Structuring argumentation in meetings

Visualizing the argument structure

Master Thesis

Erik van der Weijden
University of Twente
Computer Science, Human Media Interaction

Graduation Committee

ir. R.J. Rienks
dr.ir. H.J.A. op den Akker
dr. D.K.J. Heylen

November 2005

Abstract

The goal of this assignment is to find a way to apply some kind of structure to the arguments that are used in meetings in such a way that this argument structure can be visualized, therefore this assignment has the title "Structuring argumentation in meetings: Visualizing the argument structure". An argument structure consists of the arguments that are used during a meeting and the relations that can be established between them. I have chosen to do this because people that are interested in what is discussed during a meeting could normally look into the meeting minutes, but from a visualized argument structure much more information can be gathered than from the text of the meeting minutes. Also a visualized argument structure aids in the understanding of decisions that are made during the meeting.

In order to capture the arguments that are used in a meeting some kind of structure has to be applied to the text. Therefore I have looked at a number of different annotating methods and tried to annotate a meeting with them. When the argument structure was captured in this way, these annotating methods were also used to try to visualize the argument structure. I have listed all the problems that I ran into while using these annotating methods. All of these methods have their strengths and they are very good for the purpose for which they are designed, but none of the methods was built with the visualization aspect in mind and therefore none of them could be used to accomplish the goal of this assignment.

So a new annotating method aiming to capture and visualize argument structures was needed, and therefore we have developed a new method called the "Twente Argument Schema". This method has been used to label the same meeting as I have done with the other annotating methods and this time most of the problems I had with the existing methods were not present while labeling with the new labelset. In order to label a lot of meetings to obtain a whole corpus of meeting annotated with the Twente Argument Schema we have constructed a tool, called the ArgumentA annotation tool, to make the annotating process a little easier.

The first results from annotating with the Twente Argument Schema look very promising when they are compared with the results from other theories, and therefore I found the Twente Argument Schema a useful addition to the great number of annotation methods that already exist.

Preface

One thing I have learnt during the writing of this thesis is that doing research is much harder than I thought. And sometimes this could have a bit of a negative influence on my motivation, but thanks to all the people who supported me the final result is one that I am very happy with.

First I would like to thank my girlfriend for supporting me when I had a hard time when something did not go as I had planned. Also I would like to thank my parents for their continuing support, and also all the people who worked in the Young Talent Room for creating such a nice working atmosphere. Also I would like to thank my graduation committee for the time and effort that they have put into the guidance they have offered me during my graduation process. And my special thanks go to Dennis Reidsma for always helping me with my programming issues and without him my assignment could have never become what it is now. And I would also like to thank all the people that have supported me but did not get mentioned above. Thanks to you all, I could not have done it without you.

Contents

CHAPTER 1

The assignment

In daily life a lot of meetings are held and this costs a lot of people a lot of time. A lot of time is also spent by getting to the location where the meeting is held and sometimes (when there is a traffic jam for example) this also leads to frustration amongst the meeting participants and this not good for the effectiveness and efficiency of the meeting. So it would be nice if there was a way in which the meeting process could be made less time consuming.

With modern techniques meetings can be held with participants who reside at different locations, so the need to get to the location where the meeting is held is not a problem anymore. Such a meeting can be held for example in a Virtual Meeting Room where the attendants of the meeting are represented as characters in a room on a computer screen. Each of the participants sees the room on the computer screen at their location and can interact with the other participants with the help of the computer and in this way meetings can be held. This approach of holding meetings could also work whenever all of the participants are in the same room, but they could still use the extra functionality which the Virtual Meeting Room can offer, like the recording of what is said during the meeting and possibly even automatic generation of meeting minutes.

Since each of the participants of a meeting held in the Virtual Meeting Room has a computer through which they participate in the meeting, this computer could also be used to assist the meeting attendants. A lot of different tasks can be thought of which range from the recording of what is said during the meeting to the creation of a lifelike behavior of the avatars which represent the other participants in the virtual meeting room.

The goal of this project is to create a structure of the arguments that are used in a meeting and to visualize this structure. The decisions that are made during a meeting and the reasons why they are made can be found perfectly in the meeting minutes, but from a visual form of these decisions the same information can be extracted more easily since you do not only see the arguments which you are looking for but also the relations between them.

1.1 Approach

In this paragraph you can find an overview of what I have done in this assignment, and how I have done it. To illustrate the process I will use an example of a fragment of text from a transcript of a meeting. The choice for this particular fragment is made because it contains typical situations which will be important

in the assignment, situations like agreement, disagreement and meeting participants giving their opinion. Although the piece of text below is not a discussion and it only contains one argument it will still be referred to as discussion, because in this assignment the term discussion is used for fragments of text that contain the situations mentioned earlier. The information that is contained in the transcript are the speaker who says something and what he or she says (this I will call the remark of the speaker) and it is depicted below:

p3: *"If the last European championship will be replayed the winner will be?"*
p0: *"Switzerland."*
p3: *"Yeah definitely."*
p2: *"No way."*
p3: *"Switzerland will win."*
p2: *"I see more chances to France."*
p0: *"There's no Spain so."*

The general goal of this assignment is to be able to visualize the usage of arguments and motivations for choices during discussions in meetings, using the text from the transcripts of the meetings. I choose to do this because a person may achieve a better understanding of the argument than they would when it was written as text. Also weaknesses of the arguments that are used can be picked out more easily in this manner [Hair and Lewis, 1990]. I can for example visualize the remarks that are being made as nodes in a tree and the relations that hold between them could be represented by the edges connecting the nodes. When the arguments from the example discussion are structured and visualized a possible outcome of this could be the tree shown in figure 1.1. This tree is an example to illustrate what is meant by a visualized argument structure, but it is not the only correct visualized argument structure of the example discussion. More information on argument structures can be found in chapter 2.

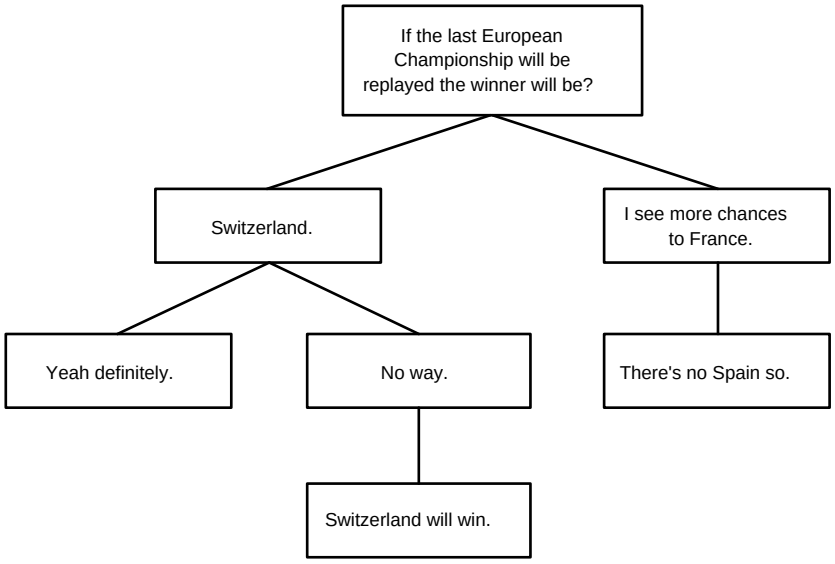

Figure 1.1: *A possible visualized argument structure of the example discussion.*

In figure 1.1 the remarks that are made in the example discussion are represented as nodes in a tree structure. Remarks are made for a reason, people do not just say something, and this can be seen in the figure when you look at the question about who will win the European Championship. Two possible countries that could win are mentioned and these are connected to the question with edges between the node of the question and their own nodes. In this way it can be shown to which remark this is a response, this reaction can be an answer to a question or a way to show how the speaker thinks about what is said before. When P0 says that Switzerland will win the European Championship P2 and P3 both react to this by giving their opinion about

the remark of P0 and this can be seen in the visualized argument structure by the edges that connect the remarks. In this way the entire tree in the figure is built up and in this process the course of the discussion is taken into account too. The tree should be read from top to bottom and take the leftmost branch down the tree and then the branch to the right of it, this is called "depth-first left-to-right". This ensures that the order in which remarks are made is preserved. Also the speaker who makes the remark would make a nice addition to the displayed information in this visualized argument structure.

There are a number of ways in which texts can be annotated in order to extract argument structures from them. In chapter 3 I will present an overview of the various methods and I will choose the ones that I will use to annotate the text from the transcripts, more information about the transcripts can be found in chapter 4. The annotation process with the various theories will be described in chapters 5 up to and including 8, and the general issues of the annotation process independent of the theory will be explained in chapter 9.

After the annotating process with the existing methods was done, we tried if we could develop a new labelset with which we could label the transcripts. In this manner we could try and develop a labelset which aids us even better in realizing the goal of this assignment. This labelset, the Twente Argument Schema, is explained in more detail in chapter 10.

When the text of the transcripts is labelled I still have to produce something with which the argument structure can be visualized (like in figure 1.1). For this purpose I have built a component for a tool which is able to do this. Because it seemed very practical to also be able to annotate transcripts with the same tool that possibility is also present in the tool. This tool will be described further in chapter 11.

1.2 Focus

In this assignment the interest lies in finding answers to questions dealing with agreement, disagreement, discussions, decisions and arguments. The questions that are meant here are the questions that people have about information that is contained in the meeting, questions like "Did person A and person B agree or disagree on topic X?" or "What are the reasons to choose for option M in discussion Y?". Therefore I have tried to find an approach that is able to capture the decisions of a meeting as well as the lines of deliberated arguments. The focus does not lie in formulating an opinion about the contents of the argumentation, but in wanting to identify the relations and the forthcoming structure between the arguments.

The goal of my assignment is to provide people with a graphical representation of this structure between the arguments that are used within a discussion, in this case the discussions from meetings. I want to provide a graphical representation since people absorb the information easier from a graphical representation than from plain text. Also people who did not attend the meeting should be able to extract information from it to be able to figure out what decisions have been made during the meeting and why. To accomplish this I had to look at various annotation methods and the text from meetings had to be labeled for this purpose.

With this labeled text(s) you could also envision some other applications like automated chairman detection, or automated chairman assistance for example, which could include something like introducing a new topic when no new arguments are presented any more in the current topic of the discussion. But the general use of the labeled text for my assignment is the creation of argument structures which can be visualized to provide people easy access to data from meetings.

With the goal of this assignment in mind the focus lies on visualizing the data. The data that is annotated has to be converted to a clear visualized argument structure, and whether the annotating of the data is done automatically or manually is not as important as getting the visualization right. This visualization should be able to give people who look at it information about agreement, disagreement, discussions, decisions and arguments.

CHAPTER 2

Argument structures

In the previous chapter the term 'argument structure' was mentioned, in this chapter I will explain what I mean by it. Before argument structures can be discussed properly the definition of what is called an argument in this assignment is given. This chapter is more than just a definition of a new term, I will also give some examples of them and I will point out what information they contain too.

2.1 What is an argument?

There are a lot of different definitions for or meanings of the term argument of which a few are listed here:

- "A course of reasoning aimed at demonstrating truth or falsehood." [Answers.com, 2004]
- "A discussion in which reasons are advanced for and against some proposition or proposal." [Answers.com, 2001]
- "A reason or the reasoning given for or against a matter under discussion compare: evidence, proof." [Merriam-Webster, 1996]

As can be seen in the list of definitions they all have something in common, they all say that arguments are used as support or objection to the topic of the discussion. For this assignment the definition of arguments that will be used is:

> "An argument is a remark that is made to support or object to another remark that is made."

This definition is used because the focus lies on capturing reasons why decisions are taken and agreement and disagreement between meeting participants.

2.2 What is an argument structure?

In a meeting the participants often have different opinions about topics that are handled in the meeting and this sometimes leads to discussions between the participants. In these discussions people often tell what their

opinion about the subject of the discussion is or why their choice of a solution to the discussion is the right one and the means with what they try to do this are arguments. Also answers to questions can be justified using arguments, or statements can be made which are backed up by arguments. So these discussions have the potential to contain a considerable amount of arguments. And this is the main reason why I choose to take the discussions from the transcripts of these meetings as the source from which I will extract the arguments.

The arguments that are given during a discussion in a meeting do not appear there by coincidence, they are there for a reason. They can for example support a claim that is made. First a structure has to be applied to the situation, and this is done by taking the claim and the argument that supports the claim and define a supporting relation between the two. After this a way to visualize the structure is chosen and one possible visualization of this argument structure is shown in figure 2.1. As can be seen in figure 2.1 the argument structure contains more than just arguments, also the elements that make up the context in which the argument is used are included. In this case these elements are the claim that is supported and the supporting relation between the claim and the argument. The labels in this figure are just there to illustrate the example and it may very well be the case that these labels are not a part of the final form of the argument structure which I create in my assignment.

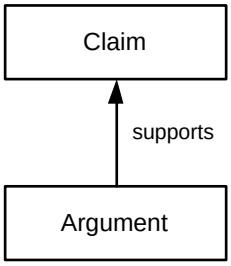

Figure 2.1: *A simple example of an argument structure.*

When we use the same approach while looking at all the arguments that are made in a an entire discussion, the structure can become far more complex than in this example. But the structure of arguments can be still be visualized using some sort of tree from in which the arguments can be organized. The form of this tree depends on the theory which is used while labelling the transcripts of the meetings (see chapters 5 up to and including 8), the used theory is also the source of the labels which appear in the tree. So an argument structure consists of the arguments that are made in the discussion and the relations between them.

The term argument structure is used a lot in this paragraph but an argument structure as such does not necessarily mean that it has to be visualized. A term that is used frequently to signal an argument structure that should be visualized is an *argument diagram* as can be seen in the following quote. "The primary tool currently in use to give an account of argument structures is the argument diagram. There are many different kinds of argument diagrams. An argument diagram generally provides a map or snapshot of the overall flow and structure of the extended chain of reasoning in a given passage of discourse containing argumentation. A typical argument diagram gives a map of the overall structure of an extended argument. The diagram is generally a graph containing a set of points or vertices joined by lines or arcs. The points (nodes) are used to represent statements and conclusions of the argument, the lines (arrows) join the points together to represent steps of inference" [Rienks et al., 2005].

The first researcher to represent the structure of argumentation by using diagrams was Beardsley [1950]. His diagrams consisted of numbered statements and arrows indicating support relationships. Coherence between various aspects of the dialogue were revealed in this way. Argument diagrams often serve as a basis for criticism and reflection of the discussion. A related term in relation to argument diagramming is design rationale, which is a systematic approach to layout the reasons for decisions that led to the design of an artifact [Shum, 1991]. According to Schum and Martin, Kanselaar et al., Yoshimi, Veerman, Buckingham Shum, Palotta et al., Reed and Rowe and Van Gelder in [Rienks et al., 2005] argument diagrams can be used

for various other purposes and these are listed below :

- Argument diagrams provide a representation leading to quicker cognitive comprehension, deeper understanding and enhances detection of weaknesses.
- Argument diagrams aid the decision making process, as an interface for communication to maintain focus, prevent redundant information and to save time.
- Argument diagrams keep record and function as organizational memory.
- The development of argument diagrams may teach critical thinking.

It is obvious that they can serve very similar functions when applied to records of meetings.

2.3 Making the argument structure useful

To be able to extract useful information from the visualized argument structure I have to make sure that the right information is present in it. For example the visualized argument structure in figure 2.1 does not contain all the information I want, although it does quite a good job when you are just interested in the structure. For example when we create an argument structure of the remarks in our sample discussion in the same way as we did to get figure 2.1 we would get the result as can be seen in figure 2.2.

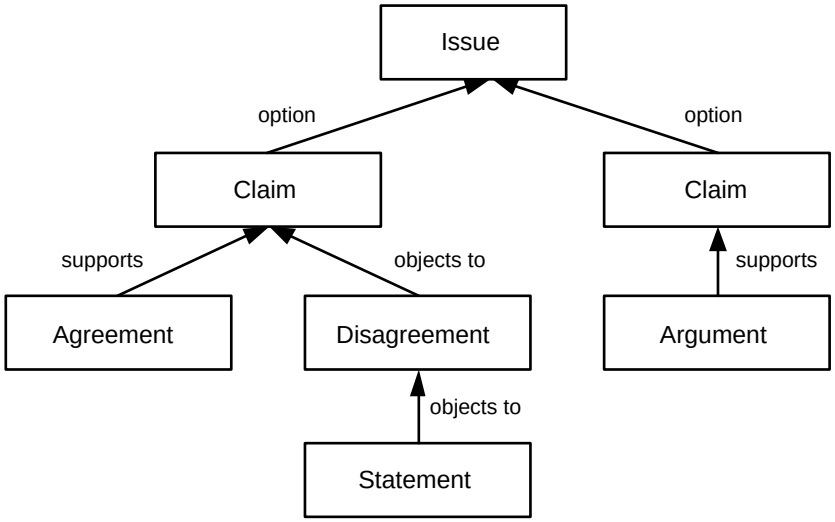

Figure 2.2: *An example of the argument structure of our sample discussion.*

But for my assignment I want it to contain a little bit more information so people who were not present at the meeting can simply see what is discussed during the meeting. One thing that has to be included for this purpose is what is said in the meeting, since people that did not attend the meeting will never know what was discussed there. I want to use the the exact words of the transcript for this purpose because there are multiple ways of interpreting remarks sometimes. For a nice visualization of the tree it could be better to show a short summary as the text of a node in the tree, and when you click the node the whole text appears for example. But this has a problem when we look at figure 2.2, the label that the node has cannot be shown any more since the text should be there. So for this reason I try to label the function of the nodes using colors, for example a node which has the function of an argument becomes green and nodes with the function of question get an orange color. The relations between the nodes do not have this problem, they can keep their labels perfectly.

It would also be nice when you could see who made which remark in the tree, since then you could look at who said something and if you for example see that one speaker makes negative remarks all of the time you could choose to pay a little less attention to those remarks than you would normally do. So to include information about the speaker we add the name of the speaker in front of the text of the node in the example. Another situation where you could decide to pay less attention than normal to a remark is when the speaker signals that he doubts his own remark by adding things like "I think" or "I guess" at the end of a remark. This information is not a part of the argument structure since the text of all the remarks is present in the tree and people have to see for themselves what they do with that information. In this manner everyone can determine their level of attention for each remark by themselves.

When we add all this information to the visualized argument structure in figure 2.2 we get the visualized argument structure which is shown in figure 2.3 (without the colors).

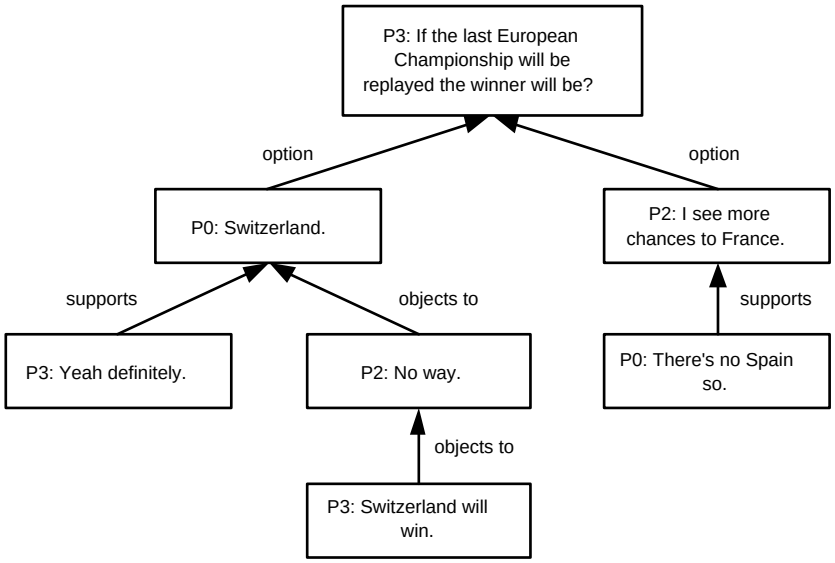

Figure 2.3: *An example of the argument structure of our sample discussion.*

In the tree depicted in figure 2.3 all the elements I need for this assignment are present. Questions of the type that is suggested in section 1.2 can be answered using this visualized argument structure. The idea of this figure is to give an idea of what the final argument structure of my assignment might look like. The labels of the relations in the figure are purely illustrations for the example and are not based on any theory or model which I use. A number of methods that are concerned with the visualizing of the argument structure are mentioned in the next paragraph.

2.3.1 Diagramming methods

Several diagramming techniques have been developed, all with their own goals in mind and their own ways of creation. We discuss three of them : Wigmore's charting method, Toulmin's model and the model developed for the EUCLID project.

Wigmore's charting method Wigmore [1931] developed a graphical method for charting legal evidence and looks like the traditional diagramming methods one encounters nowadays in logic textbooks (e.g. Govier Govier [2005]). The purpose of his charting mechanism is to represent proof of facts in evidence presented on either side of a trial, to make sense of a large body of evidence. His charts depict the arguments that can be constructed from this body of evidence as well as possible sources of doubt with respect to these arguments. In his model each arrow represents an inference or a provisional force. The nodes are the facts

or the kinds of evidence that are put forward. Each type of evidence has its own shape. Circumstantial evidence is, for example, represented by a square, whereas testimonial evidence is represented by a circle. Furthermore there are possibilities for including a type of relation between facts where one fact explains away the other, whether the evidence was offered by the defendant, or whether the fact was observed by a tribunal or judicially admitted.

The Toulmin model In the late 1950s Stephen Toulmin developed a model where a schematic representation of the procedural form of argumentation is presented [Toulmin, 1958]. Toulmins model is only concerned with pro argumentation and the acceptability of a claim, that is to say the role played by verbal elements in the argumentation during the justification process. Toulmin regards an argument as a sequence of interlinked claims or reasons that between them establish the content and force of the position for which someone is arguing. He states that an argument consists of six building blocks: A datum which is a fact or an observation, a claim related to the datum through a rule of inference which is called a warrant, a qualifier which expresses a degree of certainty of a claim, a rebuttal containing the allowed exceptions and a backing, which can be used to support a warrant.

The EUCLID Model A final model we discuss is the EUCLID model, a hypertext-like model of arguments developed under the EUCLID project. This diagramming method relies on the segmentation of a discussion into a series of claims. This model is rather simple as the resulting claims can only be related to each other by either support or refute links [Smolensky et al., 1988]. What we see is that these diagrams all serve their own purpose and show differences in application domain or level of detail. They have one thing in common: they all have their own labels and with these labels they structure parts of discourse in a way to facilitate comprehension and point out possible flaws. As our model should be able to reveal similar structures, not from evidence used in trials but from meeting transcripts, we are faced with limitations. Not all argumentation will be in favor of a particular issue, neither will all the components as defined by the Toulmin model be present.

CHAPTER 3

The theory behind the structure of texts

To be able to construct argument structures from meetings, I had to find a way to annotate transcripts of meetings. To aid me in this task there are a lot of different theories which all describe different annotation methods. In this chapter I will explain which theories I have taken a closer look at and why. After this I have given a more detailed description of the theories and finally I have explained my choice for the theories I have used to annotate some transcripts of meetings with.

3.1 Connection

My goal is to be able to visualize argument structures which are present in meetings. In these structures there are nodes and relations between these nodes and they form a tree like you can see in figure 2.3. The nodes in an argument structure may or may not have a function but if they do, the expressing power of the argument structure is greater than when they do not, since extra information can be read from the function of a node. For example when someone wants to know the reason why a decision is made, he should look for nodes with a function like 'clarification' or 'backing' and not functions like 'info-request' or 'option'. So one criteria on which I judge the theories is their ability to label the function of a fragment of text.

Another aspect on which the theories are judged is the ability to form relations between pieces of text. These relations are the relations that also appear in the argument structure. Just as with the nodes these relations may or may not have a function. In case they have function the expressing power of the argument structure is greater than when they do not have a function, in the same way as with the functions of nodes.

Because the arguments that I need are present in meetings, the theory should be able to deal with dialogues and not only monologues. As can be seen in this chapter some theories are not designed to work with dialogues, but some of them are. So this is another measure along which I rate the different theories. The rating of the theories yields the result that can be seen in figure 3.1.

In figure 3.1 is shown what the characteristics of the different theories are. When you look at Rhetorical Structure Theory for example, it is clear that it focuses on monologues and makes use of relations between fragments of text. And if you look at Dialog Acts then it is clear that they focus on dialogues and they do not incorporate any relations between fragments of text but just focus on the function of a certain fragment of text.

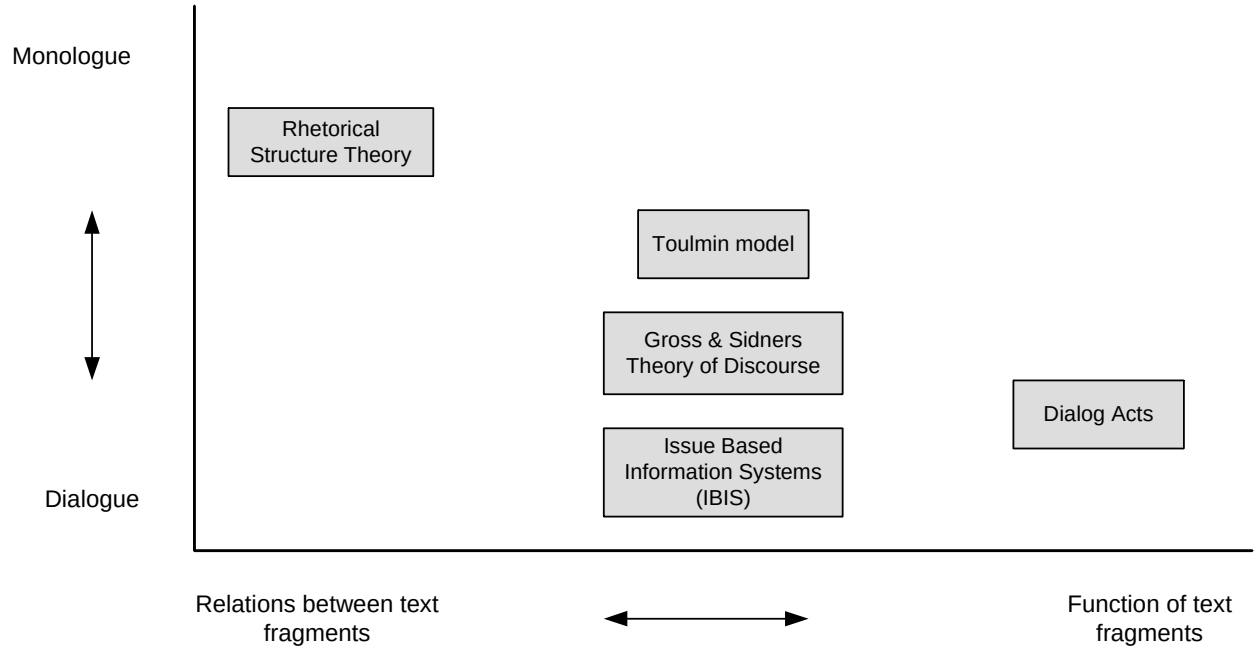

Figure 3.1: *The theories and their place according to the measuring criteria.*

There are much more theories concerning the structure of text, but these are the five theories that I have looked at in greater detail. I have chosen these five because in the argument structures consist of nodes joined together by relations. And the argument structures are taken from transcripts from meetings so the capability of the theories to deal with monologue or dialogue also has to be taken into account. As can be seen in figure 3.1 there is one theory that handles relations between fragments of text very well and that is Rhetorical Structure Theory. It has a large set of already defined relations at its disposal and if a new relation is needed it can be added, so you have limitless possibilities, it is not the best theory you could use for dialogues though. I also needed a theory which could assign labels to fragments of text. And for that purpose I have looked at Dialog Acts and they also have an advantage in being applicable in dialogues, although they lack support for establishing relations between fragments of text. Then there are three other theories that have a bit of both. They all have some possibilities to label fragments of text, and they also define some relations that can be applied between the fragments of text. In the next sections I will explain the chosen theories in greater detail. In paragraph 3.7 they will be evaluated and using this evaluation in 3.8 the theories with which I will annotate some transcripts of a meeting are chosen.

3.2 The structure of discourse

In [Grosz and Sidner, 1986] Grosz and Sidner explain their theory about the structure of discourse (which will be called Grosz and Sidner's Discourse Theory (GSDT) for convenience). Discourse structure is a composite of three interacting components:

- A linguistic structure
- An intentional structure
- An attentional state

These three components of discourse structure deal with different aspects of the utterances in a discourse. The linguistic structure is the sequence of utterances, utterances are the actual saying or writing of particular

sequences of phrases and clauses. It consists of discourse segments into which the utterances naturally aggregate, in a way which is comparable to the way that words in a single sentence form constituent phrases. Just as the words in a phrase, the utterances in a segment serve a particular role to that segment.

The intentional structure is the structure of purposes and its basic elements are provided by intentions of a particular sort and a small number of relationships between them. It captures the discourse-relevant purposes, expressed in each of the linguistic segments as well as relations between them. A property of a discourse is that it has a purpose, and sometimes even more than one. One of these purposes is seen as foundational to the discourse and this purpose is called the discourse purpose (DP). For each of the discourse segments there is a comparable purpose, the discourse segment purpose (DSP). The following are some of the types of intentions that could serve as DPs or DSPs, followed by one example of each type (from [Grosz and Sidner, 1986]).

- Intend that some agent intend to perform some physical task.
  Example: *Intend that Ruth intend to fix the flat tire.*
- Intend that some agent believe some fact.
  Example: *Intend that Ruth believe the campfire has started.*
- Intend that some agent believe that one fact supports another.
  Example: *Intend that Ruth believe the smell of smoke provides evidence that the campfire is started.*
- Intend that some agent intend to identify an object (existing physical object, imaginary object, plan, event, event sequence).
  Example: *Intend that Ruth intend to identify my bicycle.*
- Intend that some agent know some property of an object.
  Example: *Intend that Ruth know that my bicycle has a flat tire.*

The attentional state contains information about the objects, properties, relations and discourse intentions that are most salient at any given point. The distinction among these components is essential to provide an adequate explanation of such discourse phenomena as cue phrases, referring expressions and interruptions. It is an abstraction of the focus of attention of the discourse participants; it serves to summarize information from previous utterances crucial for processing subsequent ones, thus obviating the need for keeping a complete history of the discourse.

In GSDT two relations are identified which play an important role in discourse structure:

- Dominance
- Satisfaction-precedence

The dominance relation holds between two actions whenever one action satisfies an intention, say DSP1, that in its turn satisfies part of another, say DSP2. In this case it is said that DSP1 contributes to DSP2, or that DSP2 dominates DSP1 (DSP2 *DOM* DSP1). The dominance relation invokes a partial ordering on the DSPs that we will call the dominance-hierarchy. Whenever DSP1 has to be satisfied before DSP2 can be satisfied, we will say that DSP1 satisfaction-precedes DSP2 (DSP1 *SP* DSP2).

An example of how GSDT can be used is given below, this example is taken from [Grosz and Sidner, 1986]. For this example the text from figure 3.2 is used.

As can be seen in figure 3.2 the text of the movies essay is divided into eight different DSPs. For all these DSPs the intentions can be established, which are listed in table 3.1.

An based on the intentions from table 3.1 the dominance relationships can be established too. These can be found in table 3.2.

I0: (Intend ICP (Believe OCP P0))
where P0 = the proposition that parents and teachers should guard the young from overindulgence in the movies.

I1: (Intend ICP (Believe OCP P1))
where P1 = the proposition that it is time to consider the effect of movies on mind and morals.

I2: (Intend ICP (Believe OCP P2))
where P2 = the proposition that young people cannot drink in through their eyes a continuous spectacle of intense and strained activity without harmful effects.

I3: (Intend ICP (Believe OCP P3))
where P3 = the proposition that it is undeniable that great educational and ethical gains may be made through the movies.

I4: (Intend ICP (Believe OCP P4))
where P4 = the proposition that although there are gains, the total result of continuous and indiscriminate attendance at movies is harmful.

I5: (Intend ICP (Believe OCP P5))
where P5 = the proposition that the content of movies (i.e. the character of the plays) is not the best.

I6: (Intend ICP (Believe OCP P6))
where P6 = the proposition that the stories (i.e. the plays) in movies are exciting and over-emotional.

I7: (Intend ICP (Believe OCP P7))
where P7 = the proposition that movies portray strong emotion and buffoonery while neglecting the quiet and reasonable aspects of life.

Table 3.1: *The intentions of the DSPs for the Movies essay.*

I0 DOM I1
I0 DOM I2
I2 DOM I3
I2 DOM I4
I4 DOM I5
I4 DOM I6
I6 DOM I7

Table 3.2: *Dominance relationships for the DSPs for the Movies essay.*

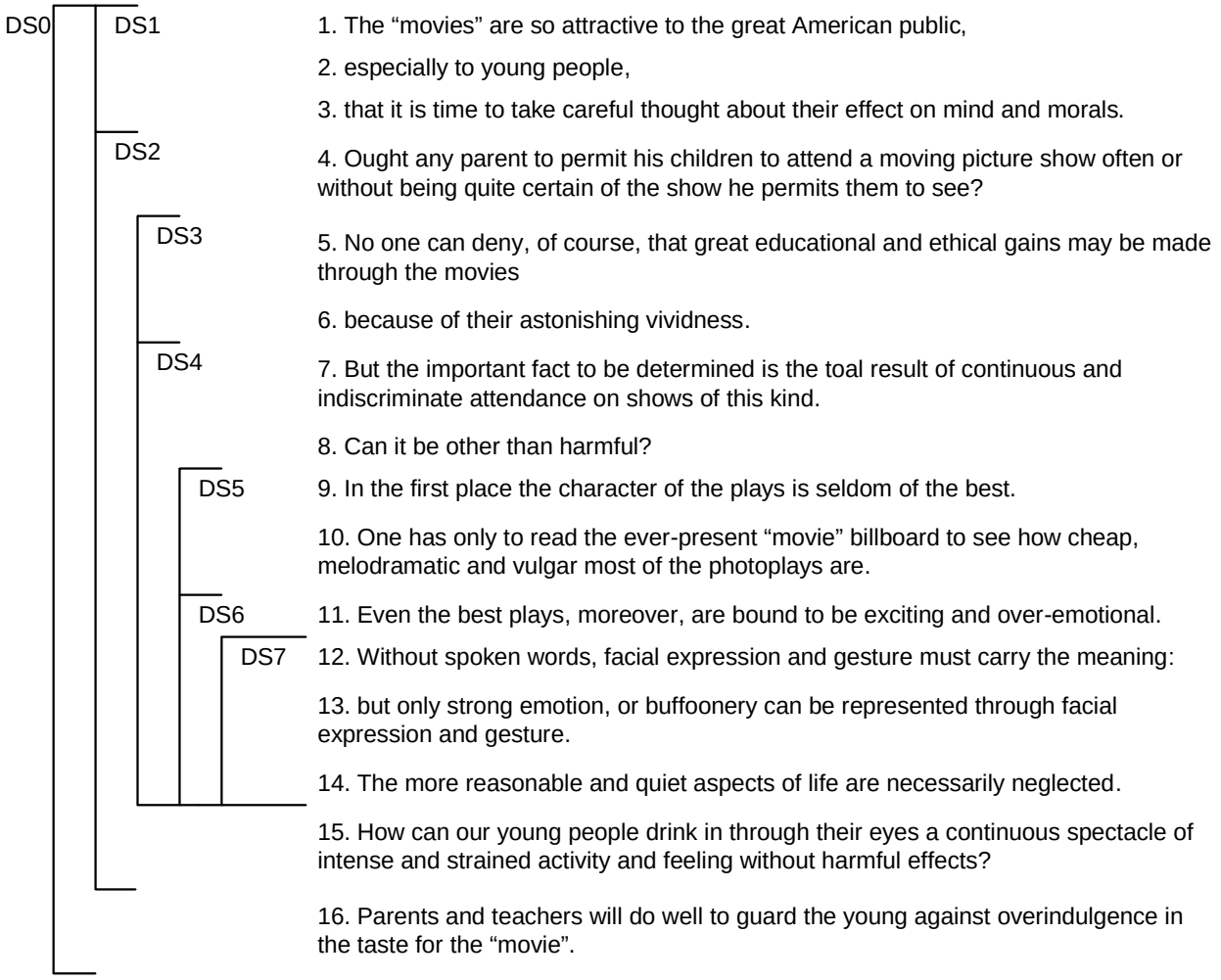

Figure 3.2: *The Movies Essay for the example use of GSDT.*

3.3 Rhetorical Structure Theory

The theory is called Rhetorical Structure Theory (RST) because it provides a framework for describing rhetorical relations among parts of a text [Mann et al., 1992]. RST describes texts in a rich and highly constrained way and thus predicts much about their character and effects. It describes functions and structures that make texts effective and comprehensible tools for human communication. RST is a theory with which you can identify functional relationships between parts of text and can be used to visualize the structure of a text [Marcu, 1997, Stent, 2000]. RST has become one of the most popular discourse theories of the last decade driven mostly by research in natural language generation [Marcu, 1998].

First RST provides a general way to describe the relations among clauses in a text, whether or not those relations are grammatically or lexically signed. Second, descriptive RST has been used as an analytical tool for a wide range of text types. Third, descriptive RST lays a foundation for studies in contrastive rhetoric. Fourth, RST has proven to be useful in analyzing narrative discourse as well. Finally, RST provides a framework for investigating Relational Propositions, which are unstated but inferred propositions that arise from the text structure in the process of interpreting texts [Mann and Thompson, 1987].

In RST four kind of objects are defined:

- Relations
- Schemas
- Schema applications
- Structures

Relations identify the relations that can hold between two parts of a text. Based on the relations, schemas define the way in which spans of text, a text span is an uninterrupted linear interval of text, can be analyzed in terms of other spans. The schema application conventions define the ways that a schema can be instantiated, somewhat more flexibly than just literal part-for-part instantiation. The notion of the structure of an entire text is defined in terms of composition of schema applications.

The RST structural descriptions of texts that can be generated all have the following property: each relation and schema definition is made from particular judgments that the text analyst must make. So all the definitions apply only when it is plausible to the text analyst that the writer (the writer of the text being described) wanted the spanned portion of the text to achieve the effect, therefore judgments about the writer must be plausibility judgments rather than judgments of certainty.

The first step that has to be taken when analyzing a text is to divide it into units. Unit size is arbitrary in RST, but the division of the text into units should be based on some theory-neutral classification. The units should have independent functional integrity to get interesting results. The next step that has to be taken is to identify the text spans and the relations between them, working either from the top down (progressive refinement) or from the bottom up (aggregation), or both as deemed convenient. With the text spans and relations defined, a following step that can be taken is the generation of RST trees from the structure of the text [Marcu, 1996]. With the help of these trees, the structure of the text can be visualized.

An example of how RST is used (from [Marcu, 1997]) is given below. Consider the following text:

> [*Although* discourse markers are ambiguous,[1]] [one can use them to build discourse trees for unrestricted texts:[2]] [this will lead to many new applications in natural language processing.[3]]

In this text the following rhetorical relations can be identified:

$$rhet\_rel(\text{CONCESSION}, 1, 2) \vee rhet\_rel(\text{CONCESSION}, 1, 3)$$
$$rhet\_rel(\text{ELABORATION}, 3, 1) \vee rhet\_rel(\text{ELABORATION}, 3, 2)$$

In this example there is also made use of the cue word *Although* to find out that we have to deal with a CONCESSION relation between satellite 1 and either 2 or 3 as nucleus. The same sort of situation can be seen with the colon, in this case we can use it as a cue to signal that there exists an ELABORATION relation between satellite 3 and nucleus 1 or 2. From these relations we can construct the tree in figure 3.3.

In the RST tree in figure 3.3 the structure of the text fragment can be seen, first utterances 1 and 2 are combined using a rhetorical relation and thus forming an element, and after this the created element and utterance 3 are joined using a rhetorical relation. Of course this is just a simple example but it should make clear what the capabilities of RST are.

3.4 Toulmin model

Toulmin [Toulmin, 1958] is known for the attack he mounted against deductive logic as it dominates as a paramount model for good argument. He has been praised for this as well as criticized, but the general consensus seems to be that he has gone too far in his attack against deductive logic. Toulmin is particularly noted for having set forth a graphical model for arguments [Hair and Lewis, 1990].

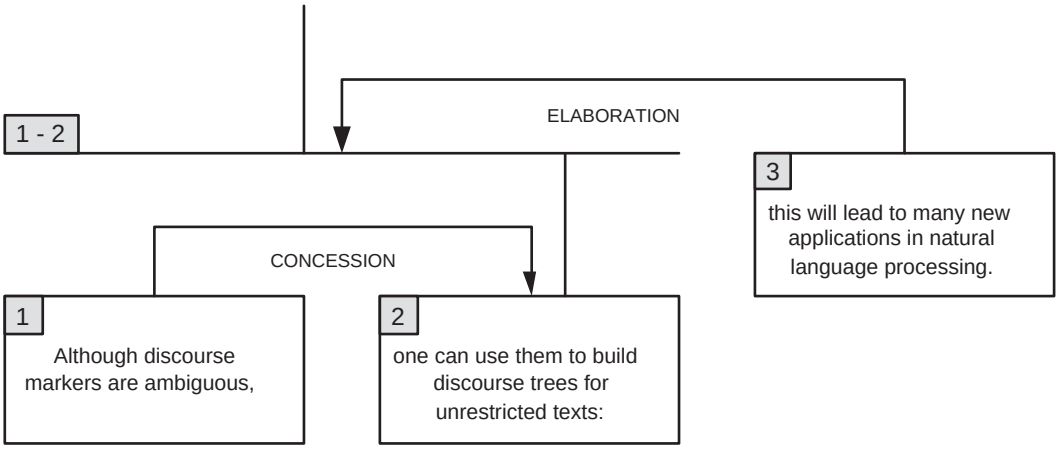

Figure 3.3: *A sample RST tree*

Toulmin's approach is one of three basic approaches to modeling arguments. The most common approach has been to base a model on some form of deductive logic. Another approach has been to propose a variety of different kinds of arguments, each with its own peculiar characteristics. Toulmin takes the third approach, which is to propose a graphic representation scheme which he asserts can be used to represent all argument types [Hair and Lewis, 1990].

In Toulmin's model [Toulmin, 1958] arguments are represented in a graphical way with the help of a schema which contains at least these three basic components:

- Data, which support
- A claim, where the support is justified by
- A warrant.

As an addition to those parts three other parts are defined: qualifiers, backings and rebuttals. Qualifiers signal the level of confidence with which the claim is made, like "probably". Backings are used to further justify warrants. And rebuttals can also occur to restrict the applicability of the claim. Each individual argument scheme that is constructed in this manner can be linked to other argument structures as a datum, claim or the like for that argument.

This model can be used to display the arguments which appear in the transcript in a way that clarifies why certain choices are made. And this is exactly what I want to do here, making the argument structure clear for other users so they can see why a design is made the way it is. Using this model also enables us to graphically display this structure in some kind of tree, an example of such a tree can be seen in figure 3.4 [Newman and Marshall, 1991].

3.5 Issue-Based Information Systems

In the early 70s Horst Rittel and Werner Kunz developed a method to be able to structure the discussion about controversial issues as can be found in politics, planning, design, development and other activities concerned with processes forming states or objects in the future [Isenmann and Reuter, 1997]. They called the instruments in this method Issue based Information Systems (IBIS), and from here I will call it the IBIS method. It is based on the principle that the design process for a complex problem is a conversation between the participants who each have their own area of expertise [Conklin and Begeman, 1988].

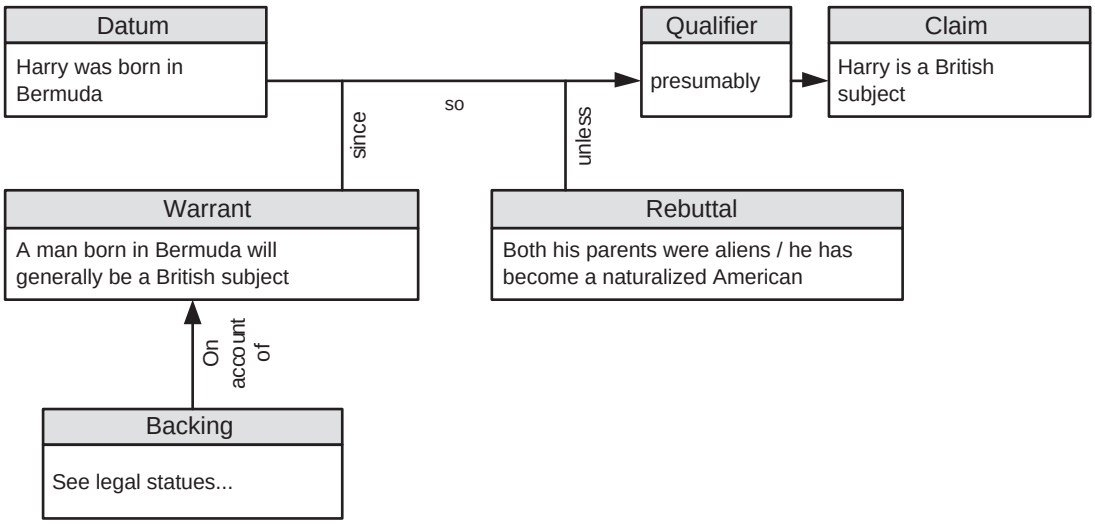

Figure 3.4: *The structure of a Toulmin tree*

IBIS is used to solve problems by using argumentative processes to apply a structure to a problem. In the process the problem is also called the *topic*. Within this topic, speakers bring up *issues* within the problem or topic. Whenever speakers have an opinion towards an issue, they can assume a *position* to state how they look at the issue. To defend their opinion towards the issue they can construct *arguments* until the issue is settled. Frequently *questions of fact* are also constructed and the answers to them can be questioned and turned into issues. In this process the participants give their opinion and judgement about the topic and thus create a more structured look of the topic and its possible solution [Conklin and Begeman, 1988, Kunz and Rittel, 1970].

In IBIS all of the elements are joined together in one structure by *relations*. Relations can be placed between all elements of IBIS and in this manner a network can be created and this network can be used to visualize the structure of the discussion or the design rationale. The following types of relations can be pointed out [Isenmann and Reuter, 1997]:

- Every argument supports or opposes at least one position
- Every position responds to at least one issue.
- Every issue is related to at least one topic.

Relations on the issue level are very important since there are a lot of them because issues are the main elements of IBIS. An issue can be related to another issue by an "is necessary for" relation for example. These relations are possible because the nature of an issue can be a lot of things like plans and sketches etc. (and there can be a lot of these in a (design) meeting).

Because we are planning to visualize the structure of the text with this method, we also have to think of a way to visualize the the elements and relations mentioned above. For this purpose I choose the method of visualizing this structure in a way like the one used in gIBIS (graphical IBIS) [Conklin and Begeman, 1988]. This system also contains a relational database server to allow users to query the information which is present in gIBIS. In order to make gIBIS work nicely, a few extensions to the IBIS model have to be made [Conklin and Begeman, 1988]:

1. An additional node of type *other* when an opinion can't be expressed in the normal IBIS framework.
2. An additional node type *external* for nodes which contain material like sketches or code.

3. The possibility of allowing positions and arguments specialize or generalize each other.

Because I want to capture and visualize the design rationale I don't know whether I will use the other type node in my agent or not, but since it is a part of the gIBIS model I will keep it in mind. With this consideration in mind the legal options which can be visualized can be seen in figure 3.5 [Conklin and Begeman, 1988]:

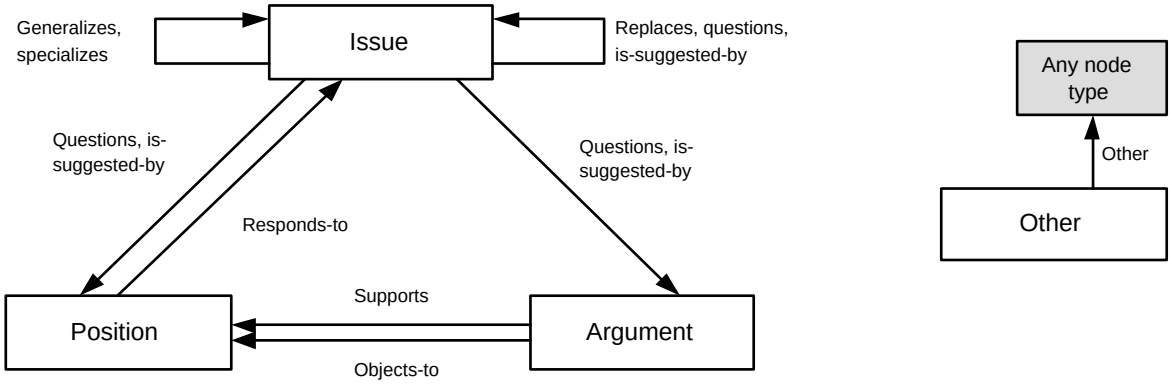

Figure 3.5: *The gIBIS model*

An example situation of the application of the gIBIS model is shown in figure 3.6. When you take a closer look at the figures of visualized argument structures in chapters 1 and 2 now, you can see that the structure there is derived from the structure of IBIS.

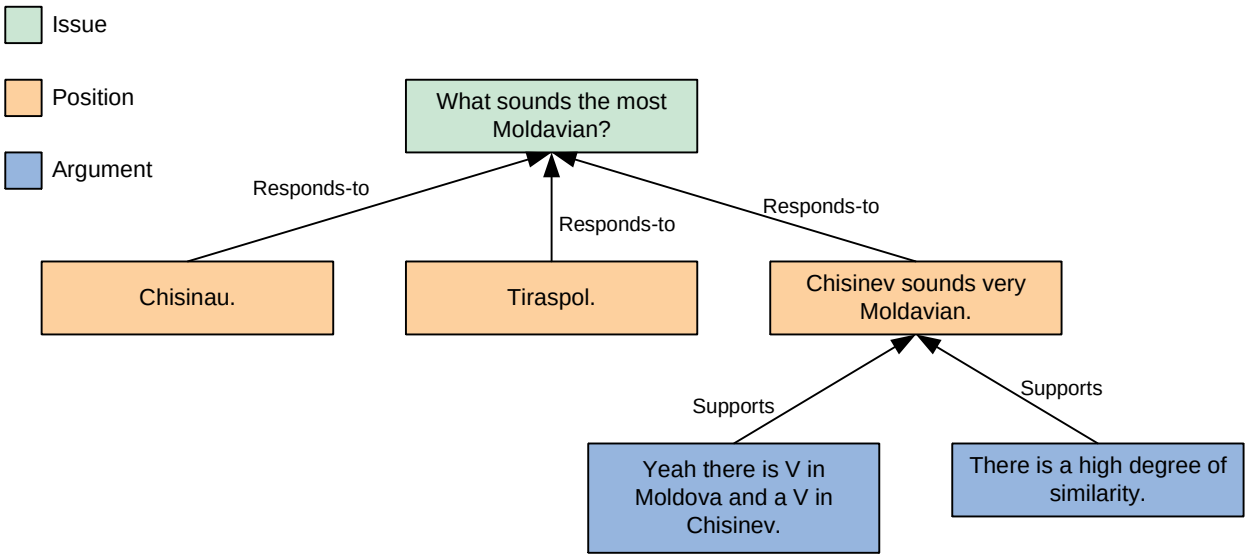

Figure 3.6: *The gIBIS model*

3.6 Dialog Acts

Since 1978 information dialogues between two people or a human and a computer have been studied. From these studies a theory of information dialogues has emerged. In this theory an information dialogue is viewed

as a sequence of communicative acts called "dialog acts" (DAs) [Bunt, 1979], and are related to the speech acts described by Searle [Searle, 1969].

A DA is defined by four components: 1. the speaker; 2. the addressee; 3. the communicative function; and 4. the ("propositional") content. Dialogue acts are viewed as operations that modify the addressee's state (goals and beliefs), the communicative function of a dialogue act being defined as the function that specifies how the addressee's new state depends on the previous state and the content of the act.

DAs are used to mark important characteristics of utterances labelling their function. This technique has been developed to be able to annotate dialogs for many different purposes. For any particular project, we would expect that the annotation scheme would be refined to provide further detail on the details we are interested in. By agreeing to this standard, however, we would gain the benefit that data from different dialogs are comparable. In table 3.3 I give the list of DAs I will use to apply a structure to the transcripts [Allen and Core, 1997].

Communicative-Status	**Uninterpretable**		
	Abandoned		
	Self-talk		
Information-Level	**Task**		
	Task-management		
	Communication-management		
	Other-level		
Forward Looking Function	Statement	**Assert**	
		Reassert	
		Other-statement	
	Influencing-addressee-future-action	**Open-option**	
		Action-directive	
	Info-request		
	Committing-speaker-future-action	**Offer**	
		Commit	
	Conventional Opening Closing		
	Explicit-performative		
	Exclamation		
	Other-forward-function		
Backward Looking Function	Agreement	**Accept**	
		Accept-part	
		Maybe	
		Reject-part	
		Reject	
		Hold	
	Understanding	**Signal-non-understanding**	
		Signal-understanding	**Acknowledge**
			Repeat-rephrase
			Completion
		Correct-misspeaking	
	Answer		
	Information-relation		

Table 3.3: *List of possible Dialog Acts (in bold) and their categories*

With this technique a role can be assigned to each (part of an) utterance but not the relations between them. For each specific task the annotation scheme can be altered to provide an optimal classification for the text which has to be structured. In this case I will begin with the set of DAs which are shown in table 3.3 but when it is necessary I might add a new or remove an existing DA.

To give an idea of how the DAs can be used to annotate a text I will give an example as described in Core and Allen [1997], which can be seen in figure 3.7. In this discussion person A asks a question and person B can give a number of possible answers to the question. And all of the answers have a different role depending on their content, and according to their role a DA is applied to label that role. The whole text can be structured this way.

```
Context:      A:   Would you like the book and its review?
Accept        B:   Yes please.
Accept-Part   B:   I'd like the book.
Maybe         B:   I'll have to think about it (intended literally).
Reject-Part   B:   I don't want the review.
Reject        B:   No thank you.
Hold          B:   Do I have to pay for them?
```

Figure 3.7: *An example of annotations with DAs*

3.7 Evaluation

RST and GSDT are strongly related partly because they share several important assumptions about the nature of the use of language and how to account for it. Although not all of these assumptions are related to this we still choose to display them here for completeness. The assumptions are [Mann and Thompson, 1987]:

- Accounting for discourse requires explicit accounts of the involvement of the speaker and hearer. Just analyzing text relative to the conventions of language is inadequate.
- The structure of discourse reflects more than anything else the intentions and goals of speakers. Intention is generally hierarchic.
- Attention and intention are usefully regarded as independent interacting aspects of texts.
- Mann and Thompson take Grosz and Sidner to be saying that language form, language function and discourse structure are related in a loosely co-constraining way, not by anything resembling one-to-one mappings. Thus there are no structural features that are always signaled uniquely by particular forms. Mann and Thompson agree.

Although the theories agree on some points, there are also differences. RST and GSDT are very different in scope, GSDT tries to cover attentional, intentional and linguistic phenomena, whereas RST primarily focuses on what Gross and Sidner would call intentional phenomena. GSDT also attempts to cover dialogue, while RST in its present form does not. There are some important differences in the way that the two theories are specified too [Mann and Thompson, 1987]:

- Most importantly, GSDT does not specify where its discourse purposes come from or how they can be constructed or verified. RST has a specific account for how particular purposes (intended effects) are assigned to text spans.
- Similarly, GSDT does not specify how discourse segmentation is done, and whether it is guided by theory or pretheoretical.

These differences make it hard when we try to predict how GSDT will account for new types of text. When the theories are compared additional differences are suggested. In natural text there can be found structural configurations, including non-interruptive discontinuous schema applications, that are allowed in RST, but not in GSDT. RST generally also produces a finer-grained account of the text, identifying goals that GSDT does not. Also RST seems to handle cue phrases better than GSDT [Mann and Thompson, 1987].

RST can be used to visualize the relations which are present in an argument structure, but the functions of the nodes in the argument structure cannot be captured using labels from this theory. RST trees are capable of visualizing the rhetorical structure of a text, so with the right labels for the relations a piece of the argument structure can be visualized, but it would not be complete.

A drawback of the visualization capabilities of the Toulmin model is that every argument has its own Toulmin tree. So when I try to model a whole discussion or a whole design process using this technique, I get a lot of loose trees since there isn't a way to connect them. Whenever a counterargument occurs I could connect the Toulmin tree of the counterargument to the tree of the argument which it opposes. Also the nesting of Toulmin trees could be of great help since sometimes a claim is based on a claim that is made earlier. According to Newman and Marshall [1991] the claim on which the new one is based can be labelled as datum in this case, but according to Shum and Hammond [1994] this is not the correct way of solving this problem (but they don't give an alternative either). I think I will choose for the option where the Toulmin trees are nested, because in the end I would like to have one tree that contains the design rationale. This method makes use of *argument chains*, *argument hierarchies*, *confluence arguments* and *connection by rebuttal* to connect the various arguments to each other to form a tree [Newman and Marshall, 1991]. Another drawback was that there were problems while trying to connect different parts of a large argument into a unified structure. Also the breaking up of argument into the datum/claim/warrant structure was somewhat clumsy and arbitrary sometimes. And Hair and Lewis [1990] use only the datum, claim and warrant parts of the Toulmin model.

The IBIS method has a lot of the properties that are needed to construct visualized argument structures. It is able to label both nodes and relations that are present in argument structures, a tree can be created to visualize this and it is quite good at dealing with dialogues. This techniques does also have its disadvantages though, the labels that can be assigned to relations that exist between nodes are static. By this I mean that you cannot place a relation anywhere in the argument structure, between two nodes of a certain type only a few (or just one) relations can be established as can be seen in figure 3.5.

The lack of relations when DAs are used to annotate the text is a serious drawback. Because of this a tree which could be used to visualize the argument structures cannot be constructed. Because RST lacks the ability to assign labels to fragments of text a combination of these two theories could be a solution to this problem, in that case the resulting tree would have the DAs as node labels and the RST labels as relation labels. This combining of the two theories would however result in a very large set of labels and this does not make the annotating process any easier.

3.8 Conclusion

I have chosen not to use GSDT to annotate a meeting because it is very good for showing intentions and dominance relations that hold between spans of text, but it is not possible to use any other relations and it has no labels that can be used to describe the fragments of text. So for the creation and visualization of argument structures this method cannot be used. The other theories mentioned in this chapter for will be used to annotate a meeting for the following reasons. Rhetorical Structure Theory, since it is the theory that has the best options for establishing relations between fragments of text. Dialog Acts because they are great for assigning functions to fragments of text. And these two goals are very important for the creation of argument structures, but a combination of the two qualities of Rhetorical Structure Theory and Dialog Acts are combined (albeit somewhat limited) in IBIS and it is able to work with dialogues, so this theory is also worth a look. And because argument structures have to be created I have also chosen to take a look at a theory which is built with arguments in mind (see 2.3.1) and this is the Toulmin model.

3.9 Further reading

In the next chapters the process of annotating a meeting using the techniques chosen here is explained. In chapter 4 I have described in detail how the meeting which is annotated looks and what some expectations of the annotation process are. The next chapters, chapters 5 through 8, show the difficulties I encountered while using the various techniques while annotating the meeting. And the results from the annotations are all put together in chapter 9. If you are not interested in the annotation details, you could skip this section and continue reading from chapter 10.

CHAPTER 4

The meetings

To be able to extract the argument structure from a meeting one has to make sure that the meetings at which will be looked contain discussions in which people have different opinions and where arguments are used. For this purpose I have selected a couple of meetings which are recorded, from which the transcripts are available and which contain some interesting discussions to annotate. This is a great advantage since I plan to use the transcripts of the meetings as the source of my data. One of these meetings, which is known as meeting 6, will be used to test the different annotation models and theories, this meeting 6 will be described in more detail in this chapter.

4.1 Meeting 6

Meeting 6 is a meeting from a series of eight meetings that are recorded for the AMI project. I have chosen this meeting for detailed analysis because it contains three discussions about different topics and from these discussions it should be possible to extract a nice argument structure. A short description of each of the topics is given below and the topics of the meeting are:

- *Moldova*
  A discussion about which city is the capital of Moldova.
- *European Championship 2004*
  A discussion about who would win the European Championship 2004 if it were to be played again.
- *Intelligence*
  A discussion about which animal is the most intelligent one.

The three discussions (one about each topic) in meeting 6 are different and therefore the arguments that are used in each discussion can also differ from each other. The first discussion is about a topic where the solution (which city is the capital of Moldova) can be checked. The city that the participants may choose is or is not the capital of Moldova, and something in between. In the discussion about the European Championship they can choose any country that enters the European Championship and their final choice for the winner can never be evaluated since that particular European Championship will never be played again. And in the discussion about intelligence the participants try to establish what is meant by intelligence and after this is

done they choose the animal that they think is the most intelligent. Whether they choose the right animal is now influenced by their definition of intelligence and therefore cannot be checked if that definition is wrong.

The transcript of this meeting consists of an *episode* (the whole meeting), an episode is divided into *sections* and sections contain various *turns*. Each *section* from the transcript deals with a topic from the list of topics in the meeting. Although the transcript contains four *sections*, one for each topic and one for the beginning of the meeting. In the beginning of the meeting the participants ask each other whether they are ready to start the meeting and this part of the meeting is not looked into any further. This meeting is held with four participants and in the remainder of this chapter situations where someone changes its opinion or convinces someone of their position on the topic will be pointed out. These situations will be mentioned in the order in which they occur in the discussions where they appear. These situations are mentioned because these are the situations that people should be able to read and comprehend when they see the visualized argument structure of the discussion. In this chapter only the situations are examined and not the way in which they could be visualized.

4.1.1 Moldova

For the answer to the question which city is the capital of Moldova a number of possible answers are given. None of the participants have a favorite though, they all do not know which city is the capital of Moldova. And at the end of the meeting they cannot agree on one of the answers and they continue to the next topic of the meeting without giving an answer to the question. Therefore no changes of opinion are present in this discussion.

4.1.2 European Championship 2004

In this discussion a few options of possible answers are given. Right in the beginning of this discussion three speakers say which country is their favorite, p0 and p3 think it would be Switzerland and p2 disagrees and thinks it would be France. A lot further down the discussion p1 states his favorite for winning the championship is Greece. But these opinions need not necessarily be the opinions of the participants at the end of the meeting and in this case it is not. At the end of the discussion all of the participants agree that Greece would win the tournament if it is played again.

So the opinions of speakers p0, p2 and p3 have to be changed in the discussion. The first speaker to change his mind is p3. A little bit further in the conversation p3 says this:

> *"But but if-f-f if if it will be replayed, then nothing change I would say Greece"*

P2 does not agree with this and sticks with his choice during the entire discussion and gives a number of arguments to backup his opinion too. But in the end of the discussion p2 is the next speaker who changes his opinion in a way that he also agrees that Greece would win. So only p0 and p1 have to agree now, and directly after p2, p1 also says he thinks that Greece could win the tournament. And in this way they eventually they reach a conclusion that Greece would be the winner again, to which p0 does not object so he must also have changed his mind about it as can be seen in the following piece of text:

p2: *"So it's Greece has more chances"*
p1: *"I think so"*
p2: *"Um"*
p3: *"In absence of of Spain,"*
p3: *"Greece has more chances"*
p1: *"R-ight right right now?"*
p1: *"??"*
p0: *"That's very diplomatic"*
p2: *"In absence of Spain Greece has more chances yeah"*
p3: *"That's very Swiss."*
p3: *"Ok so we say Greece"*
p2: *"Yeah"*

4.1.3 Intelligence

In this part of the meeting the participants have to decide which animal is the most intelligent. Again a few options are given from which the participants have to choose. And p2 begins the discussion directly by stating that ants are the most intelligent ones of the three animals. P0 disagrees but does not say what animal he thinks is the smartest one and in this reaction p0 brings up a new discussion about what intelligence is precisely. P3 says that he does not think a cow is the most intelligent one, but he does not say anything about his favorite for the most intelligent animal, and neither does p1.

After the discussion about what they think intelligence should be here, each participant gives his favorites again. P2 starts and he gives the order in which he would rate the animals from most intelligent to least intelligent and his rating is ants, cats and finally cows. P3 disagrees and gives his list cats, cows and ants. After this p0 says that he thinks ants are the most intelligent animals as can be seen in the text fragment below:

p2: *"So the mother um I would rate it as ants cats,"*
p2: *"ants cats and cows"*
...
p3: *"I would rate cats, cow, ants"*
p0: *"I would say ants."*
p2: *"Ants yeah"*

P1 still has not said anything during the discussion and therefore his opinion about which animal is the most intelligent one remains unknown. From this point up until the end of the discussion the participants argue which animal is the smartest of the two options that are the favorites now: cats and ants. P2 and p0 defend their opinion that ants are the smartest, and p3 defends his opinion about cats, an example of this is shown below:

> *"Yeah but is-t it can be a proof of intelligence if they can um they can have um critique opinion against other cats, where as ants just agree, so they don't really"*

And now p1 starts to take a part in the discussion too but he does not choose between ants or cats, he defends both options as can be seen below:

> *"I-f if there's something, an ant will eventually find it"*

> *"What do you mean by modifying the environment? If you put a cat in an environment with a a lot of rats, it will change the ??"*

In his first remark he states that if there is some food to be found anywhere, an ant will eventually find it and that this can be seen as a proof of intelligence. In the second remark he says that not only ants can modify the environment by building ant hills for example but that cats are capable of modifying the environment too.

At the end of the discussion the participants end the discussion and choose the animal that is the most intelligent in the way that can be seen below:

p2: *"We we I vote for the ant as ??"*
p0: *"Yeah same"*
p3: *"I vote for ant as well"*
p2: *"Ok so we finalize that ants are the intelligent ones compared to the given list"*
p1: *"Me too"*

P2 begins by saying that he thinks ants are the most intelligent and this is logical since p2 has thought this throughout the entire meeting. P0 agrees with this which is not a surprise also, but p1 has not said which animal he thinks is the most intelligent one and p3 thinks cats are the most intelligent ones. But as can be seen in the text above p3 votes for ants at the end as most intelligent animal so his mind has changed. And p1 also agrees with the ants, so the ants are chosen as the most intelligent animal.

4.2 Evaluation & Conclusion

As is made clear in this chapter meeting 6 contains interesting discussions because participants try to convince each other of their points of view. In these discussions arguments why they think they are right are used and these can be included in the visualized argument structures very well. Also a lot of situations are present in the meetings where participants agree or disagree with each other. Also situations where they are not sure appear a lot (especially in the discussion about Moldova). Therefore this meeting can be used very well to test the various annotating methods.

CHAPTER 5

Rhetorical Structure Theory

In this chapter I will try to keep track of the things I do while creating Rhetorical Structure trees (RSTTrees) from the transcript of meeting 6. Whenever I think to have found a pattern in the relation between the transcript and the trees this will be mentioned also. For each of the three sections of the meeting I have constructed a Rhetorical Structure Tree [Marcu, 1996].

During the annotating I will try to follow the official approach of creating RSTTrees [Stent, 2000, Marcu, 1997]. I will define the RSEs first by defining the minimal text units or utterances and giving them the function of an RSE. A speaker turn is split into parts wherever there is a comma or full stop or another form of punctuation. Also an utterance cannot exceed the limits of the speaker turn, thus an utterance ends whenever the speaker turn in which it originates ends. When all the RSEs are determined I will try to establish relations between them. The relations I will use are all Rhetorical Structure Relations (RSRs), and I took the list of RSRs that is also used the RSTTool [O'Donnel, 2003].

5.1 Moldova

The first thing I have to do here is to load the text in the tool, and when this is done I can start marking the minimal text units. I choose to mark every minimal text unit in the transcript as an RSE for this is the way it is normally done in Rhetorical Structure Theory. For the same reason I will try to establish as many RSRs between the RSEs as possible. Not all of these RSRs might be necessary for the agent which (in the end) should be able to construct the design rationale from them. And it's also better to have too much RSRs than not having enough of them.

One thing that draws attention while annotating the transcript is that I often cannot find the relation I am looking for in the set of relations I use with the tool. As can be seen in table 5.1 a lot of relations between RSEs get the label OTHERWISE in this manner, and as the name of the relation suggests this is not a very likeable option. Also the label QUESTION-ANSWER-S appears a lot in the file with RSRs, and I would like to be able to specify whether the answer to the question is positive or negative or even something different. This can be solved by using an extended set of RSRs from the RSTTool, but I am also thinkig of making my own set of relations. Because in this set of relations I can put exactly the existing ones that I need and I can also try to define additional ones.

Sometimes it was the case that there obviously was a link between two utterances, but there was no relation

to show this link. In the next few examples I will point out which relation types I would like to have had. See for example the following utterances:

> *"Moldova sounds like ting ting"*
> *"Yeah"*

The relation between these two text units currently has the label OTHERWISE but I thought that this relation could get a label like *agreement* or *confirmation*. Since this type of relation is quite common in this (part of the) meeting I expect this new relation could be used in more cases too.

Another problem was that sometimes a speaker repeated something that had been said earlier, but now as a question to signal that he does not agree with what had been said earlier. This can be seen in the following example:

> *"Chisinau"*
> *"?? Chisinau"*

In this case it could come in handy to have a relation with which you can express that there is disagreement among the participants. It could also be that doubt is behind the second utterance and that with further explanation they can reach agreement, and in this case it would not be correct to label it as disagreement. So a relation in this category like *disagreement* or *doubt* (or even both) also seems useful to me.

In table 5.1 the number of RSRs in the part of the transcript over Moldova are listed.

RSR type	**Occurrences**
QUESTION-ANSWER-S	13
OTHERWISE	11
CONCESSION	2
ELABORATION	2
INTERPRETATION	1
EVALUATION	1
RESTATEMENT	1
ANTITHESIS	1
ATTRIBUTION	1
JUSTIFICATION	1

Table 5.1: *The type and number of RSR occurrences in the discussion about Moldova*

5.2 European Championship 2004

In the process of marking the RSEs in this part of the transcript I found that I sometimes missed punctuation in the transcript. While I was splitting the text into smaller chunks by hand I still chose to do as if the punctuation was there. You could clearly see that the text should have been split when you look at the structure of the sentence and the meaning of the text, although this is not officially according to the Rhetorical Structure Theory. I will give an example of the described problem, see the next sentence:

> *"Even more well Portugal was at home and so that's also a good helping factor so they couldn't win Greece in this condition I would say that Greece will win again."*

According to the official method [Mann and Thompson, 1987] this sentence is one RSE as a whole, but this clearly is not the best option. It is quite clear that when I split the sentence in multiple parts, much more

information could be extracted from it than when I treat it as just one RSE. So I chose to add the following punctuation and to split it into the following RSEs:

Even more well Portugal was at home,
and so that's also a good helping factor,
so they couldn't win Greece in this condition,
I would say that Greece will win again.

When I look at the sentence in this manner I can for example argue that there exists an ELABORATION relation between the last RSE and the first three RSEs of the sentence.

During the assigning of the RSRs between the RSEs I found it difficult to determine which relations I had to assign in the following piece of the discussion:

"Switzerland will win"
"I see more chances to France"
"There is no Spain so"

In this piece of text three different speakers give their opinion on who they think could win the European Championship. The first one clearly thinks Switzerland will win and the second one believes it will be France. So far so good, but the third person only says that Spain is not in the tournament. What he could mean by this is that he thinks that Spain will win the tournament but it is not in the tournament so it cannot possibly win it. This is what I think too and so I give the relation the label ANTITHESIS or ELABORATION. But he could also mean that since Spain is not in the tournament that it will be easier for France and / or Switzerland to win the tournament, and in this case the label of the relation could be RESTATEMENT or ELABORATION. Since in both cases the label of the relation could be ELABORATION I will define the relation to be of this type.

I also came across another relation that I found difficult to give a type. This was the RSR between the following text units:

"so we will change the strategy against Greece but they lost again so"
"Yeah that's a good point for Greece."

In this situation you could say that the fact that Portugal lost against Greece is a good point for Greece. In this situation the relation would have the first utterance as target explaining what the good point for Greece is (the source). In this case the relation would be of type ELABORATION. There is also another way to interpret this relation where you use that it is a good point for Greece as support for the fact that Portugal did not win from Greece. In this case the first utterance will be the source and the second one will be the target. Unfortunately there is not a good relation to express this with so in this case the label of the relation would be OTHERWISE. Because I want to use a few OTHERWISE relations as possible I will label the relation as described in the first case above.

In this part of the transcript I also came across some situations in which I missed a certain relationship, and thus now have the label OTHERWISE. Since I would like to prevent the use of this label as much as possible I will give a few examples. The issues mentioned in section 5.1 also appear in this part of the transcript. Relations to signal that attendants of the meeting agree or disagree on something could also be very useful here.

A new relation that could be useful here but which I could not find in the set of relations from the RSTTool [O'Donnel, 2003] is a relation that can be used to mark situations in which an alternative is presented. Examples of this can be seen in the following utterances:

"Now they still have a good team"
"Yeah but Greece um we don't have a much better team than U_S_S_ but"

In this situation the alternative is used to give an alternative answer to the question that is discussed. The first speaker says that France still has a great team and the second speaker says that Greece also has a good team (at least that is what I think he is trying to say). When an alternative is used in this manner it could be useful to have a label like *alternative*. An alternative answer to the question can also be used in the following manner:

"Greece doesn't have big players,"
"you know like France has,"

In the fragment of the transcript above the alternative is used to make a comparison. The players of Greece and France are compared here and this could be important information in the reason why one of these two countries could win the European Championship. In this case a label like *comparison* could be of some use.

In this discussion there are also some other relations between RSEs that currently have the label OTHERWISE but for which I do not (yet) know a suitable relation type. But I think that for an optimal performance of the agent I must try to get rid of all the OTHERWISE relations.

An overview of the RSRs I identified in the part of the transcript about the European Championship can be found in table 5.2.

RSR type	Occurrences
OTHERWISE	25
ELABORATION	11
QUESTION-ANSWER-S	5
JUSTIFICATION	4
RESTATEMENT	4
CONCESSION	2
CONDITION	2
MOTIVATION	2
ANTITHESIS	1
ATTRIBUTION	1
EVALUATION	1
SOLUTIONHOOD	1

Table 5.2: *The type and number of RSR occurrences in the discussion about the European Championship*

5.3 Intelligence

While annotating the RSEs in the transcript I encountered the opposite situation as with the section about the European Championship. Here I found that there was too much punctuation, see for example the next sentence:

"I would rate cats, cows, ants"

According to the official Rhetorical Structure Theory rules this sentence exists of three RSEs. I marked it as one RSE though because the speaker just wants to say how he would rate the animals in order of intelligence. So it would not have much use to split this sentence into multiple RSEs.

A little bit further in the text of the transcript I encountered the problem of the missing punctuation again. But in the following sentence you could argue that you could split this sentence into two RSEs because of the cue word "but":

"As a group yeah but that isn't really intelligence that's organization"

Because this sentence is quite important for determining the structure of the text, or the structure of the arguments in the text I choose to divide this sentence in two RSEs with the separation point just before the word "but".

During the process of labelling the RSRs in this piece of the transcript I encountered the following utterances:

"Ants are the most intelligent animals in the world."
"?? cats"
"Yeah but there's an S_ {laugh}."

The last utterance can be aimed at each of the first two utterances. I could not tell for sure with which one I should create a relation, so I created a relation between the last utterance and each one of the first two utterances. I thought this was the best option since it is true that in both utterances the animal is referred to in plural. The type of the relation is a different story though, since the type I was looking for was not available the relations are labelled OTHERWISE.

Another problem I ran into was the following pair of utterances:

"I would say ants."
"You would say ants first"

After the second utterance there is no question mark, so at first sight it is no question. But when I looked at the meeting video and at the form of the discussion, I found that this in fact was a question. The speaker of the first utterance even answers to this question. If this had not been a question the RSR would be of the type RESTATEMENT, but now it is a question which expresses a feeling of doubt from the second speaker, and as there is no RSR type for this situation it gets the label OTHERWISE.

Apart from the relations that I would like to have from section 5.2 and section 5.1 I also would like to have a relation to be able to label the relation between the following utterances:

"Well it's a species,"
"I would say the most intelligent animal is in singular"

The relation between these two utterances could be of the type *disagreement* but I do not think that this is the right option. I think the second utterance is an explanation why the speaker disagrees with the fact that intelligence can be seen as a property of a species. I have also considered labelling this relation as a ELABORATION but that also did not feel right since with an elaboration I would expect a remark about why it is a species or something like that. The relation here is more like an explanation of why the speaker thinks there is a problem. Another example of this relation could be between the two following utterances:

"Ants are the most intelligent animals in the world."
"Yeah but there's an S_ {laugh}."

This relation is not a disagreement, the second speaker does not say that an ant is not the most intelligent animal but he thinks that the problem lies in the fact that you should consider individuals in stead of species. I cannot think of al label for this relation but I think it could be something like *problem-explanation* because the speaker explains a problem in the discussion.

In the discussion about intelligence at a certain moment one speaker says that ants are not very intelligent individuals because they do not have a critique opinion against other ants. And that this is a reason why ants are not very intelligent, but a second person thinks they are intelligent and gives his reason. The fragment of the text is shown below:

"so they don't really"
"Actually an interesting point is that ants have survise/survived o-n on the earth for millions of years without evolution"

So in these two utterances there is a little bit of disagreement between the two speakers. But it is not really disagreement because the second speaker does not say that ants are intelligent individuals but he says that ants have survived a long time. From this statement the other attendants of the meeting can decide whether this is intelligence or not. Since I cannot think of a label for this type of relation between two RSEs except for the making of a statement in case of the second RSE, I am not sure if this relation should even be a relation. But the speaker does not say this at this moment for no reason so he must have thought it was relevant, and that is also the reason why I thought there exists a relation between these text units.

Another thing that I noticed was that speakers often say something and in their remark they also state that they are not sure of it, you can see an example below:

"well I don't know here,"
"but in India the cows usually have a tendency to go into an some others field to eat the green grass if it doesn't gets it."

Here in the first utterance the speaker says that he is not sure how it is here, but that he is sure how it is in India. So he tries to say that this could be the case here also but he is not sure, and to signal this I could use a relation like *doubt*. You could even say that this is a comparison between India and "here" and you could also label the relation between these two utterances as *comparison*. But both of these relation types do not exist in this set of relations and thus the relation has to be labeled as an OTHERWISE relation here.

The results of the identifying of the RSRs in the part of the transcript about which animal is the most intelligent can be found in table 5.3. As you can see in this table the top 5 relations are almost exactly in the same order as the top 5 relations in the part about intelligence in table 5.2.

RSR type	Occurrences
OTHERWISE	33
QUESTION-ANSWER-S	12
ELABORATION	9
JUSTIFICATION	5
RESTATEMENT	5
CONCESSION	4
CONDITION	4
ANTITHESIS	3
EVALUATION	2
SOLUTIONHOOD	2
INTERPRETATION	1
MOTIVATION	1
NONVOLITIONAL CAUSE	1
NONVOLITIONAL RESULT	1

Table 5.3: *The type and number of RSR occurrences in the discussion about intelligence*

5.4 Evaluation & Conclusion

The advantages of this annotating method are that it supports the creating of trees that contain the structure of the annotated text. Another good point of this method is that adding al the relations you might need is allowed (although this good point also has a downside as will be explained below).

Disadvantages of this method are that it does not support a way in which the RSEs (the nodes of the argument structure) can be labeled. Another drawback of this method is that when you add all the relations that are needed during the annotating process, the resulting set of relations becomes very large and this makes the job of the annotator more complicated. Also the RSR type OTHERWISE appears a lot in the Rhetorical Structure Trees, and this is not very nice. I missed relation types like *disagreement* or *support*, so these types might have to be added to get better results. Also I would like to add two variants on the QUESTION-ANSWER-S relation to signal whether an answer to the question is positive or negative because it seems very useful to me for trying to capture the structure of a discussion. Sometimes there was also a problem in determining between which RSEs a relation should be placed. Another problem which occurred was that sometimes punctuation was missing from the transcript although an utterance from the transcript should clearly be split into multiple RSEs. And last but not least the theory can handle a lot of situations, just not the situations where typical dialogue properties appear like people agreeing or disagreeing with each other.

When you want to create trees that visualize the structure of the text and you do not care about labeling the fragments of text this method can be used. But in this assignment we do care about the labeling of the fragments of text because these are the nodes in our argument structure. So this method is not applicable in this situation.

CHAPTER 6

The Toulmin model

I will explain the process of creating Toulmin trees [Shum and Hammond, 1994] from the transcripts of the meeting in this chapter. I will do this manually and try to capture the relevant actions during the creation of the Toulmin trees. When there are patterns in my approach I will try to describe them here too. An example of a Toulmin tree is depicted in figure 3.4.

In the annotating process I will stick to the original Toulmin structure because adding more complexity, like nested Toulmin trees and my own attributes or even my own tree, can be done at a later time. Another reason is that I do not know which problems I might encounter and what the best way is to solve these problems.

6.1 Moldova

In this section I will explain the manner in which I have created Toulmin trees from the first part of the transcript which has "Moldova" as subject. The first thing I have done here is to filter the empty speaker turns out of the text, I use speaker turns because the speaker turns are already marked in the transcripts, and with empty speaker turns I mean speaker turns that consist merely of (..). After that I have removed the speaker turns which only consist of laughter, like "hehehe" or "haha". Also speaker turns which do not have any meaning were removed, the first speaker turn of this section for example was "wragh" which is not an existing word, so it has no meaning.

In the text of the transcript I found only one clear case of a claim and a datum joined together by the cue word "so". It was in the following sentence:

> *"Yeah, there is a V in moldova and a V in Chisinev, so.. There is high degree of similarity"*

The rest of the claims that I extracted from the transcript were based on the content of the utterances. Because of the nature of the discussion about Moldova there were not many claims in the discussion. None of the attendants of the meeting knew what the capital of Moldova was, and they all started guessing. This guessing can also be noticed in the transcript as can be seen in the following utterance:

> *"I would say Chisinau, I don't know"*

This utterance makes the claim that Chisinau is the capital of Moldova, but in the same sentence the speaker says that he is no t sure of this claim (a rebuttal to the claim). So from the structure of this utterance you could say that it is not even a claim, but then again why would the speaker make say this when he does not want to make a claim? Apart from this the utterance starts with the words "I would say" and this is a nice cue phrase for recognizing claims, but since I am not sure whether it is a claim this cue cannot be defined with certainty. Another problem which also occurred was that the participants of the meeting made a lot of remarks which were not serious so those could not really be counted as claims.

While creating the Toulmin trees for this section I missed a relation to be able to mark an utterance as a support for another utterance, but this could also be solved by nesting the Toulmin trees. And here also relations for agreement and disagreement could prove to be useful. But when introducing all these new relation types it could be useful to look at other representations of the structure of the text, or even designing my own schema of relations. But I have to be careful here that I do not look too much at the input for the agent, but at the visualizing of the structure of the text or the design rationale. For the creation of the structure of a text it could also be useful to have some sort of way to visualize the question answer structure and this also is not possible with the Toulmin model.

6.2 European championship 2004

The first thing I have done here is (again) remove the empty or meaningless utterances as described in section 6.1. Using the same approach as in the Moldova case I again split the text of the transcript into Toulimn trees using the meaning of the text. I did however find an example of a warrant, which could be detected by the cue words "if-then". It occurred in the following utterance:

> *"But i.i..if it will be replayed then nothing change"*

The claim in this Toulmin tree is that Greece would win the tournament again if it would be played again. And the speaker gives this warrant for his claim. Because this is not based on a fact it did not think it was a datum and therefore I chose to classify it as a warrant.

Something else that occurs frequently is that sentences are not finished, because the meeting participants do not need to hear the end of the sentence to know what the speaker means. This could also have another cause, when the speaker is not entirely sure of what he says he sometimes does not finish his sentence. While constructing the Toulmin trees manually this was no problem because most of the time I also knew what the speakers meant, but how to place this in a Toulmin tree was difficult for me too. Mainly because there are a lot of claims that are made in this manner and none of them have a direct warrant or datum, at least not one that is mentioned. So a lot of the fields of the Toulmin tree remain empty and to solve this it could be a possibility to try and create nested Toulmin trees. In this manner a previous utterance can be seen as a warrant for a claim that is made at this moment. Another option is to construct my own schema or tree to capture the structure of the discussion in. Also the remarks made in section 6.1 about missing relations apply for this part of the transcript.

6.3 Intelligence

Here I look at the last part of the transcript in which a discussion is held about which animal is the most intelligent. The participants try to reach consensus by rating the animals from the most intelligent one to the least intelligent one. The animals from which they can choose are: cats, cows and ants. So one possible solution from a participant who thinks that cows are the smartest, followed by ants and that ants are smarter than cats would be: cows, ants, cats. At the end of thee discussion they look which animal occurs the most at the top of the list and they come to the conclusion that his animal must be the most intelligent one.

Again I use the same approach as with the two other parts of the transcript, so the first thing I did was delete the meaningless utterances (see section 6.1) from the text in the transcript. And again I noticed that with this part of the transcript I used the meaning of the utterances a lot more than the cue words to put the utterances at their correct place in the Toulmin trees. But I also found a couple of word combinations which signal a claim, these are word combinations like "I think" and "I would say".

While constructing the Toulmin trees I ran into a problem right at the beginning of the transcript. Two participants disagree on which animal is smartest, one participant says that ants are the smartest animals and another one says that this might be true when they are in a group but it is not true when you look at the individuals. The fragment of the transcript where this occurs is this:

> *"Ants are the most intelligent animals in the world."*
> *"Well, taken as a whole maybe, but individually, no."*

When I just classify these utterances according to the Toulmin model it would be two claims, but anyone can see that there exists more of a relation between these remarks than just two claims. A possible way of combining these two utterances into one tree is to classify the first utterance as a claim and the second one as a rebuttal, but this does not feel quite right. So again the option of nesting the Toulmin trees comes to mind. The first utterance can then be seen as a claim and the second one can be seen as a claim too in its own Toulmin tree. But the second Toulmin tree can be used as a whole to serve as a rebuttal in the Toulmin tree of the first utterance.

Another problem that occurred was that claims are often supported by opinions of participants which are also claims. An example of this can be seen in the following part of the transcript:

> *"I think we can eliminate Cow anyway."*
> *"It doesn't look very intelligent."*

This could be solved using nested Toulmin trees, by letting the datum for the claim be a separate Toulmin tree in which the utterance has the function of a claim. In this manner it could be pointed out that the support for a claim is not based on a fact but on another claim.

I also found another issue in the text of the transcript. The speaker relies on the common sense of the other participants while he tries to express his opinion on the matter. See for example the following (edited) piece of the transcript in which a speaker tries to explain why he thinks that cats are more intelligent than ants:

> *"it can be a proof of intelligence, if they can ehh they can have a. critique opinion against other cats, where as ants just agree, so they don't really"*

In this sentence the speaker claims that ants just agree and do not have a critique opinion against other ants. But this claim is not expressed literally but all of the other participants know what the speaker means (it does not matter whether they agree with it or not). So while creating the Toulmin trees manually I could extract this information from the text and construct a Toulmin tree with what the speaker means, but here I try to create a tree which contains what is in the transcript. Also this total utterance could be seen as the claim that cats are smarter than ants, but it could also be separated in claims like "having a critique opinion can be proof of intelligence". On what the speaker bases this claim is not mentioned in the transcript and I too could not come up with an explanation. In the XML file I have created of this transcript I have chosen to label the whole utterance as a claim although there is clearly much more information in it than meets the eye.

During the meeting (and thus in the text of the transcript) there occurs a lot of referring to remarks that have been made earlier during the meeting. This cannot be modelled in a very nice manner without the use of nested Toulmin trees, because without them you get a lot of information which occurs as a claim in one tree and has another function in another tree. This phenomenon can be compared with the problem of claims supporting claims. And also in this part of the transcript I missed the same relations as in the first two parts of the transcript (see section 6.1).

6.4 Evaluation & Conclusion

The advantage of the Toulmin model is that the structure of an argument can be described in great detail. Another nice property of this method is that argument trees can be constructed when the Toulmin trees are nested instead of a number of individual Toulmin trees.

Disadvantages are that a lot of space in the Toulmin trees remains empty in the Toulmin trees because speakers rely on the common sense of the other participants, and do not spell out each rationale they follow. The other attendants understand what they mean without it, and the people who read the information from the Toulmin trees also understand what has been said during the meeting without this information. But the visualization of trees with a lot of empty spots in them does not help to achieve a nice way of visualizing the argument structure because it disturbs the clarity of the visualization. Why do I not leave the empty spaces out of the tree then? Because the empty spaces are needed for the Toulmin model to make sense I cannot do this.

So when you want to get a deeper understanding of the structure of each argument, the Toulmin model is a very good choice of model but for the purpose of this assignment the Toulmin model is not the right choice.

CHAPTER 7

Dialog Acts

Since RST alone could not provide the structure of the text in such a way that it could be used for this assignment (see section 5.4), the transcript of the meeting is annotated with the help of DAs (see table 3.3) this time. Just as with the annotation of RSR and the Toulmin model I will again show in detail what issues I encounter during annotation of each topic inidividually.

7.1 Moldova

The first topic of the meeting is which city is the capital of Moldova. In this paragraph I will try to show the most problems I ran into while annotating the transcript of this part of the meeting. Right in the beginning I find the following two utterances:

> *"It's pretty warm here"*
> *"Ok."*

I have chosen to model the second utterance as a DA of type *hold*, but this could also be *accept*. From the text I cannot determine which of the two things is the most appropriate, but I choose for the first option since the second speaker does not explicitly say that he agrees with the first speaker (the utterances are made by different speakers). So I think he says "Ok" to signal to the other speaker that he has heard his remark, and in that case the DA *hold* can be used.

A little bit further down the conversation p0 makes a remark which contains multiple RSEs:

> *"What what what sounds the most Moldavian? Chisinev? Chisinau"*

This could cause for problems later however since the purpose of the labelling of the text with DA types is the extraction of additional information about the meeting. I think it is important to remove the RSRs of type OTHERWISE since they do not give much information about what goes on in the meeting, and the way to do that could be to apply DAs to the RSEs which are related by an RSR (of all types, but especially OTHERWISE). But the utterance of p0 is composed of three RSEs "What sounds the most Moldavian?", "Chisinev?" and "Chisinau". The first RSE could be labelled with a DA of type *Info-request*, the second

could also get this label and the third RSE could get the label *Correct misspeaking* since this is what p0 does. So far so good, but the entire remark of p0 could also be labelled as an *Info-request* and this is the issue here. In the examples given the entire remark is labelled as a single DA [Core and Allen, 1997, Allen and Core, 1997], and this is probably the way it should be done, but for the purpose of this assignment I choose to label each RSE with a DA. The extra information obtained in this way can be used to aid in the recognition (of the agent) of what is going on in the meeting.

Just as with the other techniques the problem whether a remark has to be taken seriously for the flow of the discussion appears here too. See the following utterance:

> *"Are all those name fake?"*

Of course the options that are given as possible solutions to the question are names of real cities, but with this remark the speaker shows clearly he has not heard of them before. This remark is labelled as an *Info-request* since it is a question to which an answer can be given, but whether this is a serious question remains to be seen.

Since I have chosen to apply the DAs at the level of the RSEs, I decided to divide the larger DAs into smaller chunks and here is a nice example of why I made this choice:

> *"Ha-ve have you ever heard about Moldova and about those cities?"*
> ...
> *"Moldova yeah, but cities no."*

When I conform to the rules I should treat the answer to the question as a single DA, but it could also be split in two parts [Allen and Core, 1997]. The part in which the speaker signals that he has heard of Moldova and the part where he says he has not heard of the cities. The first part would be labelled as *Accept-part* and the second part would be labelled as *Reject-part*.

I have another problem with the following utterance:

> *"It sounds familiar to me I don't know if is exist but but the"*

According to the standard way of DA annotating this utterance would be labelled as *Abandoned*. And this is also my choice here but it is also more than that because the speaker gives his opinion about the different names, he thinks they sound familiar and he also signals that he does not know if they exist. And according to this fact this utterance could also be labelled as *Assert*, but since the utterance does not make a claim about the world I do not choose this label for it. But I do feel that there is more information contained in this utterance than just the fact that the speaker does not finish it.

And again the issue of the seriousness of a remark comes up with the following utterance:

> *"I would say consider"*

To someone that has only seen the text of the transcript this probably will look like a remark which does not make any sense. But when you look at the video of the meeting you can see that the question and the possible answers are projected onto a screen. And the answers are presented in the following way "Consider: [possible answers]". So here the speaker makes a joke and says that "Consider" could also be an answer to the question. But since he also makes a claim about the world and some people react to it, it surely has a role in the discussion and therefore I label it as *Assert*.

In the following piece of text there is something tricky too:

> p0: *"What about H_ what about H_D_ distance?"*
> p0: *"Can you compare it by just looking at it?"*
> p3: *"I think it is I for all of them"*

In the first two utterances there are requests for information and these can be perfectly modelled using DAs, but in the last utterance (by p3) there are two possibilities. The first possibility is that I label the utterance as *Answer* and that it gets the function of answer to the questions. The second possibility is that I give the utterance the label *Assert* because in his answer p3 makes a claim about the world. I choose for the second option in this since I also have the RSR QUESTION-ANSWER-S with which I am able to model the question-answer relation between the utterances, and thus I get a bit of extra information using a combination of the two different methods.

At the end of the annotating process I identified the number of DAs of each type as can be seen in table 7.1.

DA type	**Occurrences**
Info-request	11
Assert	9
Accept	6
Reject	3
Abandoned	2
Correct-misspeaking	2
Repeat-rephrase	2
Task-management	2
Accept-part	1
Answer	1
Hold	1
Maybe	1
Open-option	1
Reassert	1
Reject-part	1
Signal-non-understanding	1

Table 7.1: *The type and number of DA occurrences in the discussion about Moldova*

7.2 European Championship 2004

In this part of the meeting the conversation immediately starts with a question and an answer to that question, which could be labelled in two different ways. The problem occurs at the following utterances:

> *"If the last European champion-ship championship will be replayed the winner will be"*
> *"Switzerland"*

The question should be labelled as a DA of type *Info-request*, but the answer to the question could be marked as *Answer* or as *Assert* because the answer makes a claim about the world (i.e. the topic of the meeting). I choose for the second option because I also have the RSR annotations of this part of the meeting in which I could model the question answer relation but nothing like the assert DA. So in order to retrieve as much information as possible from the text, I use the two techniques (RST and DAs) to model different aspects of the relation between two utterances.

A little bit further in the conversation p0 makes the following remark:

> *"There is no Spain so"*

This remark is an argument why p0 thinks Switzerland will win the tournament, but there is no way this can be modelled with the help of DAs. So I choose to label this utterance as an *Assert* since it makes a claim

about the world, and with the help of the JUSTIFICATION relation in RST I can model the relation between this remark and his earlier claim that Switzerland will win.

In the previous part of the meeting I had the problem of an utterance that had to be split in two parts because it had an *Accept-part* and a *Reject-part* in it. In this discussion the exact opposite is the case, as can be seen in the following piece of text:

> *"But but if-f-f if if it will be replayed,"*
> *"then nothing change I would say Greece"*

Since I want to try and stick to the RSEs while labelling the DAs, I choose not to combine the two utterances into one DA. I choose to label the second utterance as an *Assert* instead, since the claim about the world is made in that part. The first utterance cannot really be labelled with the help of DAs, but in the annotation that is made with RST the relation between the two utterances is labelled as OTHERWISE. So it is clear there exists a relation between the two, but what it is precisely cannot be modelled using these techniques.

The following utterance also does not have a straightforward label:

> *"Yeah yeah but"*

The speaker signals that he agrees with what being said, hence the "Yeah yeah" part in the utterance, but he ends the utterance with a "but". This word signals that he probably does not completely agree with the previous utterance(s). Since I am not sure whether I should model this as an *Accept*, *Accept-part* or a *Reject-part* DA I choose to label this utterance as a DA of type *Abandoned*. I also find the option *Accept-part* very appealing since there is at least a part which signals acceptance in the remark, but since I do not know what other things he wanted to say I choose to not model this.

A bit later in the discussion there are the following utterances:

> p2: *"If France could come up with strategy to break the Greek defence."*
> p2: *"It's always there"*
> p0: *"Well,"*
> p0: *"that's the problem I think"*

The problem lies in the last utterance, this utterance could be labelled in two ways. It could be labelled as an *Accept* because p0 agrees with p0 that the Greek defence is the problem, but it could also be seen as an *Assert* since p0 makes a claim that the Greek defence is the problem. I choose to model the utterance as an *Accept* since I cannot model this type of relation using RST, and in this manner I get some extra information.

After this a large part of the discussion can be annotated smoothly, but after a while the following utterance comes up:

> *"That is how it goes you know?"*

This utterance is a question of one of the meeting attendants who has just said that Portugal missed a lot of chances and Greece took the one chance they got, and that this is the ways things go. This question therefore cannot be labelled as a DA of type *Info-request*, since the speaker does not want to know if the others know that this is the case. The speaker does want to say that the way things go cannot be predicted, because Portugal is seen as a better team than Greece (in general, this need not be the case amongst the meeting participants) and therefore was expected to win, but they lost. So in a certain way the speaker makes a claim about the world and this could be labelled as an *Assert*, but this does not seem quite right to me too. Instead I choose to model this relation as a DA of type *Hold* since it does address an earlier

statement of the speaker but it does not contain an opinion about the statement. I think this description comes closest, but it is difficult to label this relation using DAs and thus it could very well be that another DA is better, but this seems the best choice for now.

The following issue comes a little bit later, and lies in the following utterances:

"I would say that Greece will win again"
"..."
"Well against Portugal"

The first utterance does not pose any problems, but the second one does. It can be interpreted in two ways, first it can be seen as a accepting reaction to the first utterance which can be modelled as *Accept*, and second it can be seen as a claim on its own which is based on the first utterance which has to be labelled as *Assert*. The difference is not very large but the accepting reaction is modelled in RST using a RSR of type CONDITION, and to retrieve some extra information I choose to model this second utterance as a statement which makes a claim about the world and thus receives the label *Assert*. The claim that is made still is that Greece would win against Portugal, and it does not say anything about playing other teams.

The next utterance does not really cause any trouble but is worth mentioning since it also has two ways of interpreting it:

"Ok we agree that it would be Spain."

The first way this could be interpreted is as a new claim about the world that Spain would win the tournament if it were to be played again, in which case it should be modelled as an *Assert* DA. The second way is that this utterance is used as a solution to the problem who will win the European Championship, and than it should be modelled as a *Task* DA. Since the same speaker said earlier that Spain was out of the tournament I choose to model this utterance with a DA of type *Task*, or else he would be rejecting his own remark and this does not seem very logical to me.

At the end of the discussion when the participants try to reach consensus, the following utterances appear:

"In absence of of Spain,"
"Greece has more chances"

These utterances are definitely related to each other, and this relation was modelled using RST, the CONCESSION relation. So while annotating the meeting with DAs I do not have to label the relation explicitly again, but the first utterance remains difficult. It is not really a claim of the world on its own, but in combination withe the second one it is. The second utterance on its own can also be seen as a claim about the world. So I choose to label the second utterance with a DA of type *Assert*. But the first claim is more difficult since it can be seen as a claim about the world which gives information to support the second utterance, but on its own it has no meaning. This problem is (again) caused by the fact that I want to stick to the RSEs as much as possible. And because the relation between the two is already labelled I choose to give the first utterance the label *Assert* since it gives some information about the world, but I do not think this is a really strong label, but I also cannot think of a better one.

Also in the final phase of the discussion when the meeting attendants are formulating their conclusion the following remark is made:

"Ok so we say Greece"

This remark can be seen as the final conclusion on which all of the participants agree, in which case it should receive the label *Task* since it is the final answer to the problem they are trying to solve and thus is task

related. It can also be seen as a claim about the world and it should get the label *Assert*. But I think the first way is more important in this case and so I choose to model the utterance as a DA of type *Task*.

At the end of the meeting I counted the number of times each DA type occurred and the result can be seen in table 7.2.

DA type	Occurrences
Assert	32
Accept	9
Abandoned	6
Info-request	6
Reassert	4
Reject	3
Exclamation	2
Maybe	2
Task	2
Task-management	2
Acknowledge	1
Explicit-performative	1
Hold	1

Table 7.2: *The type and number of DA occurrences in the discussion about the European Championship*

7.3 Intelligence

This part of the meeting starts with an introduction to the topic and a claim that ants are the most intelligent animals in the world. Which leads to the following reaction:

> *"Well taken as a whole maybe,"*

This remark can be labelled in two ways. The first way is when I look at it as a accepting relation with the claim about the ants but only in a special case, so in this case it should be given the label *Accept-part*. The second way is when it is viewed as a remark to show that the speaker doubts the claim about the ants, which is hinted at by the cue word "maybe". In the second case the utterance should therefore be labelled with a DA of type *Maybe*, this is also my choice because the other relation can be modelled using RST, with the CONDITION relation.

A little bit further the following utterances appears:

> *"Yeah but there's an S_ {laugh}."*

This utterance can also be explained in two different ways. One way is the making of a claim about the world, making it a DA of type *Assert*. The second way is a remark about the performing the task in which case it should get the label *Task*. I choose for the second option here because the task at hand is trying to determine which animal is the most intelligent one, and now the speaker tries to look at how the intelligence of animals should be measured. He signals that the animals which are talked about are referenced in plural and maybe intelligence should be measured per individual animal. And the first option does not seem quite right to me because he does not really say anything about the world.

The next thing I come across are the following two utterances (both made by p0):

> *"Well it's a species,"*
> *"a species yeah"*

The first utterance is labelled as *Assert* and does not give me any troubles, but the second utterance is more difficult. It can be labelled in three different ways *Reassert*, *Accept* and *Repeat-rephrase*. The *Reassert* option does not seem very likely to me because both utterances are made by the same speaker and he does not say it in another way to try and convince people to believe it by giving more or different information. The *Accept* option is considered because the word "yeah" appears in the second utterance, and this could signal an acceptance relation, but since both utterances come from the same speaker this does not seem very logical to me. So that is the reason for my choice for the final option *Repeat-rephrase* and this also fits nicely since this is probably the case when someone repeats himself, this could also be done to give the first utterance more influence but then it would still be labelled in this way.

The next problem that occurs lies in the following two utterances:

"Cause cow as a group,"
"I would bet on cow {laugh}."

These two utterances could be taken together and receive the label *Assert*, but since I would like to stick to the RSEs in the annotating process this is not an option. The relation between the two is already labelled using RST, but the relation is of type CONDITION, so the information I have is that there exists a relation between the two utterances in which the first utterance is a condition for the second one. The second utterance is about solving the problem of which animal is the most intelligent, so it should receive the *Task* label. The first utterance is harder to label, since it does not say anything on its own, but it does support the second utterance. The relation between the two does not need to be modelled again so in this case I choose not to label the first utterance.

A little bit further the following remark is made by one of the speakers:

"In that in that order I'd rate them as ??"

This remark can be seen as two different things, first an utterance that is not finished and hence gets the label *Abandoned*. Second it could be seen as an answer to a question that is asked before, and in this case it should be labelled *Answer*. Since the question and answer relation is already modelled while using RST, the QUESTION-ANSWER-S relations can be used for this purpose. I choose in this case to model the utterance as *Abandoned* since this gives me a little bit of extra information.

The next issue is contained in the following utterances:

p0: *"I would say ants"*
p2: *"Ants yeah"*
p0: *"Yeah"*

The first two utterances can be labelled according to the DA rules of annotating and get the labels *Assert* and *Accept* respectively. But the third utterance is a little bit harder to classify since it could be an accepting reaction to the utterance of p2, or it could be an utterance made to express that p0 thinks it is nice that he and p2 agree. In the first case the utterance would be labelled *Accept* but this is weird because it would be very illogical if p0 would not agree with someone who agrees with him. In the second case I would give the utterance the label *Acknowledge* because then p0 signals that he has heard the remark of p2, but according to the definition [Allen and Core, 1997] of the *Acknowledge* label the speaker does not give his opinion about something. And in this case p0 does not give his opinion explicitly but you cannot say that the opinion of p0 does not play a part in making this remark. But I still choose this option because the other one seems less fitting to me, so the third utterance receives the label *Acknowledge*.

A little bit later the next remark is made by p3, the remark consists of two utterances:

"As a group yeah"
"but that's not really intelligence that's organization"

This remark can be seen as consisting of an *Accept-part* (the first utterance) and a *Reject-part* (the second utterance). But in both utterances a claim about the world is made and that makes it possible for the two utterances to receive the label *Assert*. Although the claim about the world in the first utterance could be seen as an *Accept* relation with the remark that is made before and thus you could also argue that the label combination *Accept* and *Assert* could be correct. The relation between these two utterances is also modelled in the annotation with RST with the CONCESSION relation, which is comparable to the *Accept-part* and *Reject-part* DA situation. So I choose to drop this option, and from the other two options I think the one with something that signals acceptance is better because the remark is more than just the presenting of claims about the world. So I choose to label the first utterance as *Accept* and the second utterance as *Assert*.

The following utterance can also be labeled in two different ways:

> *"Yeah but is-t it can be a proof of intelligence if they can um they can have um critique opinion against other cats,"*

The first way in which I can label this utterance is with a DA of type *Accept-part* because the first part of the utterance signals the acceptance of an utterance which is made earlier. The second way in which I could label this utterance is with the label *Assert* since in the last part of the utterance a claim about the world is made. As can be seen in table 3.3 the *Assert* DA is part of the forward looking functions and the *Accept* DA is part of the backward looking functions. Since this remark comes closer to making a new point than reacting to an already made one, I choose to label this utterance with a label from the category of forward looking functions, the *Assert* label.

A bit later in the conversation the participants argue about which animal can modify the environment and the following remarks are made (by the same speaker):

> *"What do you mean by modifying the environment?"*
> *"If you put a cat in an environment with a a lot of rats"*
> *"It will change the ??"*

The first of the three utterances in the remark is easy to label since it is clearly a question and therefore gets the label *Info-request*. The last two utterances both are not finished on their own, but since I choose to stick to the RSEs as minimal units this cannot be helped. The two utterances are related to each other through the use of a CONDITION relation which is defined with the help of RST, so the relation between the two units is clear. The second utterance on its own does not say anything about the world and on its own it has very little meaning, it is a condition but this cannot be modeled using DAs. I choose to label this utterance as *Abandoned* because I think this utterance is too important to be left unlabeled since it is related with the utterance that follows. The last utterance of the three is not finished, but when you look at the utterances before it, the meaning of the utterance is clear. The speaker tries to make a point that cats can also modify the environment but in another way than ants do. But since he does not finish his sentence and there are question marks in the transcript of the meeting I cannot label this as *Assert* as I would like to. Now I have to choose between *Info-request* and *Abandoned*, and since the speaker clearly does not want extra information I choose the label *Abandoned*.

At the end of the meeting the number of occurrences of the present DA types can be seen in table 7.3.

7.4 Evaluation & Conclusion

The main advantage of DAs for annotation of the discussions is the abundance of labels to label the transcripts with. Because a lot of labels exist a lot of situations can be labeled. Another advantage is that the labels that can be applied are aimed at dialog and can handle agreement and disagreement very well, which is very important in this assignment.

DA type	Occurrences
Assert	41
Abandoned	19
Accept	18
Info-request	12
Communication-management	7
Answer	3
Open-option	3
Reject	3
Repeat-rephrase	3
Task	3
Exclamation	2
Task-management	2
Acknowledge	1
Maybe	1
Reassert	1

Table 7.3: *The type and number of DA occurrences in the discussion about Intelligence*

Disadvantages of this method are the fact that without any way to establish relations between the different DAs, you cannot create some sort of tree which contains the structure of the text. Also in some situations it is not clear what label should be applied to a fragment of text, this is mainly caused by the great number of labels which there are to choose from and the slight difference in meaning between the labels. In those cases the annotator has to choose a label guided by his thoughts about the text that has to be labeled.

When the function of the utterances in the text is the main thing you are interested in, then DAs are the annotating method you should use. But in our case the relations between the various DAs are also very important for the argument structure, so DAs cannot yield the result that is desired in this case.

CHAPTER 8

Issue Based Information Systems

Because the Toulmin trees were not able to visualize the structure of the text, I will look at another annotating method in this chapter. I will explain what the problems are while annotating the transcript of the meeting with the help of the IBIS method.

8.1 Moldova

The main question of this part of the meeting is not posed by anyone but all the participants try to answer it. So I could introduce this question as if it were a question that was posed by some kind of "narrator" and therefore can be annotated using IBIS. For example the following sequence of utterances makes me want to do this:

p2: *"Probably Tiraspol"*
p2: *"Tiraspol"*
p3: *"Tiraspol"*
p3: *"Chisinau"*
p2: *"?? Chisinau"*
p3: *"Chisinev"*
p0: *"I would say Chisinau."*
p0: *"I don't know."*
p3: *"I would say consider"*

In the text above you can see that all of the speakers tell what they think is the capital of Moldova, which can be seen as positions towards the issue "What is the capital of Moldova?". But since the question is not mentioned by any of the participants out loud this issue is not present in the IBIS tree, and for this reason the positions cannot be put in the IBIS tree. But I think these positions do have to be a part of the IBIS tree so I have to come up with a way to fix this. The best solution to the problem is adding the topic of the meeting as an issue to the IBIS tree, because then the participants can react to it and all of these reactions can be added into the IBIS tree, and this is what I want. That none of the positions uttered by the participants are backed up by arguments is another issue, but this does not affect the annotating process.

When you look at the final remark that is made above, it may seem a bit odd but there is an explanation for it. On the sheet in the meeting room the question is displayed as well as the possible answers the meeting attendants can give to the question. But the answers are preceded by the word "Consider", so one of the participants tries to be funny and chooses this as answer to the question. Every person can immediately understand this is not supposed to be a serious remark, but if this process gets automated in the future this becomes problematic so I have two options. Annotate the remark as any other and include it in the IBIS tree, or I could skip the remark in the annotating process. I choose for the first option because when people look at the visualized tree of the structure of the text they can immediately see that this remark is meant to be funny rather than seriously. When I leave the remark out of the tree, people cannot just guess it was present in the meeting, and this does not seem right to me.

The next strange thing while annotating is that sometimes participants make a remark that is not really anything that can be put in the IBIS tree. Because an issue in the tree has to have the form of a question [Kunz and Rittel, 1970], and sometimes participants say something new that is not a question, as is shown in the following piece of text:

p3: *"Chisinev sounds very Moldavian"*
p1: *"At least is the only word I can pronounce"*

The first utterance in this example can be seen as an answer to a question which was under discussion earlier during the meeting. So I choose to model the first utterance as a position of p3 towards that question which is adopted in the tree as an issue. The second utterance is reaction to the first one, but it is not a position towards the issue and since it is no question it cannot be an issue too. So the only options I have left to label the utterance with are the "argument" label and the "other" label. It is quite clear that the second utterance is not an argument for the first one either, so this option is not right either. So the last option is to label it as other and although this is exactly what the other label is for [Conklin and Begeman, 1988], I am not very happy with it since it does not give much information about the structure of the discussion.

At the end of this part of the meeting I have counted the number of IBIS elements which I have annotated. This includes the issue that contains the main question of the discussion which is not explicitly spoken out loud by any of the participants. The results are depicted in table 8.1.

IBIS element	**Occurrences**
Position	26
Issue	10
Other	4
Argument	2

Table 8.1: *The type and number of IBIS element occurrences in the discussion about Moldova*

8.2 European Championship 2004

The first thing that is said in this part of the meeting, is something very much like the question which the discussion is about. So in this case I do not have to introduce the question myself in the IBIS tree. In the beginning of the meeting there is another problem though:

p0: *"Switzerland"*
p3: *"Yeah definitely"*
p2: *"no way"*
p3: *"Switzerland will win"*
p2: *"I see more chances to France"*
p0: *"There is no Spain so"*
p1: *"A European audience"*
p2: *"Well Spain is out of the tournament"*
p0: *"It was"*
p2: *"It was thrown out of the tournament"*

In this piece of text the participants try to answer the question about which country would win the tournament. There are different positions, someone says Switzerland and someone else France, so the issue has different positions in the IBIS tree. But there are also some arguments in this piece of text, but no one exactly says which argument goes with which position. To solve this problem I have chosen to group the arguments and the positions of the same speaker. From the utterances above I have combined for example the position "Switzerland" of p0, with argument (also of p0) "There is no Spain so".

The next issue in the meeting is contained in the following utterances:

"But but if-f-f if if it will be replayed,"
"then nothing change I would say Greece"

Both utterances are made by the same speaker and are contained in a single speaker turn, but just as with the DAs I would like to keep the RSEs as minimal units so I choose to split this remark in the two utterances shown above. This approach will be used also during the annotating of the remaining part of the meeting, since this situation will occur a few more times. My solution for this problem is to label only the second utterance as a position towards the issue. I choose to do this because when you look at the first utterance in itself, it does not contain any information about what the opinion of the speaker is, and the second utterance does.

Another problem that occurs fairly often is the early aborting of sentences, a lot of remarks are not finished for example the ones shown below:

"to change the result so that kind of thing so that a"
"Stop that kind of"

When this situation occurs I choose to look at the meaning of the utterances with regard to the issue to which they respond. In the utterances shown above there is no position of opinion of the speakers towards any issue presented so far during the meeting. So I choose to exclude them from the tree which visualizes the structure of the discussion (or the design rationale in a design meeting).

The next problem I encounter while annotating the text can be seen in the following example piece of text:

p1: *"No I I I don't know because Portugal played well against Greece twice and the second time the period thought ok we learn the strategy,"*
p1: *"so we will change the strategy against Greece but they lost again so"*
...
p2: *"Well Portugal also Portugal also missed so many easy chances to score in that match so,"*
p2: *"in fact they had a good strategy,"*
p2: *"but the-y the-y they did not take their chance."*
p2: *"Greece got only one chance and they took it."*

The first utterance of p1 is the position to which all of the following utterances react. The rest of the utterances can therefore be seen as arguments, and so I label them accordingly. But the final four utterances of p2 originate from one speaker turn of p2 and thus can be seen as one very large argument to the position taken by p1. And since I want to stick to the RSEs as units that have to be labelled I choose to label each of the utterances as a separate argument. This is plausible I think because each of the four final utterances state a fact about the domain of the meeting. First p2 says that Portugal has missed many chances, in the second utterance he says Portugal has a good strategy, in the third utterance he says they did not take their chances and in the last one he says that Greece took their only chance. So these are all arguments supporting or opposing the position which is made by p1 in his first utterance.

The number of IBIS elements and the number of times each element occurs are summarized in table 8.2.

IBIS element	Occurrences
Argument	19
Position	19
Issue	5
Other	2

Table 8.2: *The type and number of IBIS element occurrences in the discussion about the European Championship*

8.3 Intelligence

In this part of the meeting the main question is which animal is the most intelligent. A few options are given and the participants choose the animal which they think is the smartest. The question is not spoken out loud (just as in the part of the meeting about Moldova), so I have to introduce a "narrator" once again to be able to include the main question in the IBIS tree.

In the beginning of the discussion I come across some minor difficulties which I think are still worth mentioning:

p2: *"?? cats"*
p3: *"Yeah but there's an S_ {laugh}."*
p3: *"There is a problem here"*
p0: *"Well it's a species,"*

The issue to which p2 reacts with the first utterance is the main question of the discussion, thus which animal is the most intelligent. The position taken by p2 in the first utterance is doubted by p3 in the two following utterances, and in the last utterance p0 defends the position taken by p2. All these reactions to the position are labelled as arguments because they do signal how the speakers think about the position of p2. However they do not say anything about what p3 and p0 think that the answer to the main question should be, so I choose not to label them as positions. Another reason for this choice is that the speakers only give reasons why they think about p2's position in their way.

The next problem I encounter comes a little bit further in the discussion:

"As a group yeah"
"but that's not really intelligence that's organization"

These two utterances are made by the same speaker and originate from one speaker turn, and that is where the problem comes from because if I did not have to stick to the RSEs this could be a position towards an issue as a whole but now I have to label the two utterances apart. The first utterance can still be labelled as a position towards an issue but the second one cannot, and it cannot be labelled as an argument for the

first utterance either since it does not say anything about why the speaker thinks that his first utterance is correct. With a little creativity you could say that the second utterance is an argument which objects to the first utterance, but since they are both made by the same speaker this is also a little bit strange. But because this is the option which resembles the reality closest and avoids the use of the "other" type IBIS element I choose to do this.

A little bit further I find another difficult situation with the following utterance:

> *"Um the cats hardly live together,"*
> *"you know"*

With the first utterance shown above a discussion is started within the main discussion of the meeting, and therefore I would like to model it as an issue. But according to the definition of issues in IBIS [Kunz and Rittel, 1970] the utterance has to have the form of a question, which clearly is not the case here. But this utterance is so important that I choose not to follow the definition exactly in this case, so I do model this utterance as a new issue. I think this is justified in this case because when you look at the second utterance you could argue that the remark as a whole has the form of a question and thus could be modelled as an issue. So here it is the RSE problem again when you look at it in this way.

Another situation that occurs in the transcript of the meeting is the following:

> *"If you put a cat in an environment with a a lot of rats"*
> *"It will change the ??"*

The utterances above are again from a single speaker turn and again I would like to treat them as a whole, but the RSE problem comes up again. And this time it can be solved by labelling the first utterance as a position and not labelling the second utterance because on its own this utterance does not say anything. The second utterance also is not finished but everyone understands what the speaker means with it, so I think it should be part of the IBIS tree. Therefore I choose to model both utterances as positions towards the issue to which they respond.

The number of times each IBIS element appears in the discussion about which animal is the most intelligent one are depicted in table 8.3.

IBIS element	**Occurrences**
Position	43
Argument	20
Issue	13
Other	0

Table 8.3: *The type and number of IBIS element occurrences in the discussion about the European Championship*

8.4 Evaluation & Conclusion

A great advantage of this technique is that it is very easy to create a tree which represents the structure of the text. Especially when the approach of gIBIS [Conklin and Begeman, 1988] is used, the visualization is also not very hard because the tree is adapted to this purpose.

Although the IBIS technique can be used to visualize the structure of the text, there still remain some difficulties that are mentioned in this chapter. These difficulties include remarks that should be labeled as issues but are not in the form of a question like this method requires. Another difficulty is that remarks are not finished and therefore no label can be applied to them. The greatest part of these difficulties can

be solved manually because when a person (in this case me) labels the utterances he or she knows where the discussion is about. If this process would become automated this could cause some trouble since the meaning of the text is not known to the program, it only knows the function and some relations between the utterances which are defined in its input.

Also the lack of flexibility in assigning the labels causes some trouble, in a number of situations a little more freedom would be nice. Although this method is quite good at representing argument structures it is not the right method for this assignment because of this "handicap".

CHAPTER 9

Annotating analysis

In this chapter I will look at how many utterances can be labeled using the four annotation methods from chapters 5 through 8. Since none of the methods used in these chapters can be used to create the argument structures that I need which are discussed in chapter 2, I will look at the strengths and weaknesses of each method to see if these can be combined in any way.

9.1 Annotating method comparison

The strengths and weaknesses of each annotating method are listed here again for clarity. The strengths for RST are:

- Capable of creating RSTTrees.
- It is allowed to add any relation you need.

and its weaknesses:

- No possibility to label the nodes in the argument structure.
- Relation set could become very large.
- Cannot handle typical dialog situations like agreement or disagreement.

The strengths for the Toulmin model are:

- Capable of creating Toulmin trees.
- The structure of an argument can be described in great detail.

and its weaknesses:

- A lot of empty spaces in the Toulmin trees, since not all the information is present in the tarnscripts.

The strengths of DAs are:

- A lot of labels that can be applied.
- It is aimed at dialogs and can therefore be used very well for meeting transcripts.

and its weaknesses:

- No way to relate the different DAs.
- It is sometimes difficult finding the correct label for a situation.

The strengths of IBIS are:

- Very easy to create trees of the structure of the text with.

and its weaknesses:

- Little flexibility when assigning the labels.

When we look at these strong and weak points of each annotating method we see some similarities. Some techniques are not capable of labeling nodes or relations which are used in the argument structures. We also see that with the methods that have a large set of labels that this can be as good as well as a bad point for the method. And a strong point of the annotating methods which is very important is the capability of visualizing the structure of the transcripts. Another good point that is important is whether the method is able to deal with dialog situations like agreement and disagreement.

Since none of the annotating methods that are used here have all of these properties, a new annotating method has to be developed which has all those characteristics. And therefore we have built the Twente Argument Schema, which is explained in greater detail in chapter 10.

9.2 Annotating statistics

For each technique I used to annotate the transcripts of the meeting I kept track of the number of utterances I was able to label. I counted the number of labels two times, one time to find out how many of the utterances I was able to label and another time to find out how many utterances I could label with a label that could be used to visualize the structure of the text. The label OTHERWISE in RST for example does count in the first counting run, but does not count in the second one.

Tables 9.1 and 9.2 contain the results of the part of the meeting which is about which city is the capital of Moldova.

Tables 9.3 and 9.4 contain the results of the part of the meeting which is about the European Championship.

Tables 9.5 and 9.6 contain the results of the part of the meeting which is about choosing the most intelligent animal.

Total number of utterances: 56		
Used technique	**Labeled**	**Coverage**
RST	47	84%
Toulmin	9	16%
IBIS	41	73%
Dialog Acts	45	80%
Average coverage:		63%

Table 9.1: *The number of utterances that can be labeled*

Total number of utterances: 56		
Used technique	**Labeled**	**Coverage**
RST	29	52%
Toulmin	9	16%
IBIS	37	66%
Dialog Acts	37	66%
Average coverage:		50%

Table 9.2: *The number of utterances that can be labeled in a useful way*

Total number of utterances: 82		
Used technique	**Labeled**	**Coverage**
RST	60	73%
Toulmin	22	27%
IBIS	45	55%
Dialog Acts	71	87%
Average coverage:		60%

Table 9.3: *The number of utterances that can be labeled*

Total number of utterances: 82		
Used technique	**Labeled**	**Coverage**
RST	42	51%
Toulmin	22	27%
IBIS	43	52%
Dialog Acts	59	72%
Average coverage:		51%

Table 9.4: *The number of utterances that can be labeled in a useful way*

Total number of utterances: 126		
Used technique	**Labeled**	**Coverage**
RST	99	79%
Toulmin	29	23%
IBIS	75	60%
Dialog Acts	119	94%
Average coverage:		64%

Table 9.5: *The number of utterances that can be labeled*

Total number of utterances: 126		
Used technique	**Labeled**	**Coverage**
RST	75	60%
Toulmin	29	23%
IBIS	75	60%
Dialog Acts	89	71%
Average coverage:		53%

Table 9.6: *The number of utterances that can be labeled in a useful way*

9.3 Conclusion

As can be seen in the annotating statistics DAs have a very high level of coverage, but with DAs alone it is just impossible to create argument structures. The addition of some type of relations with which you can connect the different DAs is needed. RST also has high scores for coverage, and a combination of RST and DAs could be an option. One drawback of this option is the very great number of labels you would get. So this is not a good option for this assignment.

The scores of the IBIS method look promising, but the disadvantage of this technique is that for every issue that is discussed in a meeting a new argument structure is introduced. And because the goal of this assignment is the creation of an argument structure to give insights in the course of a meeting, this method is not preferable also.

The scores of the Toulmin model are very low. This is mostly because the information that the Toulmin model is capable of displaying misses from everyday meetings. A lot of things that can be visualized or labeled using the Toulmin model are in the common sense of meeting participants and that cannot be captured in an argument structure.

Since none of the methods I have looked can be used to create argument structures a new labelset is developed called “Twente Argument Schema” which is introduced in chapter 10.

CHAPTER 10

The Twente Argument Schema

In our research we would like to be able to visualize the argumentation structure of the discussions which are held in meetings. There exist a number of different theories for labeling (transcripts of) meetings, and we have tried Rhetorical Structure Theory (RST) [Mann and Thompson, 1987, Mann et al., 1992, Marcu, 1998, 1997, Stent, 2000], Toulmin models [Toulmin, 1958, Newman and Marshall, 1991], Dialog Acts (DAs) [Bunt, 1979, Allen and Core, 1997] and the IBIS scheme [Kunz and Rittel, 1970]. None of these methods could be used to our satisfaction in creating a tree which visualizes the structure of the discussion. Therefore we decided to compose our own labelset to annotate the meetings with and here we introduce the Twente Argument Schema, which is developed in order to structure textual units of a meeting transcript. In the process of designing the Schema we of course looked at the theories mentioned earlier and tried to incorporate their advantages in our labelset. The Schema contains labels for transcript fragments as well as labels for relations between these fragments. The Schema is explained below.

10.1 Analysis & Design

First we had to determine what we wanted to do with the labelset. The purpose is to be able to visualize the structure of the discussion of a meeting in the form of a tree structure. In this tree nodes and edges exist, and we wanted to be able to label both. So we decided that there should be labels which could be applied to utterances of a speaker (the nodes in the tree), and labels that could be applied to the relations connecting these utterances (the edges in the tree).

In the meetings we use a lot of discussions are held and it would be nice if we could model whether speakers agreed or disagreed with each other. In the annotation with DAs we could use the DAs of type *Accept*, *Accept-part*, *Reject-part* and *Reject* to model this. There is however one drawback, we could not annotate what the speaker accepted or rejected in his remark. But because we wanted this to be possible with our labelset we looked at RST because with that theory there are numerous possibilities to specify relations between utterances. But in the set of RSRs that we have used here, there is no relation with which we could model agreement or disagreement. In the IBIS model however you can label whether positions support or object to an issue, and this can be compared with our wish here. Therefore we introduced the label “Statement” to label the utterances with, and the label “Positive” or “Negative” to label the relation between them.

Sometimes participants of the meeting make remarks from which can be understood that they are not sure

of what they say is actually true. In the Toulmin model this can be visualized in the scheme by using the qualifier which is part of the scheme. This is the only one of the four theories which has this possibility. Because we wanted to be able to label whether something is said because the speaker is sure of it or he just guesses or thinks that it is true. Therefore we created the label "Weak-statement" to signal that the speaker is not sure of what he just said. This label also gives us extra information about the strength of the argument in a way that is comparable to the qualifier in the Toulmin model, in this model a qualifier is used to signal the degree of certainty with which the claim is made. Whenever an utterance is labeled as weak statement, it often contributes less to the conversation than an normal statement would. A lot of times when an utterance is labeled as weak statement, it is related to another utterance. For this relation we have the label "Uncertain", because in this case positive or negative relations could be used but, although the utterance of the speaker can be positive or negative, there still remains some doubt.

Another relation that occurred very often during the annotating process was the QUESTION-ANSWER-S relation from RST. This relation could also be modelled by using Issues and Positions from the IBIS model. In the IBIS model issues often have the form of a question [Kunz and Rittel, 1970]. We think of issues as utterances that are made with the intention for a reaction, in the same way as a question is (most of the time) followed by an answer. So when a reaction is expected by the speaker, we label his utterance as an issue. But since the speaker can expect different kinds of answers we also have three kinds of issues, an "Open issue", an "A/B issue" and a "Yes/No issue". An open issue can have all kinds of answers, and the relation between the issue and its answer is labelled as "Option". The relation between an A/B issue and its answer is labelled in the same way, but we use an A/B issue when the number of answers to the issue is limited. For example a question like *"What do you prefer, cheese or milk?"*, the expected answer is one of the two alternatives, cheese or milk. A Yes/No issue is used in the same way as an A/B issue only the expected answer to the issue is yes or no. The relation between a Yes/No issue and its answers is labelled as "Positive" or "Negative". When a speaker gives a reaction to an issue, for example an utterance saying that he doesn't know it wit a Yes/No issue, the relation between them is labeled as "Option". This goes for all the relations between issues and their reactions mentioned above.

The situation where a meeting participant gives a reaction to an issue can sometimes also be of the following kind:

> *"I think we we can eliminate cow anyway"*

This of course cannot be labeled in any way described above as a reaction to an issue. Therefore we have introduced a new relation type "Option-exclusion". This relation is used when the speaker says that something should not be the case in a reaction to an issue. This relation is adopted from the option from the IBIS model, since saying that something should not be the case can be seen as an option which is a reaction to an issue in the IBIS model. But it is a different situation when a speaker says that the answer should not be option A, in stead of saying that the answer should be option B or option C (when the choice is between either A, B or C).

Another common situation is that speakers explain something, they give their opinion and after this they tell the other participants why they have that opinion. In RST there are relations like JUSTIFICATION and ELABORATION that can be used for this purpose. But the JUSTIFICATION relation seems to impose some kind of restriction on the utterances between which the relation holds, you could expect that the utterance which justifies the other one should be a true fact. We find a comparable situation in the Toulmin model. In Toulmin models a claim can have a backing which is based on a fact but we cannot label a claim that is based on another claim (unless we use nested Toulmin trees). So we decided to make a relation for this purpose and we have named it "Clarification". This relation is used when a speaker explains why he thinks in the manner that he does, but we do not make the distinction between whether the opinion of the speaker is based on a fact or the thoughts of the speaker.

Another relation we have included is based on the possibility within the IBIS model to let an issue specialize or generalize another issue. This relation is the "Specialization" relation and can for example be used when a speaker poses a question (which is labelled as any type of issue) and another speaker makes the question

more specific for the topic which is discussed. This relation can also be used whenever there is a relation needed between two statements where one statement is a specialization of the other one.

The final relation we have included in our labelset is the "Subject-to" relation. This relation can be compared to the CONCESSION or CONDITION relations from RST. And since these two relation types occurred quite often during the annotating process we chose to add such a relation to our labelset. This relation can be used whenever one utterance needs to be related to another utterance in which something is said that is dependant of what is said in the first utterance. The point made in the second utterance could not be understood without the content of the first utterance, see for example the two following utterances:

> *"That's much more difficult with a cow. If you leave something in a kitchen, you are less likely to find a cow u-"*
> *"It depends if the cow is very hungry."*

In the example above, the second utterance does not make any sense unless it is related to the first utterance. And in this situation the subject to label should be used.

10.2 Argumentation Labels

In this section all the node and relation labels that are initially defined for the Twente Argument Schema are shown for reference throughout the chapter. For a more detailed description of them and situations where these labels should be used see appendix A. In this appendix also the changes mentioned in 10.3 are applied so it is possible that the tables shown there and table 10.1 contain differences.

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

Table 10.1: *The labels of the Twente Argument Schema*

10.3 Changes

During trials with the initial labelset some flaws were found in it. So we have made a number of changes in the original labelset as is defined above and these changes will be explained in this paragraph. The relation "Clarification" was renamed to "Elaboration". This is not just a change in the name of the relation but it could be applied in more situations also. Sometimes when the clarification relation was used, it was not really a clarification that appeared in the discussion. It was more like a person that continued talking about the same topic and sometimes clarifying himself, and sometimes just continuing his remark and giving some additional information and therefore we thought it was better to change the relation. An example of such a situation is given below.

> p3: *"Um you mean that changing the coach wou-ld cou-ld could make France win?"*
> p3: *"So with a different coach it could be France?"*

In the example above the speaker does not ask himself to clarify his first remark in the second one, but he tries to make himself clear towards the other meeting attendants. And we found that in this situation the elaboration relation is more suitable than the clarification relation. We chose not to add the elaboration to the labelset but to alter the clarification relation because the strength of our labelset is partly in the fairly small amount of relations that we have. And another reason for this decision is that whenever a clarification relation needs to be used, this can also be labeled with an elaboration relation.

Another change we have made to the original labelset was the adding of a new relation label which was the "Generalization" relation. This relation is the opposite of the specialization relation and that is also the main reason why we wanted to add this relation. But after we had annotated a number of discussions with the labelset which included the generalization relation we found out that it was never used. So after a short period we decided to remove the relation from the labelset because it is useless to include a relation label that is never used, especially since we tried to keep the number of labels in the labelset as low as possible.

The final change we have made to the labelset is the addition of a relation called "Request". This relation is used in situations where meeting attendants ask for something from their fellow attendants. For example when someone makes a statement and other person wants to have some additional information about it, that would be called a request for extra information, hence the name of the relation. How this relation is used is shown in the example below.

p0: *"I see but ants built, they're able to built um well they modify our gardens"*
p1: *"What do you mean by modifying the environment?"*

In this example speaker p0 makes a claim about ants being able to modify the environment, and speaker p1 does not know precisely what p0 means with this. So he asks p1 for some explanation what he means by environment since it can be mean a number of things and some of them cannot be modified by ants and some things can. And later on in the discussion it will be clear that p0 means that ants can change the environment by building ant hills, so that is what he means with modifying the environment.

10.4 Annotating a meeting

To test the newly created labelset we have labeled the same meeting with it as in chapter 4. The problems encountered in this process will be explained in this paragraph.

10.4.1 Moldova

The first part of the meeting I have annotated is the part about the capital of Moldova. The first situation where two possible relations can be established between nodes is the one in the following example.

p0: *"Chisinev? Chisinau"*
p2: *"?? Tiraspol"*
p3: *"Are all those name fake?"*

In the piece of text above p0 and p2 provide some options as answer to a question which posed a little earlier in the discussion. P3 reacts to this by asking whether these names of cities are real names. This situation can be modelled using a request relation or using an uncertain relation or even a negative relation. The use of a negative relation is a bit far fetched though because you would have to assume that p3 thinks all the options given are not the capital of Moldova and therefore he makes this remark, so I have not chosen to model the relation as a negative one. A request relation is another possibility here, but this also not the relation of my choice for this situation because a request relation is meant to be used in cases where the speaker (in this case p3) wants some additional information about the ongoing topic, and this is not the

case here. So I have chosen to use an uncertain relation in this situation because p3 makes his remark in the form of a question and therefore it seems logical that he does not know for sure whether these names are the capital of Moldova or not.

The next situation in which the choice for the right label is not straightforward is depicted below.

p3: *"Tirapsol Chisinau"*
p2: *"?? Chisinau"*
p3: *"Chisinev"*
p0: *"I would say Chisinau. I don't know."*
p3: *"I would say consider"*

In this piece of the discussion all the possible answers to the question what the capital of Moldova is are named. To understand this problem the way in which the question is presented has to be known. A beamer projects a slide onto the wall and that slide contains the question and the following sentence: "Consider: Chisinau, Chisnev or Tiraspol" to list the possible answers from which the meeting participants have to choose one. In this case p3 ridicules this is his last remark when he presents consider as an alternative for an answer to the question too. Normally remarks of this form receive the label weak statement just as the remark of p0 in this particular example does. But since the remark from p3 is meant as a reaction to the remark of p0 and to make fun a little bit I have chosen to give it the label of a statement which is used to signal that p3 does not agree with p0 in this case.

In table 10.2 the number of times a label occurs in the discussion about Moldova are shown.

Node labels		**Relation labels**	
Statement	15	Positive	11
Weak statement	2	Negative	4
Open issue	1	Uncertain	1
A/B issue	1	Elaboration	0
Yes/No issue	4	Specialization	1
		Option	2
		Option exclusion	0
		Subject-to	0
		Request	0

Table 10.2: *Twente Argument Schema label occurrences in the Moldova discussion*

10.4.2 European Championship 2004

This part of the meeting deals with the question of who will be the winner of the European Championship of 2004 if it were to be replayed. A difficult situation which occurred during the annotation process is the following.

p2: *"France has team to change, to change the result."*
p1: *"Stop that kind of"*
p2: *"Um so"*
p0: *"Well I wonder"*
p2: *"Huh"*
p3: *"You mean that"*
p0: *"That was true in the past but now"*

In this part of the text p2 first makes a claim that France wins the championship if it was played again. This claim is labeled as a statement and that is very straightforward. The last reaction of p0 to this claim is a

different matter though, from the text that is in the transcript (and which is copied here) cannot be seen if the remark should be labeled as a statement or as a Yes/No issue. Another possibility is to split the remark of p0 in two utterances "That was true in the past" and "but now". But since the only thing that happens in that case is the moving of the problem to the latter of the two utterances I have chosen not to do this, so the choice between the two labels is still open. If we look at the other remark of p0 in this piece of the transcript you can see that p0 is not sure of what p2 claims in his first remark. But in this remark he also makes clear he is not sure which other team would win the championship. And between the two remarks of p0 no new information is presented so I assumed that p0 still was not sure when he makes the second remark and therefore I have chosen to apply the label Yes/No issue in stead of the statement label.

In table 10.3 the number of times a label occurs in the discussion about who would wind the European Championship of 2004 are shown.

Node labels		**Relation labels**	
Statement	31	Positive	19
Weak statement	1	Negative	6
Open issue	0	Uncertain	1
A/B issue	1	Elaboration	1
Yes/No issue	5	Specialization	3
		Option	4
		Option exclusion	0
		Subject-to	3
		Request	0

Table 10.3: *Twente Argument Schema label occurrences in the European Championship discussion*

10.4.3 Intelligence

This section of the transcript of the meeting contains the text from the discussion about which animal is the most intelligent. In this discussion the topic becomes what intelligence is exactly and during this sub-discussion the following situation occurs which delivers a problem for the annotator.

p3: *"All- all of this is true, but it this not related to intelligence. Yeah good a good adaptation capacity they have good group behaviour, but they don't have any initiative or"*
p0: *"Well yeah but"*
p1: *"What is intelligence?"*

In this example the first remark of p3 is split into three utterances, but here we can leave it one remark because for this example the labels that are applied to the first remark do not matter. The difficult bit of this example is the last remark made by p1. He poses a question about what intelligence is and if you look at the question you would expect an open issue label in this case. But since in the part of the discussion that has taken place already a few possible options for how the meeting participants should define intelligence are named, I have chosen to label this question as an A/B issue. Another reason for choosing for an A/B issue is that I did not expect a completely unknown answer to this question so an open issue label was out of its place.

In table 10.4 the number of times a label occurs in the discussion about which animal is the smartest are shown.

Node labels		Relation labels	
Statement	63	Positive	31
Weak statement	2	Negative	19
Open issue	3	Uncertain	2
A/B issue	2	Elaboration	1
Yes/No issue	3	Specialization	1
		Option	14
		Option exclusion	2
		Subject-to	1
		Request	1

Table 10.4: *Twente Argument Schema label occurrences in the intelligence discussion*

10.5 Theory performance

During the annotating of the discussions of this meeting some annotating properties of this new labelset have become clear. Some of them are advantages over other theories I have used (see 9.1) and some of them are disadvantages and I have covered them in greater detail in this paragraph. Also some other issues concerning the performance of the Twente Argument Schema are shown.

One of the main advantages of the Twente Argument Schema is its number of labels that can be assigned. It is fairly limited but it still offers a lot of expressing power. When you compare it for example with RST, you see that RST defines a lot more relations than the Twente Argument Schema does, and in RST it is also possible to add any relation you might need. But since the Twente Argument Schema is capable of assigning labels to the nodes also (which RST cannot do) you get the same amount of expressing power with a smaller number of labels. For fairness you should compare all of the labels in both methods but still the Twente Argument Schema has less labels than RST has. An example of a situation where the information of both node and relation labels is helpful is shown in the example from section 5.1 below.

"Moldova sounds like ting ting"
"Yeah"

In RST there is no possibility to assign a relation between the two utterances unless you add a new label even making the difference between the number of relations greater. But when the Twente Argument Schema is used you could apply a positive relation between the two utterances, which themselves should be labeled as statements.

This advantage of the Twente Argument Schema also holds when you compare it with DAs. Only the number of labels that can be assigned to nodes is much greater with DAs since they do not offer any options for defining relations between nodes. The Twente Argument Schema is capable of doing both and the number of labels in the Twente Argument Schema is lower than with the DAs so that is an advantage of the Twente Argument Schema. Again all of the relations of the Twente Argument Schema have to be counted for fairness. An illustration of this problem is given in the example below, which is taken from 7.3.

p0: *"I would say ants"*
p2: *"Ants yeah"*
p0: *"Yeah"*

The difficulties with the labeling of this fragment are related to the lack of relations between nodes in the DA way of annotating. Since the Twente Argument Schema does offer this functionality the relations between the utterances can be shown and can the utterances themselves be labeled. The first utterance would be labeled as a weak statement, the second one as a statement and the third one also as a statement. And the

relation between the first two utterances would be a positive relation and the relation between the second and third utterance also. So this is an example of a situation that DAs have trouble handling and where the Twente Argument Schema shows its capabilities.

When the comparison between the Twente Argument Schema and the Toulmin model is made, the same advantage does not apply. Both of these annotating methods have labels for nodes and labels for relations at their disposal. But there is a difference, the labels of the Toulmin model are not dynamic. They always apply to the same elements or between the same nodes when it is a relation label. And the labels of the Twente Argument Schema can be used at any place in the argument structure at any time, and relation labels can be applied wherever they are necessary. An example, taken from section 6.3, of this is shown below.

"Ants are the most intelligent animals in the world."
"Well, taken as a whole maybe, but individually, no."

This situation does not fit very nicely into the Toulmin structure as is explained in 6.3, but when the Twente Argument Schema is used, this situation can be labeled nicely. Both utterances are labeled as statements and the relation between the two is a negative one, since the second speaker does not agree with the first one. So if you have an argument structure that fits exactly in the labels defined by the Toulmin the Twente Argument Schema cannot match the capabilities of the Toulmin model. But because the argument structure that is the speciality of the Toulmin model practically never occurs in meetings the Twente Argument Schema is better suited to our needs. The structures that do occur in everyday meetings can be captured thanks to its labels that can be applied where they are needed.

The last other theory that I have used to annotate transcripts of a meeting with is IBIS, and when I compare it to the Twente Argument Schema there are a few similarities between the two methods. They both have labels to apply to nodes and to relations and the labels have some similarities also. They both identify issues, options and relations that make clear if someone agrees or disagrees with another person. But there are also differences, like the statement label in the Twente Argument Schema. This label is not a part of IBIS and therefore in IBIS only argument structures that are related to an issue can be labeled, but in meetings not every discussion is related to an issue as is shown in the example below, which is taken from 8.1

p2: *"Probably Tiraspol"*
p2: *"Tiraspol"*
p3: *"Tiraspol"*
p3: *"Chisinau"*
p2: *"?? Chisinau"*
p3: *"Chisinev"*
p0: *"I would say Chisinau."*
p0: *"I don't know."*
p3: *"I would say consider"*

In this example the first utterance could be labeled as a statement and could serve as the node to which the rest of the discussion is related. With IBIS you would have to introduce a new node containing an issue to which all these utterances react, but in the Twente Argument Schema this is not necessary and so we can stick to the text in the transcript of the meeting. For clarity and readability of the argument structure something could be said for the introduction of an issue too, but it would be like adding something to the meeting that was never said during the meeting and it is not the purpose of an argument structure to add extra information to the meeting. This example shows exactly why the Twente Argument Schema has the statement relation to which an argument structure can be related. In IBIS you can use the the label "Other" for this purpose, but since a label other does not give much extra information except the fact that there is something that is related to a discussion, I did not find it a very attractive alternative. And for the same reason there is no label with a similar function in the Twente Argument Schema.

A disadvantage of the Twente Argument schema is the use of the uncertain relation in combination with the weak statement node. In some situations a speaker makes a claim of which he is not sure, but in this claim he agrees with another speaker. In this situation you have two options. The first one is to label the claim as a weak statement and to label the relation as a positive one, and you would read from it that the speakers agree but that one is not sure of what he says. The second option is to label the relation as an uncertain one and label the claim of the speaker as a statement. In this case you can say that the second speaker doubts what the first speaker says and that he shows this by making a statement. In most cases this problem can be solved by looking at the words which are used in the remark of the second speaker and when words like "maybe" or "probably" are used you can use the first option and otherwise the second option. This situation can also occur withe negative relation and the weak statement labels when people disagree but doubt what they say. This problem does not occur with the other methods because the nodes and relations between them are fixed, or no nodes or relations are supported.

Coverage statistics like in section 9.2 cannot be computed of the Twente Argument Schema. This is because the nodes can be counted and how many utterances are covered by them, but then we would forget about our relation labels. And since the Twente Argument Schema is the only annotation method that has nodes and relations that can be placed between almost any pair of nodes this is the only method from which the coverage measure is affected by this. To still be able to compare the Twente Argument Schema with the other methods I have made a new measurement. In each discussion I count the number of labels (node labels and relation labels) that can be assigned by each theory. And the theory with the largest amount of labels provides the highest amount of information. In this comparison I will include any label from any theory that is used in the labeling process. The results of this are shown in table 10.5.

Discussion	**Twente Argument Schema**	**RST**	**Toulmin**	**DAs**	**IBIS**
Moldova	42	34	12	45	70
European Championship	75	59	31	71	83
Intelligence	145	83	36	119	139

Table 10.5: *The number of labels that can be applied by each annotating method.*

As can be seen from table 10.5 the Twente Argument Schema outperforms both RST and the Toulmin model, and it is of the same level as the DAs and IBIS, although in the first two discussion IBIS assigns more labels than the Twente Argument Schema. But this is compensated for by the possibility of the Twente Argument Schema to create a single tree of the entire discussion which is something that IBIS cannot do. All of these results are based on the annotations of the meeting described in chapter 4 and these results were encouraging enough to continue further testing of the Twente Argument Schema.

During the testing of the Twente Argument Schema three different annotators have worked with it and I looked at the results of that process. The further testing consisted of the labeling of a large number of meetings which all had the same scenario. In these meetings four participants received the task of designing a new remote control for a certain company. From these meetings the discussions were labeled using the Twente Argument Schema and the following results are based on this data In total the three annotators have used 5152 node labels and 4890 relation labels. I have looked at the number of times each annotator used each node label from the Twente Argument Schema and calculated the percentage of the usage of each label. The results of this are shown in table 10.6.

Annotator 1		**Annotator 2**		**Annotator 3**	
Statement	81,4%	Statement	80,5%	Statement	82,9%
Yes/No issue	7,5%	Yes/No issue	9,8%	Yes/No issue	9,0%
Weak statement	6,1%	Open issue	5,6%	Open issue	4,6%
Open issue	3,7%	Weak statement	2,7%	Weak statement	2,2%
A/B issue	1,3%	A/B issue	1,4%	A/B issue	1,2%

Table 10.6: *Node label usage from different annotators.*

The results in table 10.6 are very nice, the label usage is very similar for each annotator. Only annotator 1 has somewhat different results for the labels "Weak statement" and "Open issue", but for the rest the results of the annotators are very nice. But the Twente Argument Schema also has relation labels and to see if the results for those labels are as good as these results I have done the same thing for the relation labels. The results are shown in table 10.7. A picture of the agreement amongst different annotators for a meeting is shown is appendix B

Annotator 1		**Annotator 2**		**Annotator 3**	
Positive	51,1%	Positive	54,8%	Elaboration	32,5%
Negative	14,8%	Elaboration	11,5%	Positive	28,1%
Option	11,3%	Option	11,3%	Option	15,8%
Subject to	5,9%	Uncertain	7,2%	Negative	7,6%
Elaboration	5,6%	Negative	6,0%	Uncertain	4,8%
Request	3,9%	Request	5,6%	Subject to	4,8%
Specialization	3,6%	Specialization	1,8%	Request	3,6%
Uncertain	3,5%	Subject to	1,7%	Specialization	2,7%
Option exclusion	0,5%	Option exclusion	0,2%	Option exclusion	0,2%

Table 10.7: *Relation label usage from different annotators.*

The results for the relation labels are not as good as the results for the node labels, but there is still some agreement amongst the annotators. The "Positive" label for example is used in more than 50% of all labels by annotators 1 and 2 and it is in second place for annotator 3, which has the label "Elaboration" in first place which is in second place for annotator 2. Another agreement which all annotators share is that the label "Option" is in third place and the label "Option exclusion" is in last place. When we divide the labels in two sets, one set being "Positive", "Elaboration", "Option", "Uncertain" and "Negative" and the other set being "Request", "Specialization", "Subject to" and "Option exclusion", you can see that annotators 2 and 3 both have the first set as top five and the second set as last four. Annotator 1 has something similar only the relations "'Subject to" and "Uncertain" are swapped in his list.

Another thing that has been done is that annotators 2 and 3 have both annotated a set of specified discussions and their results are compared. Annotator 2 has assigned 590 node labels and annotator 3 has assigned 557 node labels in these discussions and 358 of these labels were assigned by both annotators to the same text (the labels for the nodes can be different but the text contained in the node is the same). So that is an agreement of 61% and 64% respectively for annotators 2 and 3. In this set of 358 nodes the number of labels that are the same is 267, and that gives us an agreement of 75%.

The same thing can be calculated for the relations that both annotators have labeled. Annotator 2 has labeled 281 relations and annotator 3 has labeled 183 relations. First we look at the relations that are labeled by both annotators that have the same text contained in the source and target nodes and the labels (of relation, source and target) can be different here. In this way there can be found 37 relations and from these relations 25 have the same labels for source and target nodes but not necessarily the same relation label. So with the relations we have an agreement of 9% and 14% for annotators 2 and 3 respectively. When we also look at the relations labels and where they are the same, there are 14 relations that are completely the same for both annotators, and that gives us an agreement of 56%.

These results do not say very much but they do indicate that when the same situations exist for two different annotators, the same labels are applied in 75% of these situations for node labels and 56% of these situations for relation labels. But when we look at how often the same situation occurs for both annotators, especially with the relations, it is clear that they do not occur very often. Although this is not caused by the labelset it might be a good idea to specify additional guidelines of what should be labeled as nodes. When the situations are not the same for the annotators we have not looked at how different the situations are that do occur, this might also be something which can be done in future research. When we look at the agreement levels of the other annotating methods it can be seen that these depend very much of the corpus that is annotated. Since no annotations from other annotators were available for the transcripts annotated here it

was not possible to calculate agreement percentages for the other annotating methods.

Another thing that is looked at is at how the labels of the Twente Argument Schema relate to the labels that are used in the other annotating methods. With this is meant how often label x in annotating method a is used as label y in the Twente Argument Schema. In tables 10.8 through 10.11 results of this are shown. In the tables "X-issue" stands for all three types of issue in the Twente Argument Schema and with the RST relations an "S" at the end indicates the source of a relation and a "T" stands for target.

Twente Argument Schema label	RST label	Occurrences
X-issue	Question-answer-s-S	13
Statement	Question-answer-s-T	22
Statement	Otherwise-S	28
Statement	Otherwise-T	27

Table 10.8: *Label comparison between the Twente Argument Schema and RST.*

Twente Argument Schema label	Toulmin label	Occurrences
X-issue	Rebuttal	2
Statement	Claim	36
Statement	Datum	4
Statement	Rebuttal	4

Table 10.9: *Label comparison between the Twente Argument Schema and the Toulmin model.*

Twente Argument Schema label	DA label	Occurrences
X-issue	Info-request	17
Statement	Accept	27
Statement	Assert	55
Statement	Abandoned	9
Statement	Reject	4
Statement	Repeat-rephrase	2
Weak statement	Maybe	1

Table 10.10: *Label comparison between the Twente Argument Schema and DAs.*

Twente Argument Schema label	IBIS label	Occurrences
X-issue	Issue	16
Statement	Position	53
Statement	Argument	28

Table 10.11: *Label comparison between the Twente Argument Schema and IBIS.*

10.6 Evaluation & Conclusion

Since the advantages pointed out in 10.5 outnumber and outweigh the disadvantages I can say that the labelset that we have developed here is better suited to our needs than the other annotating methods I have looked at. So when argument structures need to be visualized the Twente Argument Schema is a better method to use. It does not suffer from the disadvantages of the other annotating methods like information that is missing from the transcripts of the meeting (Toulmin), the lack of relation labels (DAs), the lack of

node labels (RST) and the limited options for the place where a relation can be established (IBIS). But it is capable of dealing with discussions and can display a nice and clear visualized argument structure with a limited amount of labels which makes it easy to work with for the annotator and it has an expressing power that is great enough for users to understand what was discussed during the meeting.

CHAPTER 11

The ArgumentA annotation tool

For the testing of the Twente Argument Schema, we had to label some text with it. For this purpose we have built a special tool which is based on the Nite XML Toolkit. With the purpose of the tool in mind we have named it ArgumentA. The ultimate goal of this tool is to create a large set of annotated meetings which can be used in a number of situations, see 1.2 for what these situations are.

The ArgumentA tool consists of two parts. One part which is built to select the interesting discussions from the transcript of a meeting is called the 'Discussion Selector'. The other part is used to apply the labels to the text from the interesting discussions that have already been selected from a meeting and is called the 'AMI Discussion element coder'. Each of these two parts will explained below.

11.1 The Discussion Selector

The Discussion Selector is the part of the tool that most users will use first. It is used to search the text of the transcript of a meeting for interesting discussions, and by interesting discussions we mean discussions in which a number of speakers argue with each other and thereby making use of arguments to defend their opinions. These arguments can be used in a later stadium to create argument structures. When the tool is started you can first select a meeting from the corpus which is loaded into the tool and an annotator can be selected too (or a new annotator can be created). All of this functionality is already present in the Nite XML Tool and is not new for ArgumentA. After this is done the text from the transcript of the chosen meeting is loaded and discussions can be labeled. A screenshot of this part of the tool can be seen in figure 11.1.

As can be seen in the screenshot this part of the tool consists of a text window and a window to create and delete discussions with. The text window is called the transcription view because the text from the transcript of a meeting is displayed in this window. In the transcription view speaker information is present too. In this transcription view the fragments of text containing the interesting discussions can be selected and labeled as a discussion. The discussion can also be given a name and it is automatically saved in a file in the right format so that it can be used in the other part of ArgumentA.

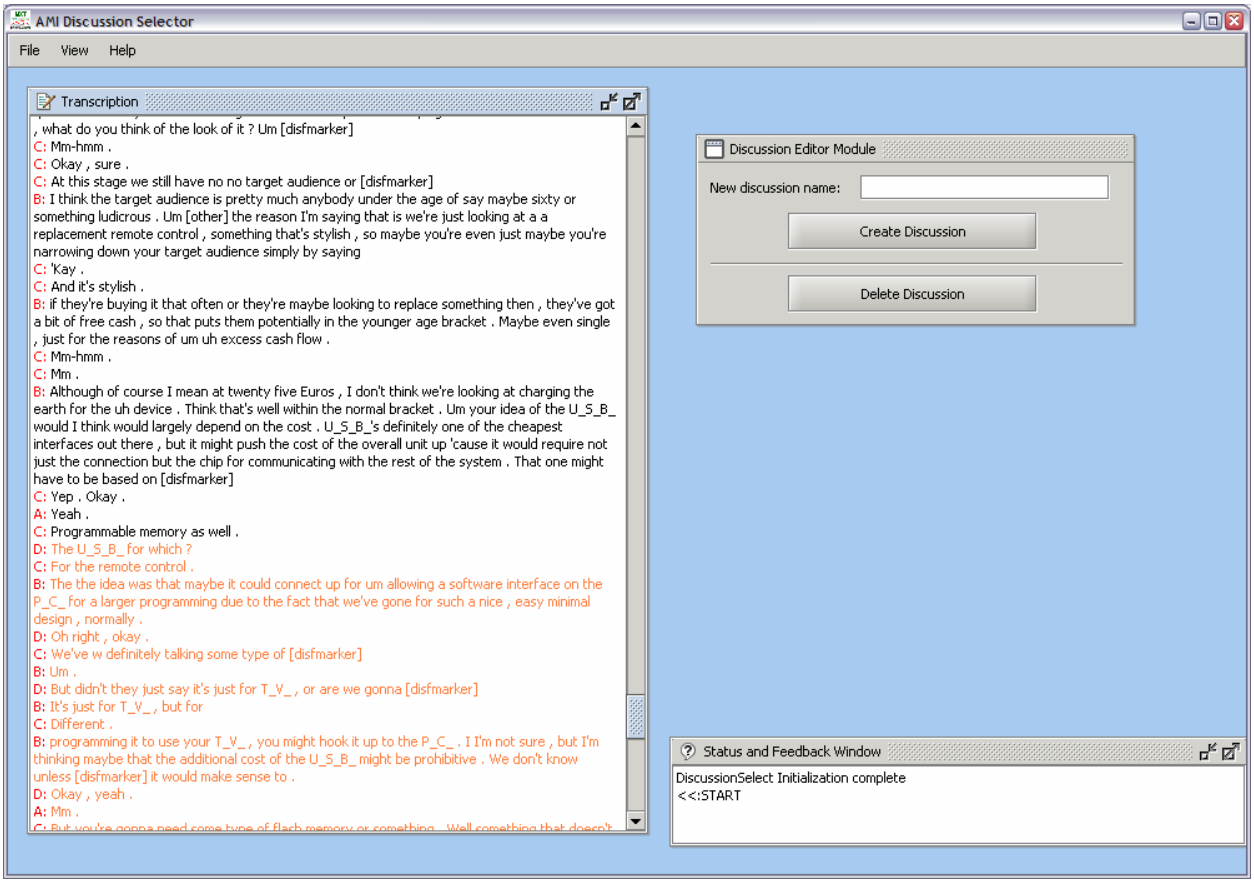

Figure 11.1: *The Discussion Selector.*

11.2 The AMI Discussion element coder

The AMI Discussion element coder is used to apply the labels from the labelset to the discussions that have been selected using the Discussion Selector. Only the text that is labeled as a discussion can be seen (and hence labeled) in this part of the tool. When no discussion is selected in the text of the transcript then no text is displayed. How a screenshot of this part of the tool looks can be seen in figure 11.2.

This part of the tool is built around three main components and some additional functionality. Each of the main components will be explained in more detail in the following sections and there will also be a section describing what the additional functionality is.

11.2.1 The transcription view

The transcription view is one of the main components of the AMI Discussion element coder. In this window the text that has been labeled as an interesting discussion is displayed and labels can be applied to it by selecting a fragment of text and assigning it a label. Just as in the transcription view of the Discussion Selector speaker information is shown here also. When the tool is started you can select a discussion and an annotator just as in the Discussion Selector and again the already present functionality of the Nite XML Tool is used. During the loading of the tool the text from the transcript of a meeting that has been labeled as interesting discussions is displayed in the transcription view in stead of all the text from the transcripts as was the case with the Discussion Selector.

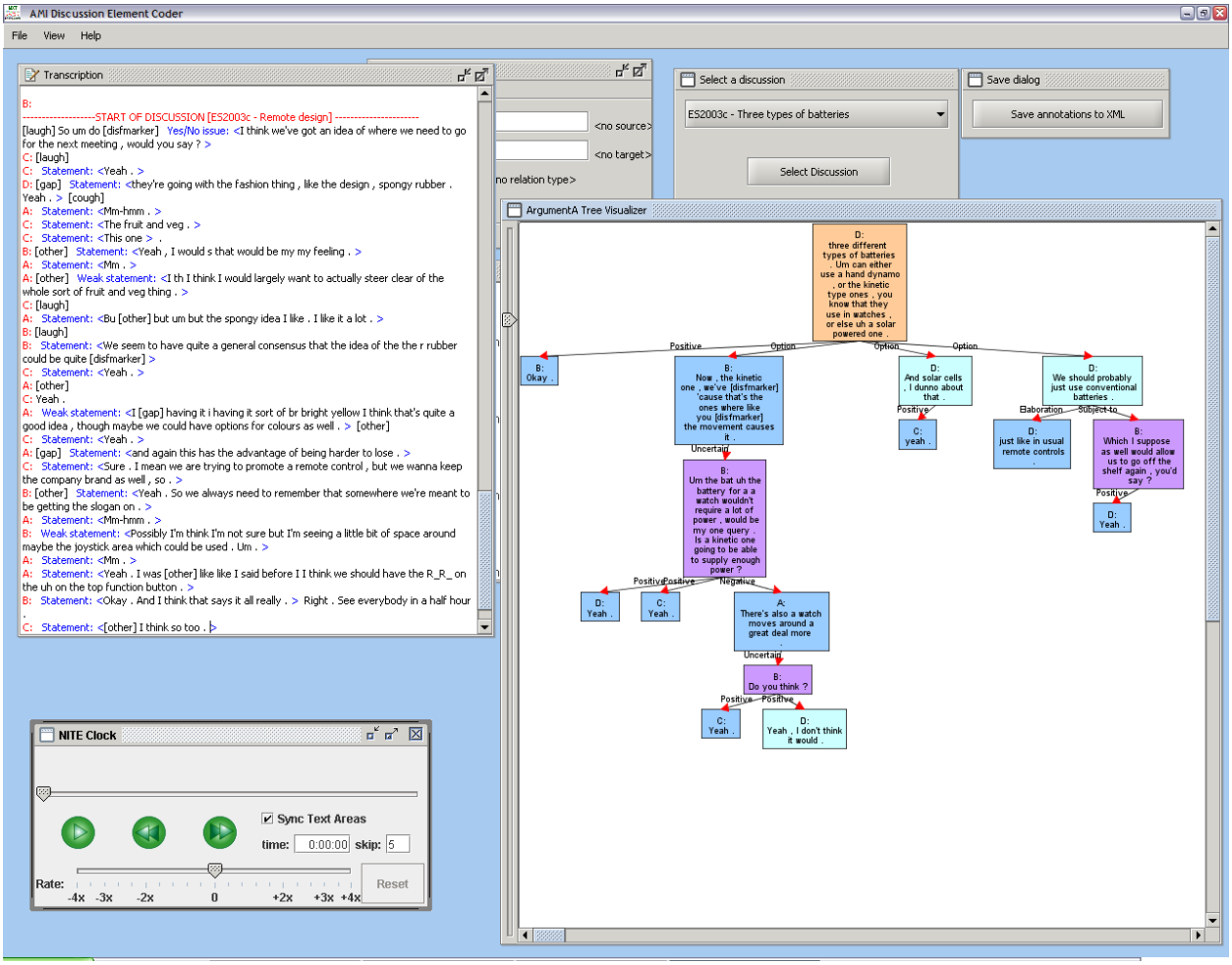

Figure 11.2: *The AMI Discussion element coder.*

11.2.2 The tree viewer

The tree viewer is another main component of the AMI Discussion element coder. In this component the argument tree is built during the annotation process. Each time a new node is created by assigning a label to a fragment of text a rectangle containing the text is created in this viewer. This rectangle is given a background color that is dependent of the type of node that is created, the types of node that can be created correspond with the labels from the labelset.

This component also offers some additional functionality in the form of being able to create relations between nodes in a graphical way. The user can drag the mouse from the source to the target of a relation while holding down the right mouse button, and in this way creates a relation which is visualized in the tree by an arc between two nodes. An example of the tree viewer containing a visualized argument structure (tree) can be seen in figure 11.3.

This part is very important in my assignment since the goal was to develop a way to visualize argument structures in a clear way 1.2. The tree that is ultimately created in this tree viewer has to accomplish all the goals that have been set in 2.3. From this visualization everyone should be able to figure out what the meeting was about and who though what about the decisions that have been made during the meeting.

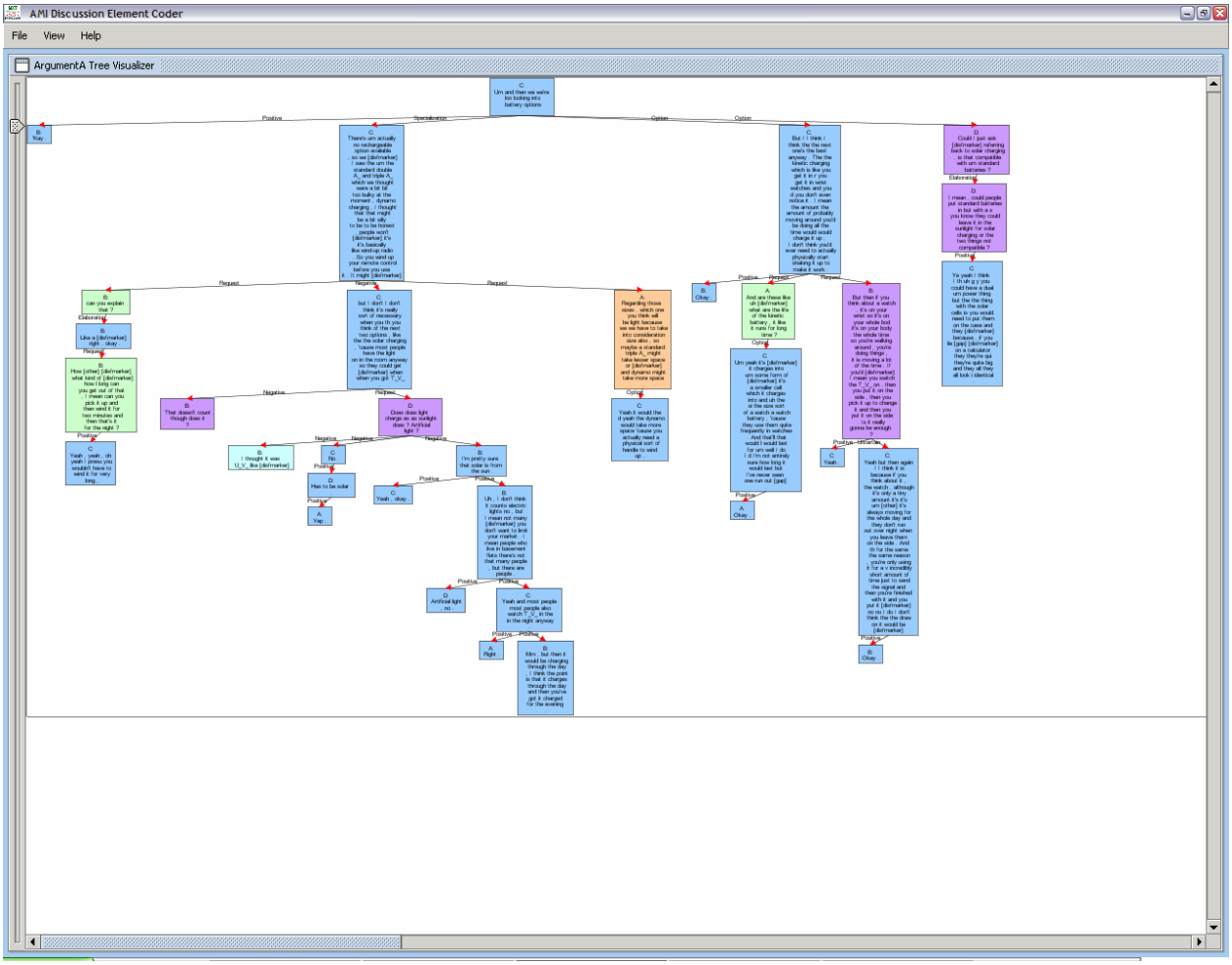

Figure 11.3: *The ArgumentA Tree Visualizer.*

11.2.3 The media player

The media player is the last main component of the AMI Discussion element coder. This media player is a standard component from the Nite XML toolkit but it can be used here in a very nice manner. A media player can be used to play the video of a meeting and the text in the transcription view is highlighted as it is spoken in the meeting video. For this tool we have altered its use a little bit, the media player is not used to play the video and the text simultaneously but as the text is spoken the argument structure is built up in the tree viewer. This should also aid in providing a nice way of visualizing the argument structure of a meeting.

11.2.4 Additional functionality

Next to the three main components the AMI Discussion element coder is also capable of some other things. For example it has the functionality to create relations between nodes using the Adjacency Pair window, which was already present in the Nite XML toolkit. In the same way nodes can be created using the Dialog Act window that was available in the Nite XML toolkit. The only thing that is different is the name of the elements that can be created with them, the way in which they function remains the same as in the Nite XML toolkit.

The tool is also capable of selecting discussions which have been labeled using the Discussion Selector. A list of discussions is constructed as the tool is loaded and the user can switch between those discussions using a dropdown list. Whenever a user selects a discussion only the text and the tree of that part of the transcript are shown in the AMI Discussion element coder.

The final piece of functionality the AMI Discussion element coder offers is the possibility to export the annotations that have been made to an XML file. This XML file just lists the identifiers of the nodes and relations, the speakers who have made the remarks, the type of node or relation, the start time of nodes, the text of nodes and the source and target nodes of a relation. Also the words that are in the transcript but have not been annotated are listed with the label type 'unknown'. This functionality is built in because the XML files that are generated this way can help in further research, for example the automatic recognition of nodes or relations.

CHAPTER 12

Future work

Now that a labelset has been developed which can be used to visualize argument structures from discussions, you could construct a system which is able to apply the labels from the labelset automatically. In this chapter I will give an example of such a system and which components could be incorporated in it. Also the data that is used by these components will be pointed out. For each component a short description is given of what it should do and how it should work. In figure 12.1 each of the components which I think are necessary for this system are shown. Also the information these components need to do their job is depicted in the figure.

12.1 Input and output data

Transcripts The transcripts contain the written text of everything that is said during a discussion. The text that is contained in the transcripts is used as input without changing or interpreting it and from this text the information that is needed is extracted. To make this task easier the transcript will be split into smaller pieces, which will be explained later.

I think it would be easiest to focus on the transcripts of the meetings, but according to Hirschberg and Litman [1993] it can be useful to look at the prosodic information of the speech signal too. In this article the authors look at the function of the cue word "now", and when they try to establish whether it is used in sentential or discourse usage they look at the intonational features, specifically pitch accent and prosodic phrasing. In Ang et al. [2005] it is also concluded that when prosodic as well as lexical information is used, the classification results are better than with lexical information alone. So whenever the results are very disappointing this could be solved by looking at the extra prosodic information.

Minimal text units Minimal text units are smallest the units of text which are used in the system. These minimal text units are the transcript of a meeting divided into little pieces. The boundaries of these little pieces depend on the theory that is used to annotate the transcripts with. In the sample argument structure in figure 1.1, all of the text in the nodes of the tree are minimal text units. In most cases these are the same as the speaker turns in the transcript. By a speaker turn I mean one or more remarks that are made by a single speaker without being interrupted by another speaker. When P0 for example had a few reasons to choose for Switzerland he could have given them all. If he pauses too long between two reasons and no one

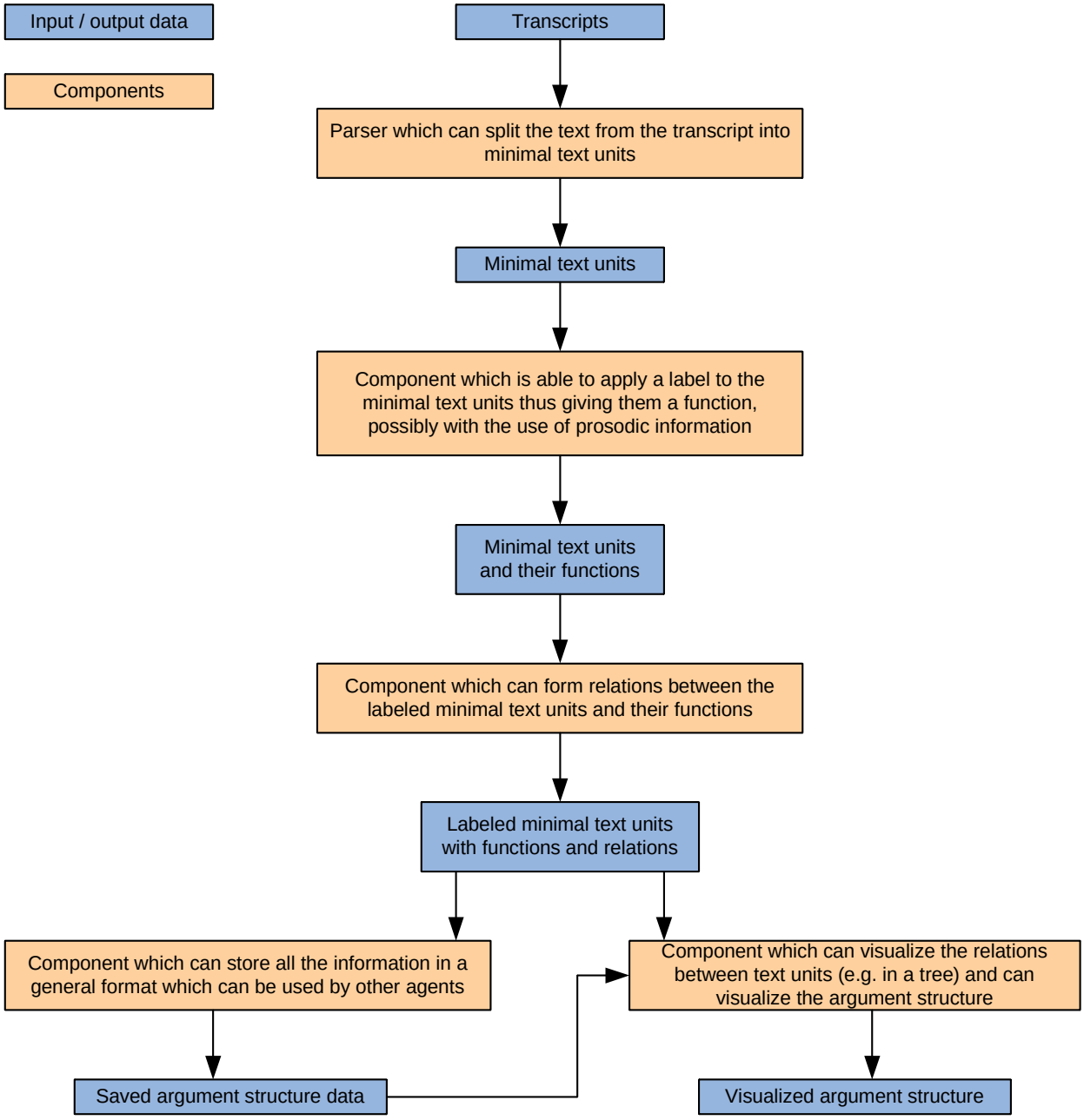

Figure 12.1: *A concept sketch of the components that could be used*

interrupts him you get several remarks of P0 following each other in the transcript. All of these remarks can be seen as part of a single speaker turn, although they are several different minimal text units. In a few cases speaker turns are split into multiple minimal text units in the sample discussion. This splitting is based on the punctuation and meaning of what the speaker says.

Minimal text units and their functions The minimal text units and their functions are the minimal text units which are assigned a function. The labels form the Twente Argument Schema can be used to assign each minimal text unit a function. For example take our sample discussion the minimal text unit "If the last

European championship will be replayed the winner will be?" could be labeled as an open issue, and thus it receives the function of an open issue.

Labelled minimal text units with functions and relations When the text units have been assigned a function relations can be established between them. Just as with the minimal text units and their functions, the relations that can be defined between labeled minimal text units are the relations the Twente Argument Schema. The general idea of what this data is, is a set of minimal text units and any relation that holds between them. For example in our sample discussion a negative relation could be defined between the minimal text units "Switzerland" and "No way".

Saved argument structure data The data that has to be saved is the data which holds the structure of the transcript which is applied to it while annotating the text. It would be very practical for this data to be saved in a file using a general format which could be used as input for others that also work with this data. One possible solution is to save all the data which is needed in an XML file.

Visualized argument structure This is the main result which the system should yield. This is the visualized form of the annotations that have been applied to the text by the system using the Twente Argument Schema. Visualizing the data can naturally be done in many ways, but I think the nicest form is a tree in which the arguments could be visualized as nodes and the relations between them as edges comparable to the example in figure 1.1.

12.2 Program components

Text interpreter This component should read the text from the transcripts that it receives as its input, and he should divide the text from the transcript into a list of minimal text units. To create the minimal text units this interpreter makes use of what is said in the meeting together with the punctuation of the transcript. Whenever the interpreter encounters e.g. a comma, it looks whether there occurs a topic shift at this point and if this is the case it marks this as a boundary for a minimal text unit, otherwise the text after the comma will also be added to the minimal text unit. In this way we can also make sure that minimal text units do not overlap each other. We do not want this to happen since the same spoken text could appear in two minimal text units and thus get two different functions, and this is not a very feasible situation because if there is a reaction to this remark, the reaction could be a very sensible reaction to one function and a completely meaningless reaction to the other function of the same minimal text unit.

Recognizer component The recognizer component receives the list of minimal text units as input and it should assign a node label to the minimal text units. The list of minimal text units and their node label is the output of this component.

This component could search the minimal text units for cue words such as "because". Whenever such a cue word is found the minimal text unit can be given a function. These functions can be used in a later stadium in which relations between minimal text units are established. An example of a list of possible cue words is presented in Hirschberg and Litman [1993]. Based on the cue word(s) found in the minimal text units a function can then be assigned to the minimal text units automatically. But fully automatic classification is difficult for an automated system, see for example the the results in Ang et al. [2005].

Maybe it is also possible to classify the minimal text units using a form of machine learning. For this the program has to have a training set which has to be built manually. This training set could exist of transcripts of meetings which are already processed to the stage in which the minimal text units and respective functions are known. In Rosset and Lamel [2004] a Memory Based Learning methodology is mentioned because it works well with small amounts of data, but they do not explain how this technique is used. This technique

is only used in the context of dialog act classification in the article, so I do not know whether it can be used here, since the Twente Argument Schema defines relations too. In Buø and Waibel [1996] they apply a parser that learns using neural networks to produce complex feature structures [Buø, 1996] and this technique could be used also.

When the results of this component are disappointing the prosodic information of the text from the transcript can be added to its input. This prosodic information can be used to recognize the function of a minimal text unit. This however is not my initial thought, but the fact that additional prosodic information yields better results had already been found by e.g. Hirschberg and Litman [1993].

Relations component This component is responsible for the assigning of relations between labeled minimal text units. It uses the list of minimal text units and their functions as input, and returns the list of the minimal text units and their functions together with the relations that are established between them. This component can look at the order of appearance of the minimal text units and their respective functions. When there are for example two minimal text units and they both have a function, this component can try to determine if a relation holds between the two minimal text units, and which relation it should be.

The relations between minimal text units can be determined by making use of "if-then-else like" structures. For example in our sample discussion we could look at the function of the minimal text units "If the last European championship will be replayed the winner will be?" and "Switzerland." we could say that if the first minimal text unit has the function open issue and the second one has the function statement, there could exist an option relation between them.

But when this approach seems incapable of establishing the correct relations between units, machine learning techniques could also be considered to link text units. The approach used in Buø and Waibel [1996] can be followed to accomplish this. In this approach a *Chunk Relation Finder* tries to link chunks of text with a relation, and in this assignment we could do the same with the minimal text units. They do not give a lot of details on how is works though so this can become tricky.

Storing unit The only task this component has to accomplish is the storing of the data in a format which is independent of this program. Whenever this component receives a set of minimal text units with their functions and the relations between them, it should be able to convert this data into a format which can be put in an XML file. This XML file can serve as a data source for others which can use the data from it in any way they seem fit.

Visualizer This component should handle the visualization of the minimal text units and their functions and relations in a from which resembles the one in figure 1.1. As input it needs the list of minimal text units together with their functions and the relations between them, and its output is a visualized argument structure.

The visualizer should draw a 2D tree to present the visualized argument structure to the users. The process of creating a clear tree involves a lot of puzzling and playing around with the layout of the tree. An option of improving the tree could be to make it interactive. Interactive means that whenever the user clicks on a node the user gets the tree which is a child of that node and represents the argumentation behind that node. So not all the information is presented in parallel but it is presented in small chunks in which you can select your area of interest and look into it in more detail.

CHAPTER 13

Conclusion

In this assignment various existing annotating methods are examined to see if they can be used for the visualizing of argument structures that are present in meetings. As can be seen in the chapters 5 through 8 these methods all have some disadvantages and are not suited for the goal of this assignment. The main reasons are listed here for clarity.

- RST does not offer any labels which can be used for the nodes in an argument structure.
- DAs do not offer any labels that can be used to relate the different DAs with each other, which is necessary for use in an argument structure.
- Toulmin models make use of information that is not present in everyday meetings. This information is present in the minds of the participants and not in the transcripts so it cannot be shown in an argument structure.
- IBIS comes close to what is needed here but it suffers from static relations and node labels, a relation can only be placed between a fixed pair of node labels and this limits the expression power of the argument structure.

Maybe a combination of a number of these methods could prove useful, but the set of labels becomes very large in this case and because of this the work of the annotators does not get any easier too.

Because the existing methods could not solve our problem, we have developed a new annotating method called the Twente Argument Schema. With this new annotation method we have run various tests and the results look very promising. The Twente Argument Schema does not suffer from the disadvantages of the other technique since it is designed with the goal of this assignment in mind. For the reasons mentioned in sections 10.5 and 10.6, this method is the best one for visualizing argument structures.

This method could also serve as a framework for further research, for example this could be something like the automatic assigning of the labels for nodes and relations. It could also be used for an automated chairman which could guide the discussion in a certain direction based on the arguments that are used in a discussion.

APPENDIX A

Twente Argument Schema labels

In this appendix all of the labels and their definitions are described in detail. First the table with descriptions of the unit labels (node labels) is shown and the second table contains the relation labels.

Unit Labels	
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.
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.

Relation Labels	
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.
Uncertain	This 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.
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.
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 open issue, 'What do you mean by modifying the environment?' can be labelled as a elaboration relation.
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.
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.
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.

APPENDIX B

Annotation overlap

In figure B.1 the annotation of a discussion is shown for two different annotators. In this figure you can see where the annotators agreed on the label that they used. Where the color is the same in the upper and lower bar the annotators have used the same label. Which color represents which label is also shown in the figure. As can be seen the annotators agreed on a large part of the discussion and this is a good sign for the Twente Argument Schema.

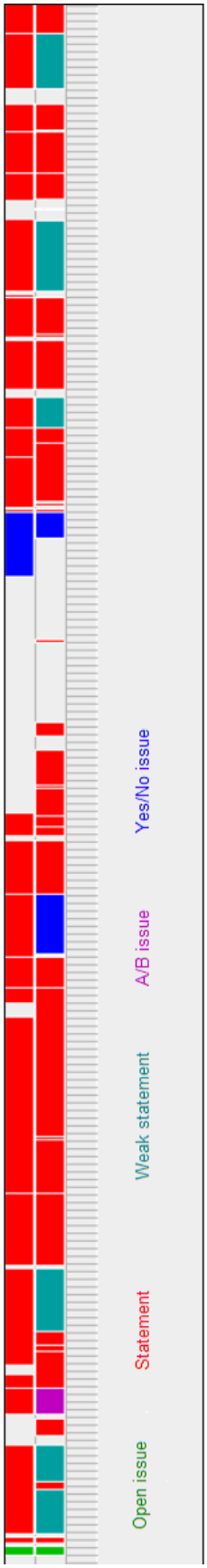

Figure B.1: *The agreement amongst annotators for a discussion.*

Bibliography

J.F. Allen and M.G. Core. Draft of damsl: Dialog act markup in several layers. http://www.cs.rochester.edu/research/cisd/resources/damsl/, September 1997.

J. Ang, Y. Liu, and E. Shriberg. Automatic dialog act segmentation and classification in multiparty meetings. In *Proceedings of the 30th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2005.

Answers.com. *Wordnet 1.7.1*. Princeton University, 2001. URL `http://www.answers.com/topic/argument`.

Answers.com. *The American Heritage Dictionary of the English language*. Houghton Mifflin Company, fourth edition, 2004. URL `http://www.answers.com/topic/argument`.

M.C. Beardsley. *Practical Logic*. Prentice Hall, 1950.

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

F.D. Buø. *FeasPar - a feature structure parser learning to parse spontaneous speech*. PhD thesis, University of Karlsruhe, July 1996.

F.D. Buø and A. Waibel. Feaspar: a feature structure parser learning to parse spoken language. In *Proceedings of the 16th conference on Computational linguistics*, pages 188–193. Association for Computational Linguistics, 1996.

J. Conklin and M.L. Begeman. gibis: a hypertext tool for exploratory policy discussion. *ACM Trans. Inf. Syst.*, 6(4):303–331, 1988. ISSN 1046-8188.

M.G. Core and J.F. Allen. Coding dialogs with the damsl annotation scheme. 1997.

T. Govier. *A practical study of argument*. Wadsworth Publishing, sixth edition, 2005.

B.J. Grosz and C.L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.

C. Hair and C. Lewis. Are argument representation schemes useful? Technical report, University of Colorado, department of Computer Science, June 1990.

J. Hirschberg and D. Litman. Empirical studies on the disambiguation of cue phrases. *Comput. Linguist.*, 19(3):501–530, 1993. ISSN 0891-2017.

S. Isenmann and W.D. Reuter. Ibis - a convincing concept ...but a lousy instrument? In *DIS '97: Proceedings of the conference on Designing interactive systems*, pages 163–172, New York, NY, USA, 1997. ACM Press. ISBN 0-89791-863-0.

W. Kunz and H.W.J. Rittel. Issues as elements of information systems. 1970.

W.C. Mann, C.M.I.M. Matthiessen, and S.A. Thompson. Rhetorical structure theory and text analysis. In William C. Mann and Sandra A. Thompson, editors, *Discourse Description: Diverse Linguistic Analyses of a Fund-Raising Text*, pages 39–78. John Benjamins, Amsterdam and Philadelphia, 1992.

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

D. Marcu. Building up rhetorical structure trees. In *AAAI/IAAI, Vol. 2*, pages 1069–1074, 1996.

D. Marcu. The rhetorical parsing of natural language texts. In *Proceedings of the 35th conference on Association for Computational Linguistics*, pages 96–103. Association for Computational Linguistics, 1997.

D.C. Marcu. *The rhetorical parsing, summarization, and generation of natural language texts*. PhD thesis, 1998. Adviser-Graeme Hirst.

Merriam-Webster. *Dictionary of Law*. Merriam-Webster, 1996. URL `http://www.answers.com/topic/argument`.

S.E. Newman and C.C. Marshall. Pushing toulmin too far: learning from an argument representation scheme. 1991.

M. O'Donnel. Rsttool – an rst markup tool. http://www.wagsoft.com/RSTTool/index.html, July 2003.

R.J. Rienks, D.K.J. Heylen, and E. van der Weijden. Argument diagramming of meeting conversations. In *Multimodal Multiparty Meeting Processing, Workshop at the 7th International Conference on Multimodal Interfaces (to appear)*, Trento, Italy, 2005.

S. Rosset and L. Lamel. Automatic detection of dialog acts based on multi-level information. In *icslp*, pages 540–543, Jeju Island, October 2004. URL `ftp://tlp.limsi.fr/public/TuB401o.2_p540.pdf`.

J.R. Searle. *Speech acts*. Cambridge University Press, 1969.

S. Shum. *A cognitive analysis of design rationale representation*. PhD thesis, Department of Psychology, University of York, December 1991. Chapter 1 - Design rationale's design roots.

S.B. Shum and N. Hammond. Argumentation-based design rationale: what use at what cost? *International Journal of Human-Computer Studies*, 40(4):603–652, 1994. ISSN 1071-5819.

P. Smolensky, B. Fox, R. King, and C. Lewis. Computer-aided reasoned discourse or, how to argue with a computer. *Cognitive Science and its application for human-computer interaction.*, pages 109–162, 1988.

A. Stent. Rhetorical structure in dialog. In *Proccedings of the 2nd International Natural Language Generation Conference (INLG'2000)*, June 2000.

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

J.H. Wigmore. *The Principles of Judicial Proof*. Little, Brown and Company, second edition, 1931.