Workshop on Annotation Architecture and Software Tools
for Multi-Media Language Resources and Large Corpora

LREC Preconference Workshop

Summary

P. Wittenburg, H. Brugman, D. Broeder

Also these two workshops ran under the flag of the new EAGLES/ISLE project, i.e. they were organized to define the actual needs of the community to be tackled in the project. At the end of this note we will draw some conclusions.

Contributions
It is not possible to be comprehensive and mention all contributions of the two workshop parts. We will limit ourselves to contributions and comments which are related to our tool-oriented work at the MPI. Some contributions focussed on the encoding of multi-modal behavior. It is fully clear that we don't have good insights about what people are doing in this area and that the EAGLES/ISLE project has to work on this. In this summary we will not comment on these contributions although they are very important for many of us.

ATLAS
The Atlas concept/architecture was introduced. It is based on an API which offers all functionality to deal with relational database structures implementing LDC's formal model (acyclic directed graphs). On top of this API various applications and APIs are planned which make use of this API. The architecture mentions AIF (ATLAS Interchange Format) files on the same level as the relational database, but operations are not symmetric: AIF is only an import/export format which can either be generated or consumed. LDC's annotation graph model is well-known, it was generalised to be able to cope with higher-dimensional cases. The term region was introduced to denote a stretch in some n-dimensional space, so a time interval is a stretch in a 1-dimensional space, but a gesture occurs in a spatial as well as in a time stretch. Based on this an ATLAS Object Model was developed which served as basis for developing the API. Currently, LDC people try to get the API stable, design the AIF, and start adapting/creating tools which work with the API. In another talk form LDC it was reported that a query language is being developed which seems to make use fo the described API. Also the well-known Transcriber tool was told to support the API. If these tools were ready they were the first making LDC's ideas available in an operational form. ATLAS is one part within the TalkBank initiative which aims at understanding the needs of a large variety of disciplines and creating a universal format and a set of tools operating on it.

Comment
LDC has done a great job with analysing the various formats and describing a formal model. It helped all of us to clarify concepts and can serve as a reference. The results are similar but more comprehensive compared to a study which was made at the MPI as basis for the EUDICO abstract corpus model years ago. And, of course, the community is highly interestes in the results of the TalkBank project. Until now LDC has the universal formalism and ideas (some code, however, not yet stable) of how to implement this with the help of a relational database structure covered by an API. More has to be available to make better judgements about the impact of this work. Until now it is not clear for us what exactly will be accepted by the community. The AIF seems to be much more important than the API, since it would allow other developers to independently build tools and in doing so support the AIF. The AIF also is the documentation format. However, the AIF is not yet specified. Using the API might only be interesting for a few developers, if the underlying machinery (database engine) provides more efficient access as other methods. Relational databases, however, are fairly common. Much excellent analysis work has been done by the LDC people until now, but the hard programming results have to come. A format unification could be achieved when the TalkBank project would be able to describe a generic AIF in not too far time.

MATE
The MATE spoken corpus annotation program is demonstratable, although it has still some bugs[1]. SDU presented a tool which has as one of its core concepts the so-called coding modules. A coding module is a realization of an encoding scheme and it can be easily (in normal cases) specified by the user. MATE is delivered with a set of ready-made coding modules. These coding modules are used in two ways: (1) They are

[1] This is not to blame the developers, since we know that bug-free programming is a very hard job.

used to constrain the annotation and (2) they are used to generate DTDs which describe the structure of the XML files which MATE can handle. MATE also uses XML as an interchange format, i.e. internally MATE operates with a relational database. MATE is delivered with a powerful search tool which allows the user to do IR by using structural information and some statistics. MATE comes with a number of well-designed user interface components.

Comment
The MATE people have demonstrated a tool with a nice and to a large extent convincing user interface. Surprising for us was the decision that MATE cannot be used as transcription tool. This is supported by the fact that the speech viewer is comparatively simple and attached. You need a first transcript and then can carry out further annotations. MATE is the first annotation program (as far as we know) which implemented an XML-import/export module. However, MATE does not apply the stand-off format, this decision is coherent with its goal to function as annotation tool based on a ready transcription. MATE might therefore have problems with multiple independent streams (channels) as they occur in multi-media annotations. Nevertheless, MATE is (almost) ready and may be used by many as a tool for manual annotations. A problem might be the limited number of input filters currently available (Xlabel, BAS). Some design decisions might make it difficult to extend MATE to a full-fledged multi-media annotation and exploitation tool, operating in distributed environments as is required for the work in our institute. Nevertheless, we can learn a lot from the MATE project.

Ghorbel
The most complex annotation situation seems to be given in TV studios where complex workflow processes influence the way annotions emerge from multiple interacting annotators. Complex relations between the different annotations are given such that the EPFL colleagues decided to use a knowledge base on top of the annotation system.

Comment
To us it is not clear whether this application introduces new types of structural phenomena in the annotation scheme which were not yet been described by others. If this complexity is covered by what has been described already, then the knowledge base can be seen as complementary, but some of the tools currently under developent and presented at the workshop should be able to cope with the annotation task. MPI investigation indicate that EUDICO's internal abstract corpus model is rich enough to handle such situations. But we are not yet sure about this.

CELLAR
With CELLAR a spin-off of the challenging but not finished Lingua-Links project was presented. The user can specify his/her data model and both a DTD as well as an SQL schema is created. The DTD could be used by an editor which is used to create XML-structured data. Such XML files can be imported to the CELLAR system which is based on a relational database engine. Applications can operate with the database. For Cellar it is claimed that the model can cope with data objects having many simultaneous properties and highly interrelated data requiring to encode associative links between related pieces of data.

Comment
In principle similar to MATE, Cellar offers a possibility to specify annotation schemas. It does so by creating both a DTD for defining the structure of an XML document as well as that one of a relational db. The idea is excellent. It seems that the designers had typical text-based annotations in mind and did not think of multi-media environments. It is not clear to us whether CELLAR can be used for complex structured annotations as we know them for for example gesture databases. It would make sense, if CELLAR would be available as a specification tool which is independent from concrete relational DBMS (since people are using different systems) and if it would be easily integratable into annotation tools. For us it is also unclear whether CELLAR can cope with dynamic environments, i.e. environments where people frequently change the annotation structure.

Romary/Lopez
LORIA people presented a layered framework to create annotation structures and to transform them into efficient internal representations. In the focus of their work is the term “free of redundance” which is similar to the term “normalized structure” in the field of database design. The first step is to create a “Relational Ressource Organization Model” which describes the set of resource entities and the set of relations between entities. Resource entities are thought to be independent, i.e. basically every annotation tier has to be represented in a separate file. A tier such as an orthographic transcription or a original text is the basis, i.e. all annotations refer

to words or group of words of this basic tier. Based on this model an XML structure is derived where each independent resource element is stored in one XML document. This comes close to what is known as stand-off model. Also the relations between the resource elements are stored in a separate XML document. Since this set of XML documents does not lend itself for efficient processing, a Finite State Representation mechanism is derived which is free of redundance. This is used to implement an efficeint access machinery.

we speak about channels or in MPI's terminology about independent streams

Comments
The approach to first build a good model of the data and from that derive an orthogonal XML structure seems to be helpful. However, it presupposes that the person exactly knows which kind of linguistic units will occur. In a dynamic environment which is often the case it is not known beforehand what users will encode, i.e. it is not possible to generate a model which goes down to the linguistic units. It is claimed that the redundancy-free FSR mechanism can be used for efficient access. However, this can only be true for certain type of access patterns. Increasing redundancy in general makes access faster. FSR are theoretical concepts which have to be mapped to physical database structures. Since the paper does not tell how this is done, it cannot be seen which type of access might be efefficientnd which not. So, although the conceptual procedure is convincing it is not clear to us whether this framework is generally applicable.

Ide
Nancy gave two papers: one mentioning requirements for the work we all are doing and one explaining the possible gain in applying XML. The first was very useful as a general reference and will not be commented further. The second reported about extended functionality in XML to create links between annotations such as XLink, XPath, and XPointer. These mechanisms may have to be applied when complex annotation structures have to be represented within the XML formalism. XML transformation possibilities such as XSL and XSLT are more on the tool side where we don't know yet where these can be applied and whether they are appropriate in multi-media environments. XML schemas will be of large importance to better describe (and constrain) the contents of XML documents. However, XML Schemas are not yet accepted as an international standard and they are still subject of changes.

EUDICO
EUDICO is MPI's baby and will not be commented by us. It is ready as player version to demonstrate its basic concepts. Still it has some functional gaps before it can be described as a full-fledged annotation and exploitation tool for multi-media langauge resources. Since it is still under development, it is not yet debugged. Nevertheless, it is one of the few operational true multi-media tools.

Discussion
The discussion after the talks and at the end of the session resulted in a number of interesting points:

- One major question focussed on the value of XML. It was generally agreed that XML will be very important as an open exchange format. The structure of a document will be well-described such that everyone can read XML-documents and use the data in some form. Therefore, it is also good for long-term documentation. However, much data will remain as it is and will not be converted into XML files. Also some of the non XML formats (TIPSTER, …) are much more suitable to the specific work people are doing, so there is no reason to step over to another format. However, tool developers should provide XML import/export modules. The main argument for using XML often is the availability of tools. However, in case of multi-media environments there is nothing. Further, there is the clear statement from LORIA people that XML is not a good modeling framework.
- There is still a debate whether XML structures can directly be used for processing. All major tool builders currently tend to provide XML import/export modules, but they internally often use relational databases or in case of LORIA a FS representation. One question which adressed the speed of retrieval was not answered although it is an important one.
- Extensibility of annotations is an important issue. Often people don't know beforehand how they will encode linguistic phenomena, i.e. there must be ways for individuals to enter just what they want and define arbitrary references and add arbitrary comments.
- The stand-off model seems to be widely accepted for XML documents. It implies that independent annotation layers are stored in different files and that links are set between these files by using structure pointers.
- Often the term "object" is used when people speak about structure elements in XML documents. This could lead to irritations, since one of the problems some tool-builders have is exactly how to map rich object models to linear document structures. This mapping is not trivial.

- It is a general agreement that the tools or formats should not impose biases towards a certain linguistic unit. This implies that the annotation structure has to allow the user to define new tiers where he/she can choose new stretches (spatial or temporal) and label them. This was already well-described in the paper from SB&ML.
- There is a debate in how far tool developers have to provide "stereotypic" views on the data or whether formalisms such as XSL can be given to the user to have him/her create their own view on the data. In a multi-media environment only stereotypic viewers will work, i.e. viewers which were defined by the system developer. Most people see XSL as a way for specialists to easily create other type of layouts for textual documents. So XSL could form a medium layer for the specialist to create new views in the case of textual data.
- There was a short discussion about the usage of SMIL. As far as could be seen from the documentation so far SMIL is a tool for making synchronised representations via the web, but it can not be seen as a multi-media analysis and exploitation tool which would serve our needs.

Summary Statement

- Together with Nancy Ide we organized two workshops about annotation structures, encoding schemes, and the architecture of tools. While part of the talks were dedicated to rich textual structures other were focussing on the special requirements when working in a multi-media environment. The requirements are partly different.
- A great problem is seen in the fact that although we speak about very similar and largely overlapping things, still the terminology is very different. This refers to the statement of HT about the non-existing ontology of our field. The area in which we are active is very dynamic.
- This dynamic situation is the reason that makes us sure that we need the competition of different approaches. This is true for the representation formats as well as for the analysis and exploitation tools. LDC did do a great job with describing the various phenomena in complex annotation structures and deriving a common logical framework for annotations. But there is no doubt that we will have to try various formats in the area of multi-media corpora and that we need a variety of tools to create them and to exploit their content. New APIs such as that one from LDC are emerging, but we don't know yet whether they will be sufficient and whether it will do what we need. The availability of open exchange formats will help us a lot on the way to re-use language resources, but there is stil a long way until suitable XML-structures for multi-modal content will have stabilized.
- As already mentioned at the beginning some projects started with annotating multi-modal behavior. But there are stil many open questions in for example encoding gestures. What we need therefore is an overview about what people are doing in this area, how they are encoding multi-modal behavior, and what kind of analysis they intend to carry out. This may end up in suggestions for new projects to achieve greater coherence and thereby improve re-usability. On the other hand we need flexibility in this area, since we just started encoding multi-modal behavior.
- Only briefly during the workshop we spoke about how to integrate media and how to do streaming. This area is suffering from high dynamics on various levels. On the signal encoding level we have the trend form MJPEG (->Cinepak) to MPEG1, MPEG2, and MPEG4 which will keep those busy who have to build multi-media tools. On the higher level we have container APIs such as Quicktime and player APIs such as Java-Media-Framework, and much incompatibilities with respect to file formats. Driven by the media community we also have media annotation initiatives such as MPEG7 and Dublin-Core which will influence what we are doing to a certain extent.
- We have seen a number of architectures of software tools (MATE, GATE, ATLAS, EUDICO, CALIN, CELLAR, …). It seems that a multi-level structure is widely accepted: (1) At the physical level systems mostly operate with a relational database as internal format for efficiency reasons. Most tend to support an XML-based format for import/export. Few also support native formats such as CHAT. (2) Although terminology differs between the teams the essential point is that after methods of abstraction a universal layer was introduced. ATLAS for example speaks about an API which is based on a generalised object model. EUDICO speaks about an abstract corpus model. The difference in these two cases is that ATLAS makes the logical level available as an API, while in EUDICO the abstract level is part of the kernel. (3) Consequently, the next level, the application level, is different as well. In ATLAS applications are separate programs on top of the APIs, i.e. due to a lack of API descriptions it is not yet clear what the shared machinery is. In EUDICO there is a kernel based on the abstract model and applications are realized as class hierarchies on top of this machinery. GATE is designed for a somewhat different purpose. Its main objective is to allow language engineers easily add NLP modules to an existing framework which provides common functions such as data access and visualization. It makes use of the TIPSTER format which is

widely accepted in the LE community and has proven its usefullness as a component framework at many sites. MATE's architecture is not yet fully clear to us. It seems that the search module was built separately from the annotation environment although all functionality is available via a unifying user interface. It is not clear to us whether there is a common API designed for such components or whether the logical description of the database is the common interface. CELLAR's major intention is the data modeling interface which generates structure descriptions for relational db as well as for XML documents. With respect to the architecture of the CALIN we cannot make statements yet, since the talk was not about such aspects.

- A short discussion was about the question to what extent we have to re-invent the wheel. It is good to have a limited number of data models which is the gasoline in our field. Therefore the analysing work about common formats is very important. Still due to the dynamics we will be far away from a situation where we have narrowed down the number of formats. In the area of multi-media annotations we see a number of activities such as TalkBank, MPEG7, Dublin-Core, EAGLES/ISLE etc all dealing with partly similar type of questions, but raised from the perspectives of different communities. Additionally, we see the many different projects which still use there own formats from various reasons which are sometimes mission critical. Of course, we need to come to unification, but it will take a while. With respect to the machinery which makes use of the gasoline we believe that we need competition of different concepts. The interests are differing and we are far away from being able to design a framework which will handle all of them.
- Some "users" argued that it would be very helpful for the field to have unbiased descriptions of what the tools can and especially what they can't do. It was also required that it would be very useful to have demo examples (possibly in the web) to make it easy for the user to understand the main concepts.

EAGLES/ISLE Project

From the workshop we can extract a few major tasks for the EAGLES/ISLE project:

- We should start making an overview about the encoding schemes used in annotations of multi-modal behavior.
- The project should make an analysis of the architectural basics of the major tools and describe the available functions. It would also be useful to select a number of corpora such that the tool builders can show how the tools can deal with such corpora. The goals must be that the users can easily understand what the tool can do for them and that the professionals get a deeper insight about structural phenomena and requirements fo the community.

Comments and questions should be addressed to ISLE@mpi.nl