Twente Argument Schema Annotation Manual v 0.99b

Rutger Rienks and Daan Verbree

University of Twente, Enschede, the Netherlands
{rienks,verbree}@ewi.utwente.nl,

1 Introduction

This manual describes the Twente Argument Schema (TAS) and how it should be applied. TAS has been developed in order to reveal the argumentation structures that take place in meeting discussions.

The TAS coding scheme is a model that formalizes observations related to argumentation patterns in meetings. The resulting annotations reveal information about the trail or path that has been taken along the course of discussions and creates the possibility of preserving the arguments and their coherence relations for future explorations.

1.1 Annotating Sensor Information

On top of raw sensor information that can be obtained from meeting discussions, various coding schemes, or annotation formats can be created in order to obtain higher level information that enables e.g. the human interpretations of this sensor data. An annotation format can be seen as a model formalizing what can be observed in the data. [Reidsma et al., 2004]. The actual models depend on the research objectives or the application.

This higher level information is useful for a number of reasons: In the first place this creates so called 'ground truth knowledge' enabling the evaluation of new techniques for automatic annotation. And in the second place, as annotations abstract from signal data they can take the observations to a level that in turn can be used for social psychological research to test or to develop models of human behavior.

An example of a type higher level annotation are for instance dialogue act annotations that are created on top of a transcription layer. Transcriptions are a written version of the actual speech that took place in a meeting. Other examples of higher level annotations that can be made are e.g. annotations of named entities, focus of attention as well as argumentation patterns.

1.2 The Ami Corpus

This coding scheme was initially devised to create argumentation diagrams for the meetings recorded in the Augmented Multiparty Interaction (AMI) Project.

The AMI project concentrates on multi-party interaction during meetings and is focused on the use of advanced signal processing, machine learning models and social interaction dynamics to improve human-to-human communications. In particular the development of tools and models that provide insight into the meetings are of primary concern.

Over one hundred hours of meetings was captured for the AMI meeting corpus in three distinct smart meeting rooms. All the captured meetings had to follow a script that described the global theme and the global structure of their meeting. There were no constraints on the way they gave content to their contribution.

The resulting format consisted of four people meetings constituting a design team from a small company, RealReactions. In these meetings, the participants, take four different roles: a project manager (PM), user interface specialist (UI), marketing expert (ME), and industrial designer (ID). The teams design a new kind of remote control from start to finish over a series of four meetings. In the four meetings, the same person takes the same role every time, but won't necessarily always be sitting in the same place.

The main corpus contains data from a wide variety of sensors : three global video cameras, two microphone arrays, individual close-up cameras, lapel microphones, smart pens, digital white-boards etc. Transcriptions were created for all the meetings in the AMI corpus, following strict annotation guidelines [Moore et al., 2005]. For more information about the Ami corpus, see Carletta et al. [2005].

In order to make the annotations reproducible, not by a machine, but by human annotators, as well as to verify the expressiveness of the schema, this manual has been written. It describes what the schema was meant for, how it is constructed and how it should be applied on top of a transcription layer.

2 The TAS Schema

The Twente Argument Schema (TAS) is a schema designed to create argument diagrams from meeting discussion transcripts. Following most of the diagramming techniques studied, application of the method results in a tree structure with labelled nodes and edges. The nodes of the tree contain parts of, or even complete, speaker turns whereas the edges represent the type of relation between the nodes. The complete label set is shown in Table 1.

In short TAS accounts for capturing the most important conversational moves in dialogues where participants discuss the pros and cons of certain solutions to a problem, providing arguments in favor of or against the various solutions. TAS distinguishes acts in which issues are raised (questions put forward) and statements for a position that are made. It allows one to indicate whether a statement is strong or weak. Whether statements agree or disagree with each other can be marked in the relations. In many cases statements are not simply in favor or against but variations of each other: restatements, specializations or generalizations.

Node labels	Relation labels
Statement	Positive
Weak statement	Negative
Open issue	Uncertain
A/B issue	Request
Yes/No issue	Specialization
	Elaboration
	Option
	Option exclusion
	Subject-to

Table 1. The labels of the Twente Argument Schema

TAS was constructed in a way that it preserves the conversational flow. By applying a left-to-right, depth first search, walk through on the resulting trees, the reader is able to read the resulting trees as if reading transcripts. This is realized by assuring that in principle every next contribution of a participant becomes a child of the previous contribution, unless the current contribution relates more strongly to an ancestor. An example of a resulting argument diagram is shown in Figure 1.

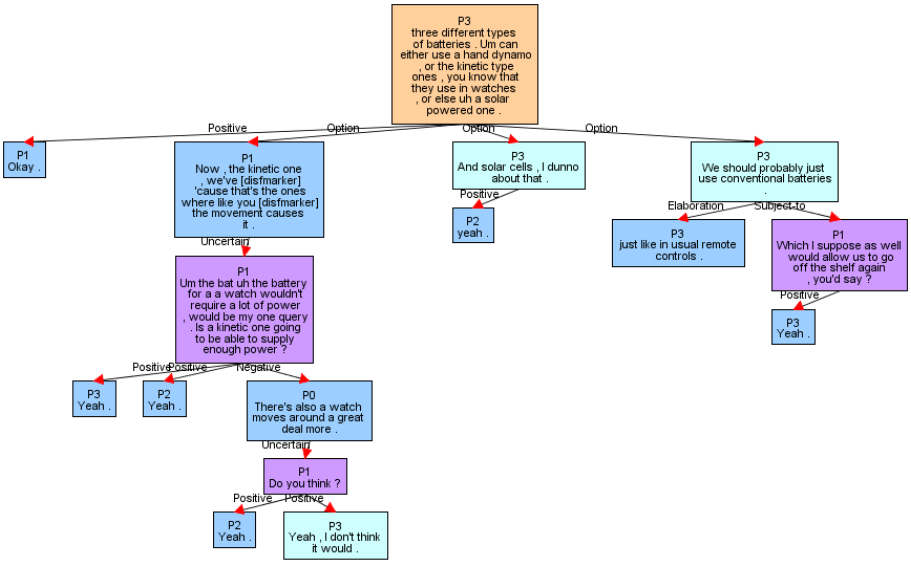

Fig. 1. A possible resulting Argument Diagram

2.1 Related theories

There exist a number of different theories for labelling (transcripts of) meetings, and we have examined most of them in order to distill one final schema by combining the best of all theories examined. Among them are : Rhetorical Structure Theory (RST) [Mann and Thompson, 1987], Toulmin's model [Toulmin, 1958], Dialog Acts (DAs) [Bunt, 1979] and the IBIS scheme [Kunz and Rittel, 1970b]. None of these methods could by themselves be used to our satisfaction in creating a tree which visualizes the structure of the discussion the way we foresaw it. Hence we decided to compose our own. For a more elaborate background about the rise of the schema the reader is referred to Rienks and Heylen [2005] and Van der Weijden [2005].

2.2 The Unit Labels

The content of the nodes correspond in granularity to the size of speech acts, resulting in most of the times in the size of a complete utterance. If utterances contain more than one act, they are split up into more than one node. In line with Galley et al. [2004] backchannel utterances such as 'uhhuh' and 'okay' are filtered out and to be neglected, since they are generally used by listeners to indicate they are following along, and not necessarily indicating (dis)agreement. The nodes in our model consist of issues and statements.

In the IBIS model issues are represented as questions [Kunz and Rittel, 1970a]. This is due to the fact that issues can be seen as utterances with a direct request for a response, in the same way as a question is generally followed by an answer. Fundamental questions with respect to conversational moves are *yes-no questions* and *why questions* [Kestler, 1982]. A yes-no question admits only two kinds of answer, be it either supportive, or negative. A yes-no question rules out the *option* 'I don't know' expressing uncertainty. Another type of question one could ask is an *open question*, this question can, in contrast to, yes-No questions be answered without the limitation of a predefined set of choices. The number of positions participants can take thus depends on the set of possible options enabled by the type of question or issue.

In our Schema we have defined three different labels for our nodes to represent the issues: The **'Open issue'**, the **'A/B issue'** and the **'Yes-No issue'**. The open issue allows for any number of possible replies possibly revealing positions or options that were not considered beforehand. This in contrast with the A/B issue, that allows participants to take a position for a countable number of positions which should be known from the context (c.f. 'Would you say ants, cats or cows?'). The yes-no issue, in line with the yes-no question directly requests whether the participants positions agree or disagree with the issue. The reader should note that we leave out the why question. This is done because a why question can be modelled as an open question with a clarification relation, explained in the next section.

Open Issue This label is to label issues that are raised where every possible response could be a solution. 'Wh. questions'
Example : 'What does it prove?' can be labelled as Open Issue.
A/B Issue This label is to label issues that are raised where the possible responses are explicitly enumerated.
Example : 'What is the most intelligent animal: A, B, or C?' can be labelled as A/B Issue.
Yes/No Issue This label is to label issues that are raised where the possible responses are Yes and No.
Example : 'Are all those name fake?' can be labelled as Yes/No Issue.

Participants' positions are generally conveyed through the assertion of a **statement**. The content of a statement always contains a proposition in which a certain property or quality is ascribed to a person or thing. A proposition can be a description of facts or events, a prediction, a judgement, or an advice (Van Eemeren et al. [2002]).

Statements can vary in the degree of force and scope. It can happen that meeting participants make remarks that indicate that they are not sure if what they say is actually true. Toulmin [1958] uses a qualifier in his model to say something about the force of what he calls 'claim'. When this qualifier is introduced, it is possible that the assertion is made with less force. As Eemeren [2003] points out that the force of an argument can also be derived from lexical cues such as by expressing the words 'likely' and 'probably'. To be able to represent this we introduce the label **'weak statement'**.

Statement This label is to label claims without a weakening qualifier.
Example: 'I Heard about their football team' can be labelled as statement.
Weak Statement This label is to label claims with a weakening qualifier.
Example : 'Yeah, probably' can be labelled as a weak statement.

2.3 The Relation Labels

Relations can only exist between nodes. For this we have defined a number of relations that can exist between the labelled nodes. When engaged in a discussion or debate, the elimination of misunderstandings is a prerequisite in order to understand each other and hence to proceed [Neass, 1966]. Participants in a discussion, according to Neass, eliminate misunderstandings by clarifying, or specifying their statements. These moves can e.g. be observed in the criteria definition phase, of the decision making process.

For a yes/no-issue the contributions that can be made are not related to enlarge or to reduce the solution space, but to reveal one's opinion to the particular solution or option at hand. In a conversation people can have a positive, negative or neutral stance regarding statements or Y/N-issues. For this purpose the labels

'Positive', **'Negative'** and **'Uncertain'** are introduced. With the aim to reveal whether contributions from participants are either supportive, objective, or unclear. The positive relation can exist for example between a yes/no-issue and a statement that is a positive response to the issue or between two statements agreeing with each other. When one speaker states that cows can be eliminated as being the most intelligent animals and the response from another participant is that cows don't look very intelligent, then the relation between these statements is positive. The negative relation is logically the opposite of the positive relation. It is to be applied in situations where speakers disagree with each other or when they provide a conflicting statement as a response to a previous statement or a negative response to a Yes/No-issue. In a case where it is not clear whether a contribution is positive or negative, but that there exists some doubt on the truth value of what the first speaker said, one should use the uncertain relation. From experience with the annotations it appears that in most cases it can easily be seen by the annotator whether the remark is mostly agreeing or mostly showing doubt.

Positive This relations relates statements (child) to issues (parent) and issues to issues and statements to statements. The label is used if the child aims to support the parent. *Example* : The relation between the statement, 'I vote for ants.' the statement 'Yeah, same.' can be labelled as positive.
Negative This relations relates statements to issues, issues to issues and statements to statements. The label is used if the child aims to refute the parent.
Example : The relation between the statement, 'I would bet on cow.' and the statement 'I would eliminate cow anyway.' can be labelled as negative.
UncertainThis relations relates issues to statements. The label is used if it is unclear how, being it either positive or negative, the child (issue) relates to the parent (statement).
Example : The relation between the statement, 'There is a high degree of similarity.' and the open issue 'What about HD distance?' can be labelled as an uncertain relation.

A specification occurs in situations where a question is asked by one of the speakers and someone else asks a question which specializes the first question resulting in a possible solution space with more constraints. The contribution 'Which animal is the most intelligent?' can be specialized with the following proceeding contribution 'Is an ant or a cow the most intelligent animal?' which again can be specialized if one for instance asks 'Are ants the most intelligent animal?'. For these occasions we introduce the label **'Specialization'**. The 'Specialization' label can for instance be applied when a particular issue generalizes or specializes another issue. It could on the other hand also very well happen that a person is not yet satisfied with the information or the argument explained. This person can explicitly invite the previous speaker to elaborate on his ear-

lier statements. For these situations we define the relations **'Request'** in case someone asks for more information and the relation **'Elaboration'** if a person continues his previous line of thought and adds new information to it.

Specialization This relations relates statements to statements and issues or issues. The label is used if the child is a specialization of the parent, and the child has a different semantic meaning, but the children's children still provide either support or refutation to the parent.
Example : The relation between the open issue, 'What is the capital of Moldova?' and the open issue 'What sounds most Moldovian?' can be labelled as a specialization relation.
Elaboration This relations relates issues to issues. or issues to statements or statements to issues, or statements to statements. The label is used if the child repeats, or asks to repeat the parent in other wordings. It can also be used if additional information about the parent is asked or given.
Example : The relation between the statement 'Ants are able to modify our garden' and the statement 'They are even able to modify whole parks.' can be labelled as a elaboration relation.
Request This relations relates statements to Yes/No or A/B or Open issues. The label is used if the child asks for more information about the parent. This can be information about the topic or asking to repeat the statement when someone did not hear it.
Example : The relation between the statement, 'Ants are able to modify our garden.' and the and the Open issue 'What do you mean by modifying the environment?' can be labelled as a request relation.

Whenever the issue is defined, an exchange of ideas about the possible answers or possible solution naturally occurs in the decision making process. Whenever a statement is made as a response to an open-issue or an A/B-issue it might reveal something about the position of participant in the solution space. In general he provides an **'Option'** to settle the issue at hand. For example when a speaker asks 'Which animal is the most intelligent?' and the response from someone else is 'I think it's an ant' the option relation is to be applied. The opposite of the option relation is the **'Option-exclusion'** relation, and it is to be used whenever a contribution excludes a single option from the solution space.

Option This relations relates statements to issues and statements to statements. The label is used if the child is a possible answer, option or solution with respect to the parent.
Example : The relation between the open issue, 'What is the capital of Moldova?' and the statement 'I would say Chisinau' can be labelled as an option relation.
Option Exclusion This relations relates statements to issues and issues to issues. The label is used if the child excludes one or more possible answers, options or solutions with respect to the parent.
Example : The relation between the open issue, 'What is the most intelligent animal?' and the yes-no issue 'I wouldn't look at an ant as a brilliant individual, by itself it is nothing, right?' can be labelled as an option exclusion relation.

The final relation of our set is to be applied when the content of a particular contribution is required to be able to figure out whether another contribution can be true or not. We named this the **Subject to** relation, which is somehow related to the concession relation in Toulmin's model. It is to be applied for example in the situation where someone states that 'If you leave something in the kitchen, you're less likely to find a cow' and the response is 'That depends if the cow is very hungry'. So the second contribution creates a prerequisite that has to be known before the first contribution can be evaluated. If the cow is very hungry the support could be either positive or negative. The uncertain label is not to be applied it this case, as the stance of the person in question is clear once the prerequisite is filled in. The uncertain label is merely to be used when an issue is preceded by a request or a specialization.

Subject To This relations relates statements to Yes/No or A/B issues or statements to statements. The label is used if the child points out criteria or dependencies that have to be fulfilled before the parent can be supported or refuted.
Example : The relation between the statement, 'If you leave something in the kitchen, you are less likely to find a cow there.' and the statement 'That depends if the cow is very hungry.' can be labelled as a subject to relation.

3 Annotating TAS with ArgumentA

This section will guide a user over ArgumentA through the process of creating TAS Annotations. In order to create TAS annotations a program called ArgumentA has been developed. A prerequisite to start annotating is that the transcriptions of the meetings are there. Once this is the case the process of creating the annotations is split up into two phases. The first phase, the discussion selection phase, described in section 3.4, segments the meeting into discussions and non discussions. The second phase then deals with the actual labelling of the

pre-segmented discussions. This second phase can be split up into the labelling of the TAS-units, described in section 3.5 and the labelling of the TAS-relations, described in section 3.6.

3.1 Preserving the conversational flow

As we are working on transcripts, it is best for our model to be constructed sequentially in order to follow the line of the discussion. To preserve the order of the discussion in the model we decided that, when applying the schema, the algorithm or annotator should follow a depth first search algorithm [Cormen et al., 1990] when extending it. This means that in principle every next contribution becomes a child of the previous contribution, unless the current contribution relates stronger to the parent of the previous contribution. If it relates stronger to the parent or an ancestor of the previous contribution, the previous contribution itself can not be linked to later contributions, it is no longer part of the so called 'fringe' (the possible places where to attach the upcoming contributions to). By following this procedure, the resulting tree structure can still be read synchronously.

3.2 Freedom of the annotator

One of the drawbacks of argument diagramming that is often mentioned is that there is no correct diagram. Walton [1996] for instance showed that various different argument diagrams can be instantiated by one single text. Although this is true, an interesting point here is the analogy that can be drawn between RST and Argument Diagramming. As Reed and Rowe [2001] point out that Mann and Thompson suggest that the analyst should make *plausibility judgements* rather than absolute analytical decisions, it is implicated that there can be more than one reasonable analysis. This also goes for argument diagramming, where the evaluator is free to interpret and to create that diagram that he considers the most appropriate according to his or her perception. As long as the schema is applied correctly, its purpose anyhow will be apparent.

We will now provide a step by step walkthrough through the annotation process.

3.3 Starting ArgumentA

Once ArgumentA has been started, the window displayed in Figure 2 a window pops up. In this window a selection of annotation tool can be made. In this case only two tools are listed :ArgumentA, for the creation of the TAS-labels and the 'Discussion Select'-tool that enables the segmentation of full transcripts into discussions and non-discussions.

Independent of what we choose the window depicted in Figure 3 pops up asking to identify ourselves as annotator. The task here is to select yourself from the list, or in case you are not listed, to create yourself as a new annotator.

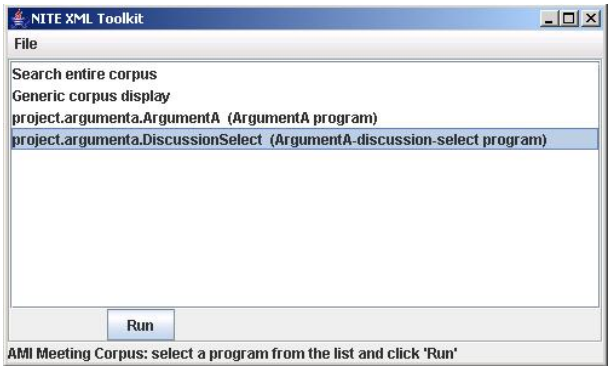

Fig. 2. ArgumentA: start-up screen

Fig. 3. ArgumentA: Select or add yourself as annotator

This way all the annotations are saved for each of the annotators individually enabling easy obtainment and continuation at a later stage.

Once you selected yourself as an annotator you have to select the name of the meeting, also called the observation, that you will be annotating. A possible list that could be presented is shown in figure 4.

So dependent on the initial choice of tool, the annotations can be performed, with the tool aware of who you are and what meeting you are going to perform the annotations on.

3.4 Pre segmentation - labelling Discussions

The first step in the process is to identify the discussions in the meeting. Where arguments take place, discussions evolve. The main question one should ask one selves when looking for discussions in a transcript is whether there is an issue at hand for which more than one viewpoint is being proposed. If this is the case then usually a discussion is going on. The discussion usually starts where the

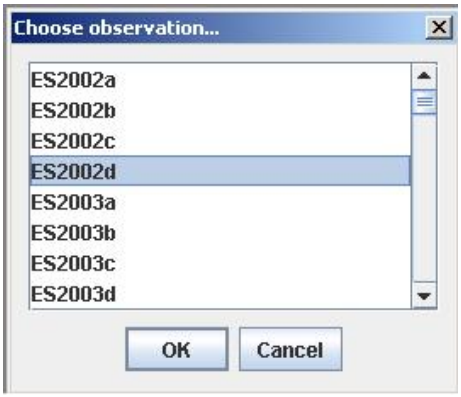

Fig. 4. ArgumentA: Choose a meeting you have been assigned to annotate.

issue is raised and usually stops where some sort of agreement is reached, or the issue is postponed or abandoned for whatever reason.

The 'Discussion Select'-tool initializes the way depicted in figure 5. In the panel where the transcription of the meeting is shown, the contributions of the participants can be selected by clicking the mouse. By dragging the mouse, subsequent contributions can be added to the selection (See Figure 6). Please note that only discussions can be selected that occur in one uninterrupted sequential part of the meeting. Though is not an absolute requirement as one could have discussions that might last even longer than one complete meeting, or discussions that could be spread all over the meeting, we decided to enforce this within the tool so that in the end we could confine ourselves to completely isolated and subsequent discussions from which a diagram can be created. If a issue is re-raised and again is debated in another part of the meeting, the annotator is able to select that part of discussion and create an own diagram for it.

Once the discussion is selected, it can be given a name and saved into the corpus containing the annotations. Once a discussion is successfully saved this is shown into the log window of the tool and the text from the saved discussion will now appear grayed and un-selectable in the transcription view.

After all the discussions have been selected from the transcript (It could even be that there exist not even one discussion in the whole meeting), the application can be closed. When asked if the information can be saved click 'yes' and restart the application if you want to select discussions in another meeting, or start ArgumentA in case you want to create diagrams from the discussions selected.

3.5 TAS unit Labelling

The process of creating an argument diagram is a continuous process that alternates between labelling a segment, usually the size of a speaker-turn, in a

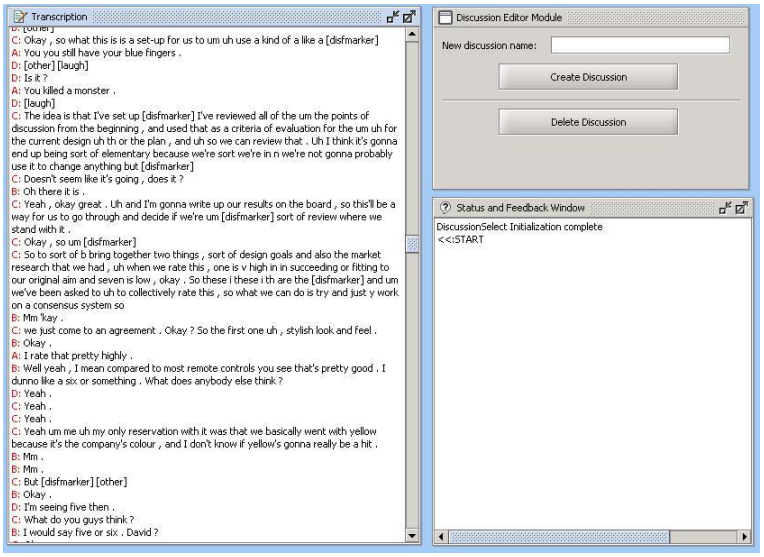

Fig. 5. ArgumentA: Opening screen of the discussion labeller

preselected discussion and adding the labelled segment into the argument diagram representing the discussion by defining a relation between the currently labelled segment and a previously labelled segment.

ArgumentA should be started with a meeting on which the discussions have been already defined. Once initialized you can select the discussion from which you want to create a diagram by selecting it in the discussion select box shown in Figure 8.

Now the units that are to be labelled can be selected from the transcription of the discussion that appeared right after selection. In general a complete utterance or speaker contribution of a single participant can be assigned a label. In case this utterance can be split-up into various individual labels the annotator can do this by first selecting part of a speaker-turn instead of the complete speaker-turn. Once the size of the unit is determined, by pressing the letter 'd' a popup window appears as shown in Figure 9. From this pop-up window the appropriate label can be selected. If you are uncertain which label to attach, see section 5.2.

Once the label has been selected, the unit will appear in the transcription pane prefixed with the selected label. One is able to alter the label anytime by selecting it and assigning a different label in the tool encircled in the middle of figure 9. Once satisfied with the label one should note that the selected unit becomes available in the color of its label in the tree-viewer panel as a node of the tree at the left bottom of the screen. (See Figure 10). One is now able to either label another unit, or to link the created unit to another unit in the tree by defining a relation and assigning a relation label between the two.

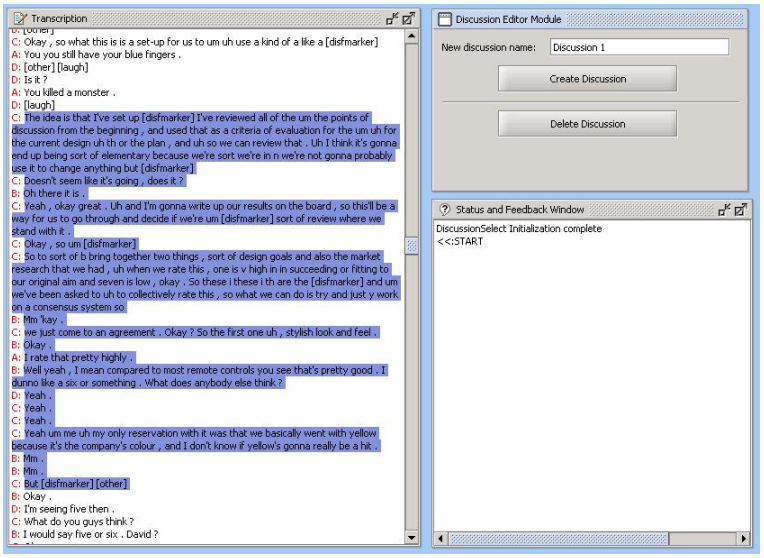

Fig. 6. ArgumentA: Select a discussion by dragging the mouse

3.6 TAS relation Labelling

The relation labelling starts with an initial unit label (node) which constitutes the root of the argument diagram. In Figure 11 three nodes have been assigned a label and are ready to be connected by having their relations defined.

The actual creation of the relation is very simple. One selects the node that one wants to position as a child of (under) another node in the resulting diagram. By right clicking on this node an arrow appears that can be connected to another node in the tree. Make sure when doing this that the node where you connect to is either the previous labelled unit in the transcription, or a first-line ancestor of this node in order to maintain the conversational flow in the resulting diagram when applying a depth-first search algorithm. Once the parent is selected a menu will pop-up shown in Figure 12.

In the transcription view both units will now be colored showing the created relation. From the popup-menu the relation type can be selected. In case you do not know which label to assign, refer to Section 5.3. Once the desired label type is clicked, the tree window will be updated (the node is now part of the tree) and the relation will be added to the list shown at the right top of Figure 13. In case one wants to alter or delete a relation one can best select the relation in this list, highlighting the text of both source and target node in the transcription view as well as depicting the relation in the relation window encircled at the middle top of Figure 13. In this window one is able to modify or delete the relation.

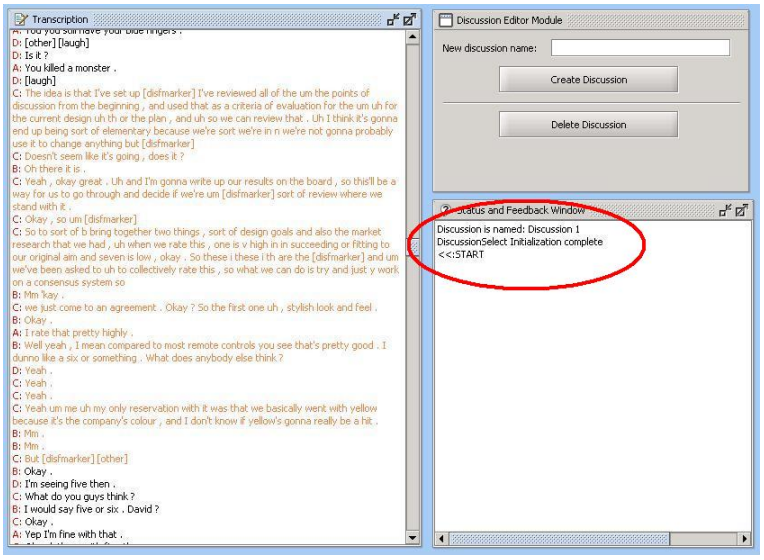

Fig. 7. ArgumentA: Discussion added to the annotations.

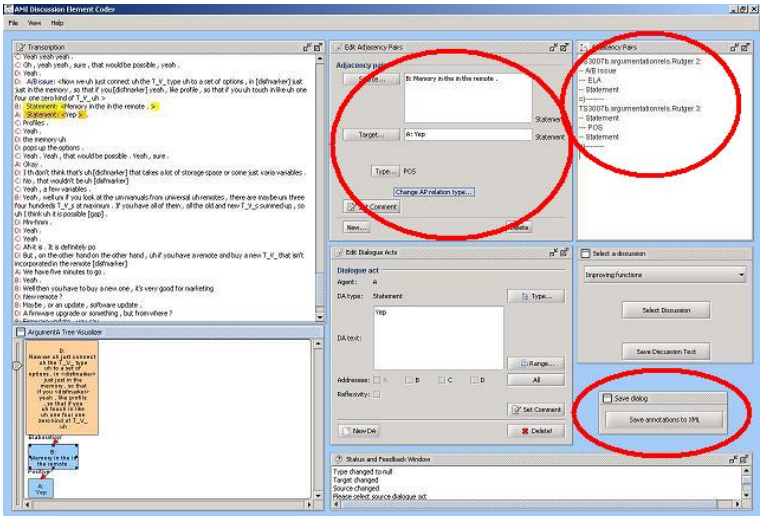

Fig. 13. ArgumentA: Altering a Relation Type and Exporting annotations

Keep in mind that, in order to preserve the conversational flow, one is only allowed to link the unit to other units that are part of the fringe, meaning the previous contribution or its direct ancestors. Once the whole diagram is

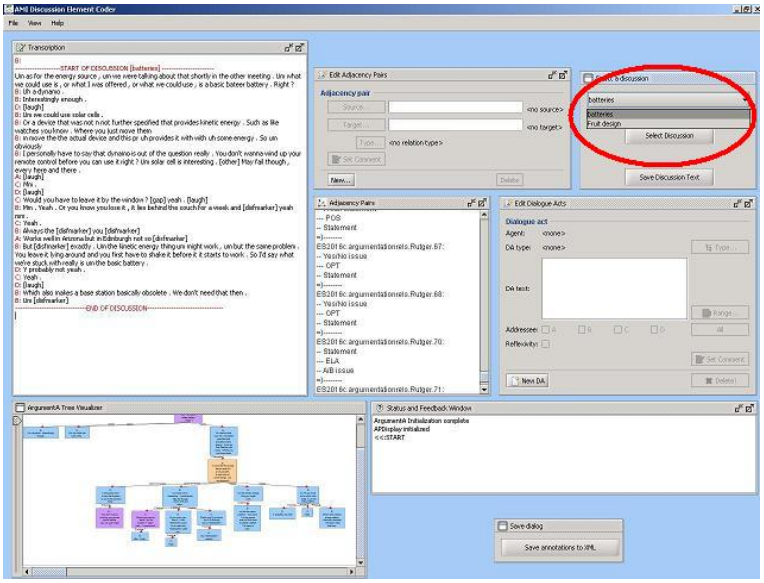

Fig. 8. ArgumentA: Selecting the discussion that is to be annotated

completed, one can exit the application. When asked to save the changes click 'yes'. If one needs to export the whole diagram to a flat XML-format this can be done by clicking the button 'Save to XML' encircled at the right bottom of figure 13.

4 ArgumentA viewer

A special tool has been created in order to view the annotations created by the annotators. The ArgumentA viewer only works though on the exported XML format produced by ArgumentA and not on the NXT-format.

The ArgumentA viewer (shown in Figure 14) expects a directory with XML-files to be given. All the discussions encountered in this directory will be listed at the left top of the viewer. All the retrieved discussions in the list can be clicked and their corresponding argument diagram will appear at the right panel. Furthermore the discussions in the list can be queried for specific node-relation-node occurrences. The result, the discussions containing that specific node-relation-node will appear in an updated list at the left top including some statistical information at the left bottom. If a discussion is selected, the diagram will appear with the sought after node-relation-node occurrence shown in red.

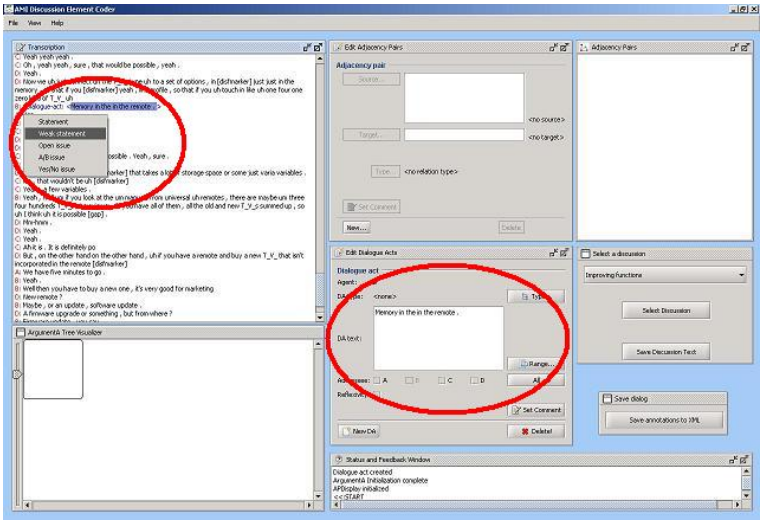

Fig. 9. ArgumentA: Utterances can be selected and labelled

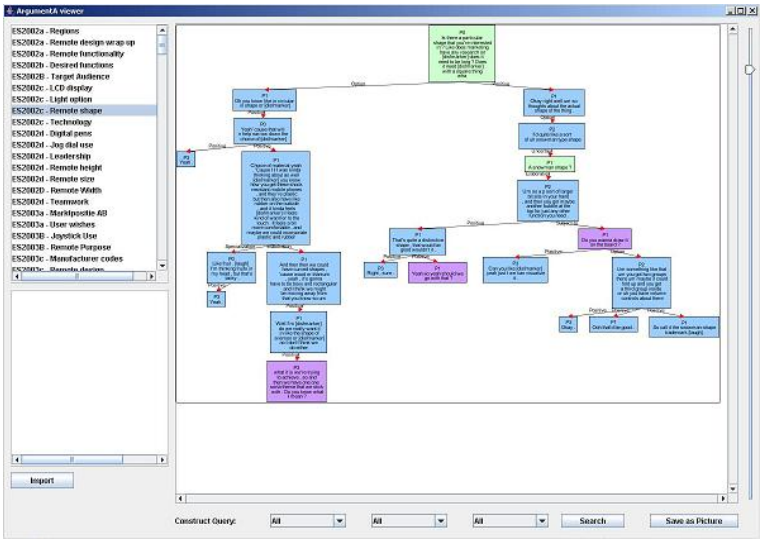

Fig. 14. ArgumentA Viewer : A tool to visualize the XML annotations

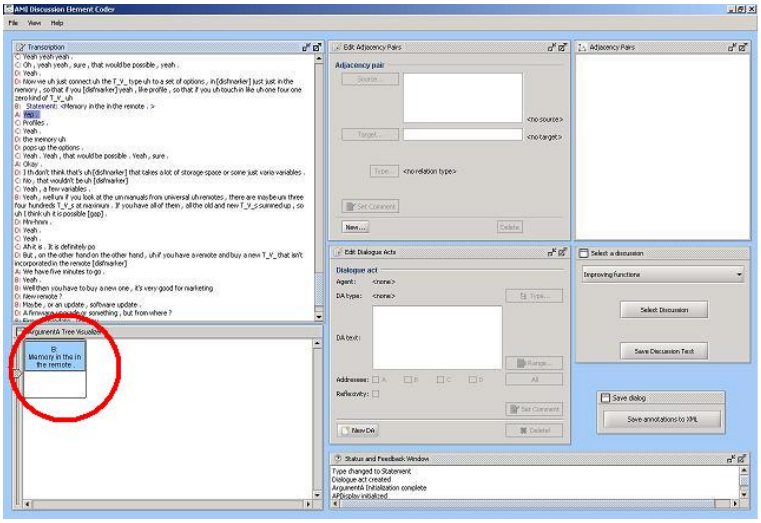

Fig. 10. ArgumentA: Labelled utterances appear as unconnected nodes in the Argument Tree Viewer at the bottom.

5 Frequently Asked Questions

5.1 How to decide which unit label to assign?

When it is clear that a particular (part of) an utterance will be assigned a label and one is unsure about the label to assign, one should proceed with the following steps :

(1) The first question one should ask himself is if the selected unit is merely an issue or a statement. An Issue can usually be detected by the fact that it raises a questions or that it explicitly invites others to respond e.g. *'What is your opinion on this?'*. Where a statement is more like an expression of facts e.g. *'This bottle is green'*. In case you think it is neither a statement nor an issue, than it is likely not to contribute to the discussion and one can leave the unit unlabelled. The unlabelled units can be taken together into a group that could be called 'unknown'. Which is required e.g. in the case of machine learning.

If the unit is more like a statement, one only has to decide about if it is a weak statement or just a normal statement. A weak statement is a less strong ventilation of opinion that usually can be recognized by words as 'probably', 'likely' and 'maybe'.

If the unit is more like an issue the thing to do is to take a look at the answers that the others can bring, or the number of options left for the other participants to take. In case of the issue *Do you think the chair is blue?* the number of possible answers are just two-fold (yes and no) and therefore the label to pick is a 'yes/no issue'. If there exist a specific countable number of answers e.g. *What is the color of this chair?* that we speak of an A/B issue. So

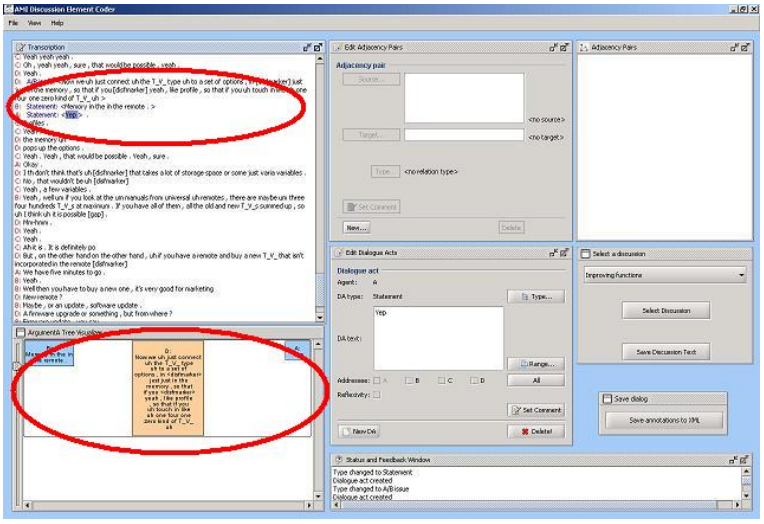

Fig. 11. ArgumentA: Three units ready to be connected

the number of possible answers is here limited due to the fact that the possible responses are confined to the countable set of 'colors'. The final label that can be given to an issue is the open-issue label. Here the set of possible options to respond to the issue is endless, e.g. *John, what do you think of this chair?* is an example of an open-issue.

5.2 When to split a speaker-turn into two or more units?

A speaker turn can be split-up (assigned more than one label) if it consists of more than a single issue or statement. This could be the case if a person first responds to a previous issue and then raises another issue. Consider the following example: *P0: I'd say we paint the chair yellow. Anyone an idea where to get the paint?*. Here the contribution of P0 consists clearly of two parts. It begins with a statement that is followed by an open-issue.

5.3 How to decide which relation label to assign?

To decide upon which label to attach to a relation between two units one should pay attention to how the the parent and the node to be attached relate to one another. The following questions can help you out:

- Does the child node express a sense of agreement or disagreement in relation to the parent? If so, choose 'Positive' in case of agreement and 'Negative' in case of disagreement.
- is the child node raising ideas or excluding ideas with respect to the parent node? If so choose 'Option' or 'Option-exclusion' as a label.

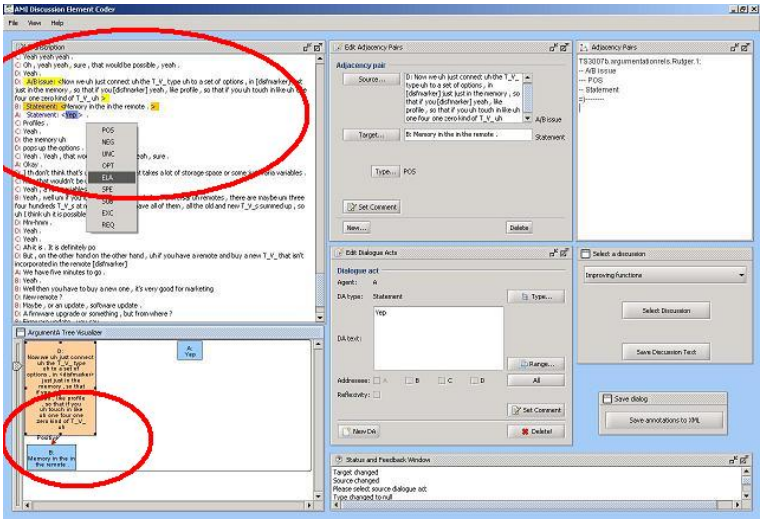

Fig. 12. ArgumentA: Creating a relation between two units

- Does the child node add extra information to something previously mentioned that is still the focus of the discussion? If so, choose 'Elaboration'.
- Does the child node rephrase an issue and constraining thereby the set of possible answers? Choose 'specialization'.
- Does the outcome of what is mentioned in the child node influences the truth value of the parent node? Choose 'Subject-to'.
- Is the child requesting for more information about a certain issue (parent), e.g. *I think we need more information*. Choose 'Request'. In case you are absolutely sure the child node relates to the parent, but none of the above labels seem to apply, choose 'Uncertain'.

5.4 Where the current unit to attach?

In principle the selected unit or node can be attached to any first-line ancestor in the tree. Dependent on the issue or statement where the current unit relates most to, it should be connected to. It is more likely, though not unimaginable, that a unit is attached to its directly preceding unit, than to a unit uttered quite a while ago.

5.5 Can I export the annotation to another XML format?

Yes this is possible from ArgumentA. Please click the button 'Export to XML' once the argument diagram of the discussion is ready. An output folder can be specified. Another possibility is to write a Perl-script that runs on the NXT-annotation created by ArgumentA.

6 Acknowledgements

This work was partly supported by the European Union 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction, FP6-506811, publication AMI-XX).

Bibliography

H.C. Bunt. Conversational principles in question-answer dialogues. *The theory of questions*, 1979.

J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner. The ami meetings corpus. In *Proceedings of the Measuring Behavior 2005 symposium on "Annotating and measuring Meeting Behavior"*, 2005. AMI-108.

T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms.* MIT Press, 1990.

F.H. Eemeren. A glance behind the scenes: The state of the art in the study of argumentation. *Studies in Communication Sciences*, 3(1):1–23, 2003.

M. Galley, K. McKeown, J. Hirschberg, and E. Shriberg. Identifying agreement and disagreement in conversational speech: Use of bayesian networks to model pragmatic dependencies. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 669–676, 2004.

J.L. Kestler. *Questioning Techniques and Tactics.* McGraw-Hill, 1982.

W. Kunz and H.W.J. Rittel. Issues as elements of information systems. Working Paper WP-131, Univ. Stuttgart, Inst. Fuer Grundlagen der Planung, 1970a.

Werner Kunz and Horst W. J. Rittel. Issues as elements of information systems. Working Paper WP-131, Univ. Stuttgart, Inst. Fuer Grundlagen der Planung, 1970b.

W.C. Mann and S.A. Thompson. Rhetorical structure theory: A theory of text organization. Technical Report ISI/RS-87-190, University of Southern California, 1987.

J. Moore, M. Kronenthal, and S. Ashby. Guidelines for AMI speech transcriptions. Technical report, IDIAP, Univ. of Edinburgh, February 2005.

A. Neass. *Communication and argument. Elements of applied semantics.* George Allen & Uwin Press, 1966.

C. Reed and G. Rowe. Araucaria: Software for puzzles in argument diagramming and xml. Technical report, Department of Applied Computing, University of Dundee, 2001.

D. Reidsma, R. Rienks, and N. Jovanovic. Meeting modelling in the context of multimodal research. In *Proc. of the Workshop on Machine Learning and Multimodal Interaction*, 2004.

R.J. Rienks and D. Heylen. Argument diagramming of meeting conversations. In A. Vinciarelli and J-M. Odobez, editors, *Multimodal Multiparty Meeting Processing, Workshop at the 7th International Conference on Multimodal Interfaces (ICMI)*, pages 85–92, Trento, Italy, October 2005.

S. Toulmin. *The uses of argument.* Cambridge University Press, 1958.

E. Van der Weijden. Structuring argumentation in meetings : Visualizing the argument structure. Master's thesis, University of Twente, November 2005.

F. Van Eemeren, R. Grootendorst, and F. Snoeck Henkemans. *Argumentation.* Lawrence Erlbaum Associates, 2002.

D.N. Walton. *Argument Structure, A pragmatic Theory.* University of Toronto Press, 1996. ISBN 0-8020-0768-6.