in English to formal arguments in EALP step-by-step, and then how the status of those formal arguments can be determined in LMA, resulting in the status determination of original natural arguments.

1. Introduction

We argue all the time individually and socially in every situation such as in daily, scientific, commercial or political fields. Argumentation is by nature a logic of reasoning under uncertain or changing information environment. Over the last few years, argumentation has been gaining increasing importance in multi-agent systems research, which is about to give a great influence not only on a new style of software development but also on a future figure of our computer-networked society. In the future, we could expect that computerized agents could help or replace humans in the fields of public policy-making, e-pedagogy, various types of conferences, even the Diet/Parliament as an advanced function of e-democracy or e-government. Prior to, or towards such a dream, we have to prepare a lot of things, of course. Among other things, many efforts have already been devoted to computer-supported argument visualization (CSAV) [3].

So far, many argumentation frameworks have been proposed from various perspectives and purposes [1][5]. However, the underlying knowledge representation language and argumentation formalism are so formal and restrictive that it lays a heavy burden of formal knowledge representation on users. It goes without saying that natural language and natural argument are far preferable in place of formal ones.

Araucaria is a software tool for analyzing and diagramming arguments written in natural language (English) [6]. This is a human-oriented system for those purposes and hence is not intended to be an automated argument system for agents. It, instead, has many unique features for analyzing and diagramming natural and informal arguments in the real world. It has been developed aiming at being used:

- in preparing teaching materials in critical thinking, informal logic and argumentation theory
- in the classroom, either for instructor or student use
- for preparing online resources
- for working with argumentation schemes
- in designing examples for academic work
- in exchanging examples and problems in a common, open format (AML)
- for the use and sharing of material between individuals and sites

*The author is now a Visiting Professor of Graduate School of Science and Technology, Niigata University.

Proceedings of the IEEE/WIC/ACM International
Conference on Intelligent Agent Technology (IAT'06)
0-7695-2748-5/06 $20.00 © 2006 IEEE

- for building and accessing a large, online repository of argumentation (AraucariaDB).

LMA is a Logic of Multiple-valued Argumentation built on top of EALP (Extended Annotated Logic Programming). It is a very formal argumentation framework developed by us [9]. LMA takes into account two kinds of intrinsic uncertainty: (1) agents can not have a perfect knowledge/belief base for the universe of discourse in resolving goals and making decisions, and (2) propositions in a knowledge/belief base can hold only partial or vague information. EALP and LMA have been devised to satisfy these two aspects of uncertainty recognition simultaneously, and differently from other approaches to argumentation theory and practice [1][5]. It is also a very general logic of multiple-valued argumentation in the sense that it allows us to specify various types of truth values depending on application domains and to deal with uncertain arguments under the specified truth values. Furthermore, LMA is notable for allowing us to deal with culturally unique arguments, that is, not only logos-oriented arguments in the West, but also tetralemma-oriented ones in the East [8].

In this paper, with these in the background, we address ourselves to realizing the following scenario, complimentarily taking into account both formal advantages of LMA and natural advantages of Araucaria.

1. Analyze and diagrammatize natural arguments with Araucaria.
2. Extract knowledge (rules and facts) from those diagrammatized arguments, and construct formal arguments in EALP automatically for each agents participating in argumentation.
3. Argue about an issue and decide the argument status with the dialectical proof theory of LMA.

The paper is organized as follows. In the succeeding two sections, we outline Araucaria, and EALP and LMA for preliminaries. In Section 4, we describe the transformation method of natural arguments analyzed in Araucaria to formal arguments in EALP, along with a moderately complicated but convincing argument example. Section 5 includes concluding remarks and future work.

2. Outline of Araucaria

Araucaria has many unique features for analyzing and diagramming natural and informal arguments in the real world [6]. Here we describe only the features that are most useful and essential in transforming natural arguments to formal ones in EALP. Conceptually, the both systems share some important notions or components but in a different manner. They are:

- Uncertainty
- Argument evaluation
- Ownership of arguments
- Argument schemes
- Enthymemes.

Araucaria and LMA see ‘uncertainty’ in knowledge and reasoning from scratch, and therefore take into account it in their design philosophy.

In evaluating arguments, Araucaria allows to indicate to an argument such a linguistic expression ‘good’ or ‘bad’, ‘strong’ or ‘weak’ as can be seen in fuzzy logic, or to indicate to the claims or the relationships between claims real values between 0 and 1 corresponding to probabilities, or ‘+’, ‘?’, ‘-’ in three-valued logic, and so on. The diversity of truth values in LMA [7] can easily admit those varieties of truth values in Araucaria. But it should be noted that they are supposed to be dealt with only by embedding them to a certain complete lattice of truth values.

Araucaria allows premises, conclusions and arguments to have their ownership that represents which speaker is associated with them. This apparatus is convenient to LMA since in argument-based agent systems, each agent has its own knowledge base separately, and puts forward its arguments and counter-arguments in the argumentative dialogue.

Araucaria has, as its unique and powerful apparatus, argumentation schemes which are forms of arguments (structure of inference) representing common types of argumentation (such as stereotypical non-deductive, non-inductive or abductive or presumptive patterns of reasoning). In EALP and LMA, they turn out to be transformed into a form of logic programming, EALP whatever they are.

In natural arguments, premises (major or minor) are left implicit or tacit (such arguments usually called enthymemes (truncated syllogism)). Araucaria allows for such an incomplete reasoning and simultaneously has a function complementing missing premises. Once they are supplied to make a complete argument, it can be easily transformed to one in LMA.

To say one, the feature unique only to Araucaria is AML (Argument Markup Language). This is to distribute, share and reuse DB of arguments. But LMA has no counterpart for it at the moment.

These many relevancy based on the similarities and idiosyncrasies between Araucaria and LMA allow for the transformation from Araucaria to EALP/LMA described below.

Proceedings of the IEEE/WIC/ACM International
Conference on Intelligent Agent Technology (IAT'06)
0-7695-2748-5/06 $20.00 © 2006 IEEE

3. Outline of EALP and LMA

EALP (Extended Annotated Logic Programming) is an expressive logic programming language we extended for argumentation by incorporating default negation in Generalized Annotated Logic Programming by Kifer and Subrahmanian [2]. EALP has two kinds of explicit negation: Epistemic Explicit Negation '¬' and Ontological Explicit Negation '∼', and the default negation '**not**'. They are supposed to yield a momentum or driving force for argumentation or dialogue in LMA below. The basic language constituents are literals with truth-values or epistemic states of agents explicitly annotated. The structure of truth-values is required to be a complete lattice so that the paraconsistency of an agent's knowledge base is guaranteed under the ideals-based semantics [9].

LMA is A Logic of Multiple-valued Argumentation constructed on top of EALP. LMA allows agents to construct arguments under uncertain knowledge in EALP and to argue with other agents on uncertain issues in the open networked heterogeneous environment.

The most primary concern of this logic is the rebuttal relation among arguments, which is to be induced by those notions of negation. The rebuttal relation for two-valued argument models is most simple, so that it naturally appears between the contradictory propositions of the form A and $\neg A$. In case of multiple-valued argumentation based on EALP, much complication gets involved into the rebuttal relation to be defined with the concepts of negation. One of the questions arising from multiple-valuedness is, for example, how a literal with truth-value ρ confronts with a literal with truth-value μ in the involvement with negation. LMA has a reasonable answer to it in its argumentation framework with an argumentation semantics proper to EALP-based argumentation under uncertainty (see [9] for the details). We have soundness and completeness for LMA under the fixpoint semantics based on the acceptability of arguments and the dialectical proof theory [9].

As we can specify truth values every application domain that has its own proper uncertainty in EALP, such diversity of truth values brings us an extensive applicability of LMA. In what follows, we will illustrate how uncertain arguments proceed in LMA using a simple argument with a somewhat deviant use of truth values, without involving in lengthy definitions.

Example1(Job schedule management) Let us consider an argumentation about the monthly job schedule management. Here we use an unconventional complete lattice of truth values which is the power set $\mathcal{P}(\{1,\ldots,31\})$ of the set of the monthly dates, with the order of set inclusion. Then an annotated atom $work(a) : \{5,6\}$, for example, reads "Agent a works on the 5th and the 6th". It asserts that the proposition $work(a)$ is true only in a certain time interval. $\sim work(a) : \{5,6\}$ reads "Agent a does not work on the 5th and the 6th". We define the epistemic explicit negation so as to be $\neg A : \mu = A : \neg\mu$ and $\neg\mu = \{1,\ldots,31\} - \mu$ as like in GAP [2], and thus $\neg work(a) : \{5,6\}$ reads "Agent a works on every day except the 5th and the 6th". The difference and significance between the ontological and epistemic explicit negations is obvious. Under this complete lattice of truth values, we consider $MAS = \{KB_m, KB_a, KB_b, KB_o\}$, where the knowledge base KB of each agent is, in EALP,

$$
\begin{array}{l}
KB_m = \{ \\
\quad finish(project) : \{6\} \leftarrow work(a) : \{3,4,5\}\ \& \\
\quad\quad arrive(component) : \{5\}, \\
\quad work(a) : \{3,4,5\} \leftarrow, \\
\quad arrive(component) : \{5\} \leftarrow, \\
\quad pay(upcharge) : \{8\} \leftarrow \qquad\qquad \}, \\
KB_a = \{ \\
\quad \sim work(a) : \{5\} \leftarrow \textbf{not}\ work(b) : \{5\}\ \& \\
\quad\quad holiday : \{5\}, \\
\quad \sim work(a) : \{12\} \leftarrow \textbf{not}\ work(b) : \{12\}\ \& \\
\quad\quad holiday : \{12\}, \\
\quad holiday : \{5,6,12,13\} \leftarrow \qquad\qquad \}, \\
KB_b = \{ \\
\quad \neg work(b) : \{12,19,26\} \leftarrow, \\
\quad holiday : \{5,6,12,13\} \leftarrow \qquad\qquad \}, \\
KB_o = \{ \\
\quad \sim arrive(component) : \{5\} \leftarrow \\
\quad\quad \textbf{not}\ pay(upcharge) : \phi \qquad\qquad \}.
\end{array}
$$

KB_m, KB_a, KB_b and KB_o stand for knowledge bases of a manager agent m, employee agents a, b and a subcontractor agent o respectively. Agent m's argument which has the conclusion $finish(project) : \{6\}$ (the project should finish on the 6th) is justified by the dialectical proof theory as shown in Figure 1.

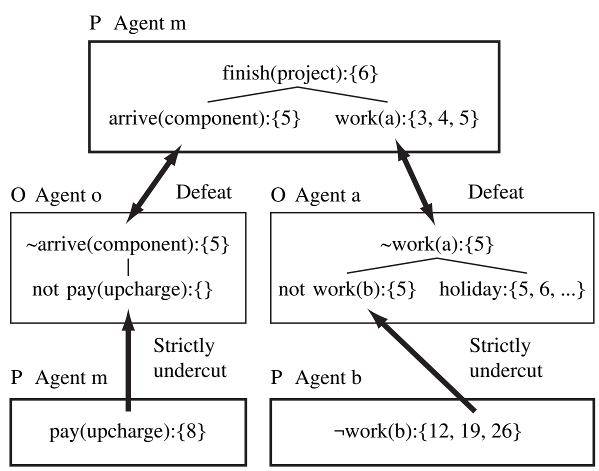

Figure 1. The winning dialogue tree in Example1.

Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'06)
0-7695-2748-5/06 $20.00 © 2006 IEEE

In the winning dialogue tree, initially Agent m (P: Proponent) says "If a component will arrive on the 5th, and Agent a works on the 3th, the 4th and the 5th, then the project will finish on the 6th", whose formal argument is constructed from its knowledge base KB_m as

$[finish(project) : \{6\} \leftarrow work(a) : \{3, 4, 5\}$ & $arrive(component) : \{5\},$
$work(a) : \{3, 4, 5\} \leftarrow,$
$arrive(component) : \{5\} \leftarrow$].

Then there are two places that can be attacked by the other party (O: opponent). In the left branch of the dialogue tree, Agent o defeats it as follows "I will be not able to bring a component on the 5th if the additional charge is not paid". But Agent m strictly undercuts o's argument by saying "I will pay it to you on the 8th". In the right branch of the dialogue tree, for the first argument of Agent m, Agent a (O: Opponent) also defeats by saying "The 5th is a holiday, and if the coworker b does not work, I do not want to work on the 5th", whose formal argument is constructed from its knowledge base KB_a as

$[\sim work(a) : \{5\} \leftarrow \textbf{not}\ work(b) : \{5\}$ & $holiday : \{5\},$
$holiday : \{5, 6, 12, 13\} \leftarrow$].

This is a semantically correct argument since $holiday : \{5\}$ can be resolved upon $holiday : \{5, 6, 12, 13\}$ with the condition $\{5\} \leq \{5, 6, 12, 13\}$ in GAP and EALP. Agent a can put forward such a counter-argument since the conclusion of Agent a's argument $\sim work(a) : \{5\}$ conflicts with the second rule of Agent m, $work(a) : \{3, 4, 5\} \leftarrow$. This is due to the defeat (rebut) relation that $A : \mu$ conflicts with $A : \rho$ each other provided that $\mu \geq \rho$ or $\rho \geq \mu$ in LMA. In fact, Agent a claims that it does not want to work on the 5th, but Agent m asserts that it works on $\{3, 4, 5\}$ which is a superset of $\{5\}$.

However Agent b (P: Proponent) strictly undercuts this Agent a's argument by saying "I will work on days except the 12th, 19th and the 26th", whose formal argument is constructed from its knowledge base KB_b as

$[\neg work(b) : \{12, 19, 26\} \leftarrow]$.

This is equivalent to $\neg work(b) : D \leftarrow$, where $D = \{1, \ldots, 31\} - \{12, 19, 26\}$ and hence can undercut the first rule of Agent a's counter-argument above. This is due to the strict undercut relation that $A : \mu$ can attack **not** $A : \rho$ in one way provided that $\mu \geq \rho$ in LMA. In fact, Agent b claims that it works on the dates D including $\{5\}$, but Agent a asserts that it does not work on the date 5th.

There is not any further arguments at this stage of the argumentative dialogue, and it finishes at the proponent's move. Consequently, the first argument of Agent m becomes justified.

The dialectical proof theory makes P and O put forward arguments and counter-arguments alternatively in this manner. In the dialectical tree, the status of an argument is defined to be justified if every possible dialogue sequence (branch) ends at P' move, overruled if the argument is defeated by a justified argument, and defensible if it is neither justified nor overruled.

4. Transforming Argument Diagrams in Araucaria to EALP/LMA

In this section, we describe the overall story of converting natural arguments in Araucaria to formal arguments in to EALP/LMA by using a moderately complicated but convincing argument example. It is an argument cited from [4] that Galileo refutes Aristotle's assertion that the heavier a body is, the faster it falls to the ground.

> Suppose that we have two bodies, a heavy one called H and a light one called L. Under Aristotle's assumption, H will fall faster than L. Now suppose that H and L are joined together. ...Now what happens? Well, L plus H is heavier than H so by the initial assumption it should fall faster than H alone. But in the joined body...L is lighter and will act as a 'brake' on H, and L plus H will fall slower than H alone. Hence it follows from the initial assumption that L plus H will fall both faster and slower than H alone. Since this is absurd, the initial assumption must be false.

The transformation process consists of the five steps.

STEP 1: Analyze and diagrammatize natural arguments The natural argument above is analyzed and diagrammed by humans, as shown in Figure 2 with the various drawing functions of Araucaria [6].

STEP 2: Complement missing premises (enthymeme) or hidden schemes Syllogisms, premises or schemes as inference patterns are left implicit (or truncated) in natural arguments very often. In this step, those missing ones (called enthymeme in Aristotle's rhetoric) or schemes are complemented by humans. Figure 3 depicts a logically completed argument complemented with missing premises that are enclosed by the dotted lines.

STEP 3: Define a complete lattice of truth values It is necessary to specify a complete lattice as truth values in order to evaluate propositions and construct an EALP knowledge base. We can use the predefined complete lattice as truth values or define it in terms of Prolog. For this argument example, we employ the well-known complete lattice $\mathcal{FOUR} = (\{\bot, t, f, \top\}, \leq)$, $\forall x, y \in \{\bot, t, f, \top\}[x \leq y \Leftrightarrow x = y \vee x = \bot \vee y = \top]$ as shown in Figure 4.

STEP 4: Extract predicates and annotate them with the specified truth values From the argument diagrams in STEP 2, we extract predicates associated with annotations

Proceedings of the IEEE/WIC/ACM International
Conference on Intelligent Agent Technology (IAT'06)
0-7695-2748-5/06 $20.00 © 2006 IEEE

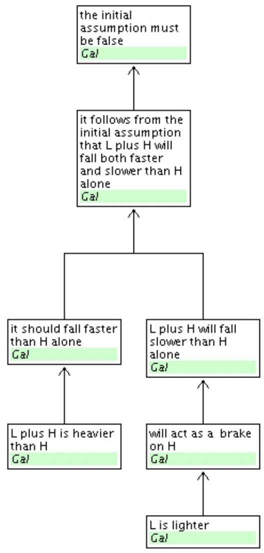

Figure 2. Galileo's original argument.

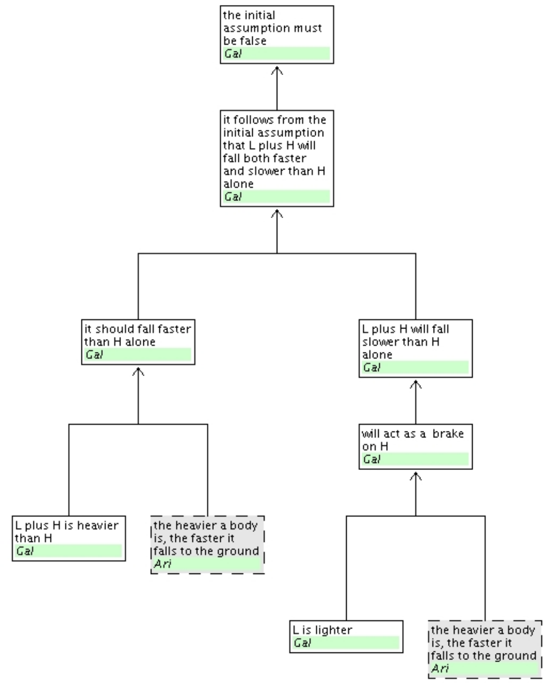

Figure 3. Galileo's complemented argument.

as the truth values specified in STEP 3. Figure 5 is a resultant diagram in which the annotations as truth values are represented in the bottom of the enclosed boxes.

STEP 5: Construct knowledge base in EALP Rules and facts from annotated literals in the diagrams constructed in STEP 4 are literally translated as follows:

sub-step 5-1: Construct literals From each vertex, extract a proposition symbol P and an annotation A. And transform them into a literal of the form of "$P : A$".

sub-step 5-2: Construct rules in EALP From each vertex representing a literal L and its children literals $L_1, \ldots, L_n$ (possibly $n = 0$), construct a rule of the form of "$L \leftarrow L_1 \& \ldots \& L_n$" (simply "$L \leftarrow$" if $n = 0$). The exemplary argument diagram produces the following EALP knowledge base KB_{Gal}.

$$
\begin{array}{l}
KB_{Gal} = \{ \\
\quad \sim aristotle\_assumption : t \leftarrow \\
\qquad fall\_faster(L+H, H) : \top, \\
\quad fall\_faster(L+H, H) : \top \leftarrow \\
\qquad fall\_faster(L+H, H) : t \, \& \\
\qquad fall\_faster(L+H, H) : f, \\
\quad fall\_faster(L+H, H) : t \leftarrow \\
\qquad heavier(L+H, H) : t \, \& \\
\qquad not\ aristotle\_assumption : f, \\
\quad heavier(L+H, H) : t \leftarrow, \\
\quad fall\_faster(L+H, H) : f \leftarrow brake(L, H) : t, \\
\quad brake(L, H) : t \leftarrow lighter(L, H) : t \, \& \\
\qquad not\ aristotle\_assumption : f, \\
\quad lighter(L, H) : t \leftarrow \qquad \}.
\end{array}
$$

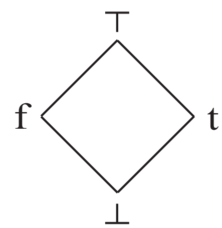

Figure 4. The complete lattice of truth values, $\mathcal{FOUR}$.

In the same manner, the arguments by Aristotle is depicted as in Figure 6, where he asserts his belief from his empirical observation. Here we make two more agent appear on the stage, who are a modern scientist having a firm belief on verificationism and asserting that "He does not admit empirical facts because they have not been scientifically verified", and an agent with an Eastern mind, saying that Easterners prefer a more holistic or dialectical argument like this: "Aristotle is based on a belief that the physical object is free from any influences of other contextual factors, which is impossible in reality" [4]. The argument diagrams for a modern scientist and an agent with an Eastern mind are depicted in Figure 7, and Figure 8 respectively.

Those trees can be transformed to the following EALP knowledge bases, KB_{Ari} of Aristotle, KB_{Mod} of a modern scientist, and KB_{Eas} of an Eastern agent.

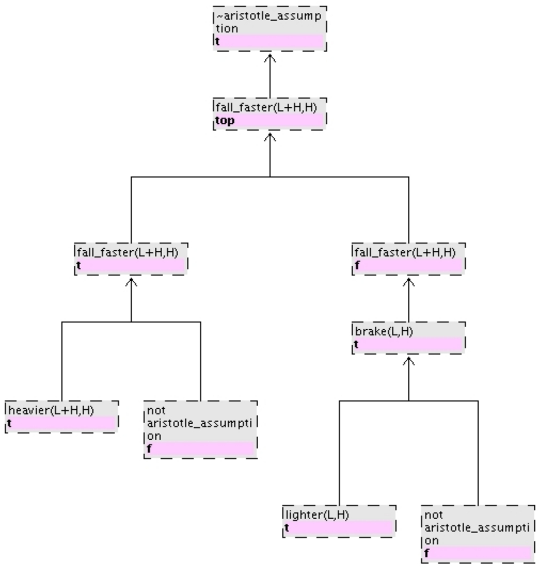

Figure 5. Predicates extraction.

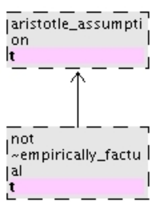

Figure 6. Argument by Aristotle.

$$
\begin{aligned}
&KB_{Ari} = \{\\
&\quad aristotle\_assumption : t \leftarrow\\
&\qquad not \sim empirically\_factual : t \quad \},\\[1ex]
&KB_{Mod} = \{\\
&\quad \sim empirically\_factual : t \leftarrow\\
&\qquad notscientifically\_verified : t \quad \},\\[1ex]
&KB_{Eas} = \{\\
&\quad \sim aristotle\_assumption : t \leftarrow\\
&\qquad distrust\_decontextualization : t,\\
&\quad distrust\_decontextualization : t \leftarrow \}.
\end{aligned}
$$

STEP 6: Argue about issues With the preparations done so far, agents can get started on arguing about the issues in LMA. For the knowledge bases, KB_{Gal}, KB_{Ari}, KB_{Mod}, and KB_{Eas}, the dialectical proof theory of LMA can produce a dialogue tree as shown in Figure 9. Wherein, Galileo's argument is justified in LMA. Also, it is obvious that Aristotle's argument is overruled, a modern scientist's ones is justified as well, and an Eastern agent's one is still

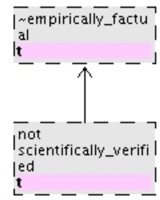

Figure 7. Argument by a modern scientist.

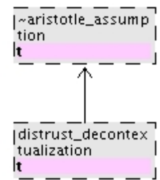

Figure 8. Argument by an agent with an Eastern mind.

defensible.

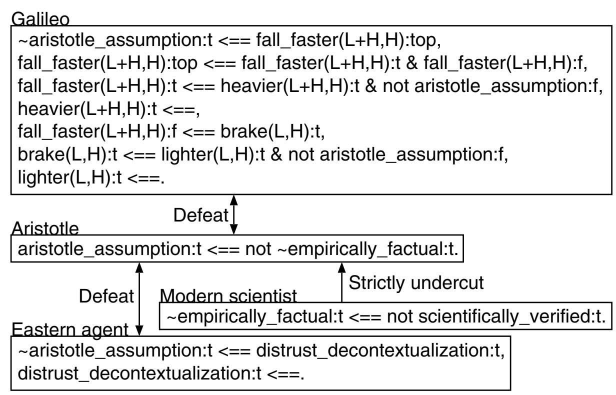

Figure 9. An argumentative dialogue tree in LMA.

It should be noted that STEP 1, 3 and 4 are not so easy for the complete automation since they are a very creative process in the transformation, STEP 2 may be partially automated, and STEP 5-6 are fully automated.

Example2 Let us consider another argument example cited from AraucariaDB[1], the online repository of the analysed arguments. The original source of the argument is The Japan Times[2].

> There have been no claims of responsibility for the Bali bombings. But some of the methods used

[1]http://araucaria.computing.dundee.ac.uk/search.php
[2]http://www.japantimes.co.jp/

Proceedings of the IEEE/WIC/ACM International
Conference on Intelligent Agent Technology (IAT'06)
0-7695-2748-5/06 $20.00 © 2006 IEEE

in the blasts indicate that they are the work of an international terrorist group, not just local Islamic radicals. For example, the bomb used in the night-club attack was reportedly made from a military plastic explosive similar to the one used in the attack on the USS Cole in Yemen two years ago.

Within this argument, “the bomb used ... two years ago” stands for the premise of “they are the ... local Islamic radicals”. Figure 10 depicts a diagram of this direct argument.

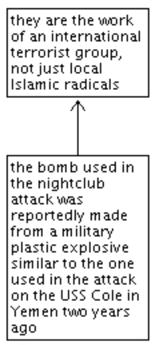

Figure 10. A direct diagram.

However, there are several propositions missing between the lines of the argument and being necessary to make the diagram logically more plausible. For example, “the bomb used ... two years ago” follows “the attack in Yemen and the attack in Bali were carried out by the same terrorist group” that in turn will follow “they are the ... local Islamic radicals”. Figure 11 shows a diagram with all the premises complemented, which seem to be necessary.

Here we see nonmonotonic reasoning appearing in this argument, which can be captured by associating with it argumentation schemes that feature largely in Araucaria [6]. Figure 12 shows the diagram with schemes appropriately associated, which are **Argument from the Sameness of Meaning** (If “A” means the same as “B” and A is true, then B is true) and **Argument to Common Cause** (If A is similar to B , C caused A, and “If A is similar to B and C caused A, then the similarity between A and B is caused by C’s causing both A and B”, then C caused both A and B).

Thus the diagram exactly same as one stored in AraucariaDB has been obtained. To develop one like Figure 5 for this argument, it is necessary to incorporate schemes associated with a set of annotated literals and their relationships in the diagram as a node, as shown in Figure 13. For this example, we (i) used the usual two-valued truth values, but with such a complete lattice structure as $\mathcal{TWO} = (\{\mathbf{t}, \mathbf{f}\}, \leq)$ such that $\mathbf{f} \leq \mathbf{t}$ that amounts to a specialization of LMA to two-valuedness, and (ii) deleted a particular proposition which doesn’t represent a fact but

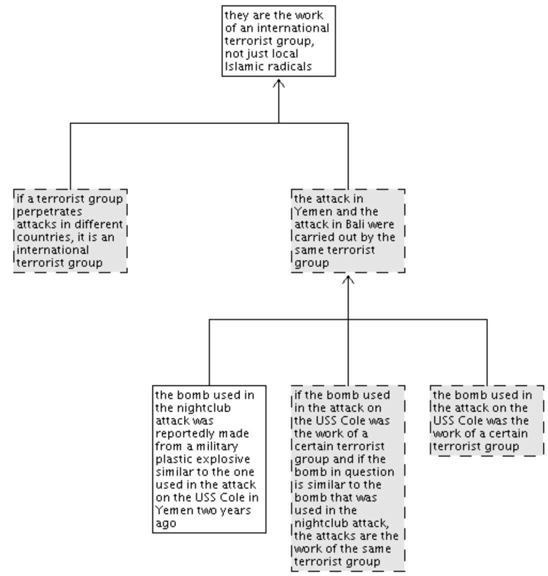

Figure 11. The diagram complemented with missing premises.

a rule, because rules are to be represented as relationships between propositions, that is, edges of the diagram.

The EALP knowledge base is constructed from this completed argument diagram as follows.

$$
\begin{aligned}
KB = \{\; & \\
& international\_terrorist\_group(attack\_in\_Bali) : t \leftarrow scheme(argument\_from\_the\_Sameness\_of\_Meaning) : t \;\&\; same\_terrorist\_group(attack\_in\_Yemen, attack\_in\_Bali) : t \;\&\; same\_meaning(operating\_in\_different\_countries, international) : t, \\
& scheme(argument\_from\_the\_Sameness\_of\_Meaning) : t \leftarrow, \\
& same\_terrorist\_group(attack\_in\_Yemen, attack\_in\_Bali) : t \leftarrow scheme(argument\_to\_Common\_Cause) : t \;\&\; similar(bomb\_used\_in\_Bali, bomb\_used\_in\_Yemen) : t \;\&\; certain\_terrorist\_group(bomb\_used\_in\_Yemen) : t, \\
& scheme(argument\_to\_Common\_Cause) : t \leftarrow, \\
& similar(bomb\_used\_in\_Bali, bomb\_used\_in\_Yemen) : t \leftarrow, \\
& certain\_terrorist\_group(bomb\_used\_in\_Yemen) : t \leftarrow, \\
& same\_meaning(operating\_in\_different\_countries, international) : t \leftarrow \quad \}.
\end{aligned}
$$

Proceedings of the IEEE/WIC/ACM International
Conference on Intelligent Agent Technology (IAT'06)
0-7695-2748-5/06 $20.00 © 2006 IEEE

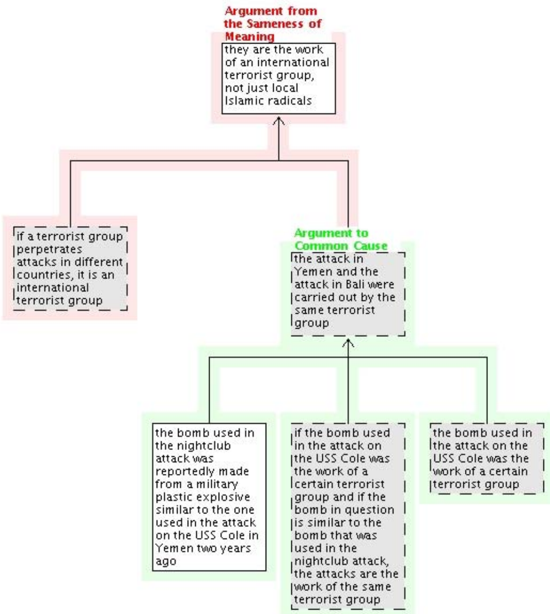

Figure 12. Complementing the diagram with schemes.

5. Concluding Remarks and Future Work

We have proposed a semi-automated method to transform natural arguments in Araucaria to formal arguments in LMA, based on the compatibility and complementarity that these two systems share reciprocally. This makes it easier for users of LMA to prepare knowledge base in EALP since they can think about the knowledge base and arguments in natural language when they want to entrust the argument on a certain issue to the agents as their avatars. For Araucaria users, LMA tells the argument status automatically. The most important features that have made it possible to combine these two systems with an inherently different design philosophy are summarized as (i) enthymemes (truncated syllogism) and its function complementing missing premises that Araucaria has, and (ii) the flexibility of evaluating arguments with user-defined truth values according to uncertainty proper to application domains. In fact, we showed that such a diversity of truth values in LMA [7] allowed for an extensive applicability of it through two considerably practical argument examples. Now it may be fair to say that the collaborative system of Araucaria and LMA is tenable to practical applicability.

In this paper, we have not clearly specified the class or fragment of natural arguments that can be well transformed to formal arguments in LMA. That is, our semi-automated transformation method is supposed to be able to rely on users' indications if need be. Our next step, therefore, will be to address to the question of specifying the class of natural arguments that can be automatically transformed to formal arguments in LMA.

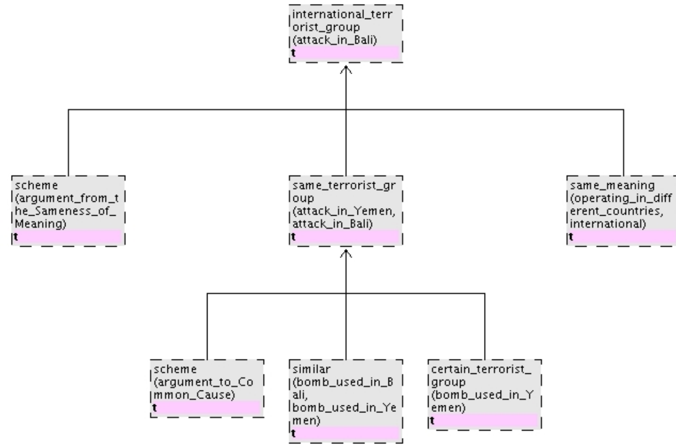

Figure 13. A completed argument diagram.

References

[1] C. I. Chesñever, A. G. Maguitman, and R. P. Loui. Logical models of argument. *ACM Computing Surveys*, 32:337–383, 2000.

[2] M. Kifer and V. S. Subrahmanian. Theory of generalized annotated logic programming and its applications. *Journal of Logic Programming*, 12:335–397, 1992.

[3] P. A. Kirschner, S. J. Buckingham Shum, and C. S. Carr, editors. *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making*. Springer-Verlag, 2003.

[4] R. E. Nisbett. *The Geography of Thought: How Asians and Westerners Think Differently...and Why*. The Free Press, 2003.

[5] H. Prakken and G. A. W. Vreeswijk. Logical systems for defeasible argumentation. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic*, pages 219–318. Kluwer, 2002.

[6] C. A. Reed and G. W. A. Rowe. Araucaria: Software for argument analysis, diagramming and representation. *International Journal of AI Tools*, 14(3-4):961–980, 2004.

[7] H. Sawamura and G. W. Jeong. Diverse applications of an automated argument system based on logic of multiple-valued argumentation. In *LNAI(to appear)*, volume 4078. Springer-Verlag, 2006.

[8] H. Sawamura, T. Takahashi, and K. Matsunaga. An eastern specialization of logic of multiple-valued argumentation to tetralemma originated in india. In *Proceedings of the second Indian International Conference on Artificial Intelligence(IICAI-05)*, pages 1274–1291, 2005.

[9] T. Takahashi and H. Sawamura. A logic of multiple-valued argumentation. In *Proceedings of the third international joint conference on Autonomous Agents and Multi-Agent Systems(AAMAS-04)*, pages 800–807. ACM, 2004.

Proceedings of the IEEE/WIC/ACM International
Conference on Intelligent Agent Technology (IAT'06)
0-7695-2748-5/06 $20.00 © 2006 IEEE