USING OBJECT BASED TECHNIQUES TO STUDY TURKIC TEXTS

Münevver Tekcan[*]

Abstract: *Word processors can be used to write letters and produce documents. Presentation programs can be used to produce presentations at conferences. Programs as these can be used in the study of Old Turkic texts. These tools are collectively known under the heading Information Technology. This is an ill defined term but behind it are methods that are becoming formalised. These methods break up everything into abstract objects and the relationships between them and the actions performed by these objects. This paper tries to use some of the techniques used in Information Technology and apply them in the study of Old Turkic texts. As a practical application, the Orkhon Stones are used. It is suggested that by using these techniques Turkic texts can be understood in a different context and add to the knowledge associated with these texts.*

Keywords: *Old Turkic texts, Information Technology, objects, computers.*

New Tools for Old Texts

Information Technology can process Old Turkic manuscripts or any document. The relationship between the written word and the reader is linked to the tools involved with its production and reception, the traditional paper paradigm has shifted. For example, with *the Orkhon Inscriptions*, the text is influenced by more than the words. *The Orkhon Inscriptions* were inscribed on stone and displayed in a public place the relationship between reader and text was governed by tools and the materials used as well as the text.

This paper will concentrate on this methodology. I will try to use techniques that have been developed from *Information Technology* (Rumbaugh, James, Ivar Jacobson, Grady Booch: UML) and apply them to the study of Old Turkic manuscripts.

Methodology and Abstraction

A methodology is the application of a method. A methodology of studying old Turkic texts is not a method of study it is how the method is used and how it affects what is studied. For example in order to study the language used in a manuscript linguists use concepts such as grammar and semantics. Grammar uses elements such as nouns and verbs. In terms of literature concepts such as allegory

---

[*] Doç. Dr.-, Kocaeli Üniversitesi, Fen-Edebiyat Fakültesi, Türk Dili ve Edebiyatı Bölümü.

and metaphor are used. Knowledge of a subject involves being able to abstract the subject and then being able to manipulate the objects you have abstracted. Mathematics is pure abstraction, numbers are an abstraction, computer science has used abstraction and Information Technology has started to realise the application of abstraction.

In terms of a methodology of studying Old Turkic texts items such as manuscript, transcript and grammatical index are classed as documents and the person doing the study is a user. Documents and users are both objects. Generating a grammatical index by computer is an action.

Relationships or actions are formed between objects and each party of the relationship plays a role to play in that relationship. When someone reads a manuscript the action is 'reading', the person doing the reading is a 'reader' and the manuscript is the 'text'.
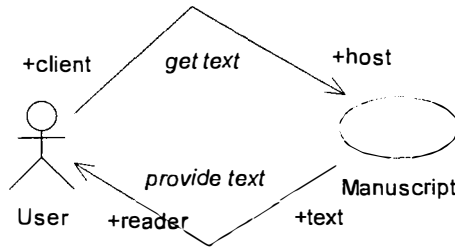
Methodology and Knowledge

The computer reads a transcript and writes a grammatical index. In order to make sense of the index I consult journals, talk to experts or do some basic research. Finally I should be able to take advantage of the grammatical index and have a richer understanding of the manuscript which I can share with a wider audience.

The methodology implies more than using a tool to generate grammatical indexes. A methodology to study old Turkic texts should include generic knowledge objects which link the generation of grammatical indexes to knowledge in a wider context.
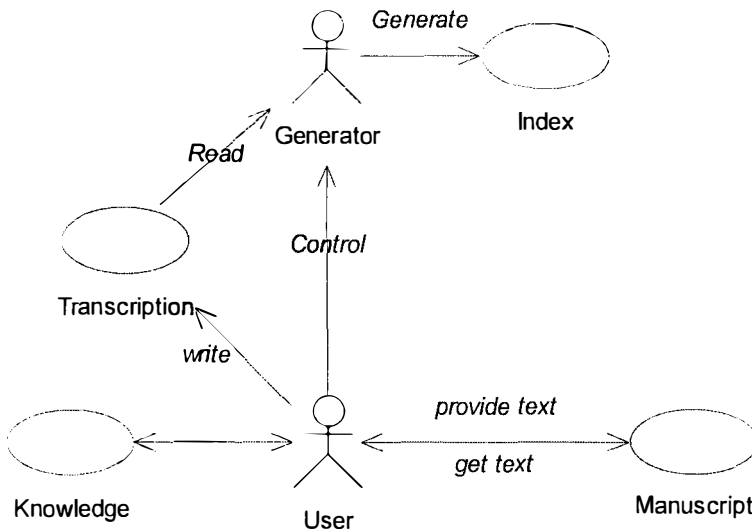
UML Diagrams

UML or Unified Modelling Language is a visual language to model systems or collections of objects that work together to achieve something. It is a language in the sense that it has its own visual syntax and semantics. It is a

modelling language in the sense that diagrams signify an aspect of a model and it is unified in the sense that brings everything together different techniques and knowledge into one system. I am using UML because it is a good application of a methodology where I can explore and build abstract models.
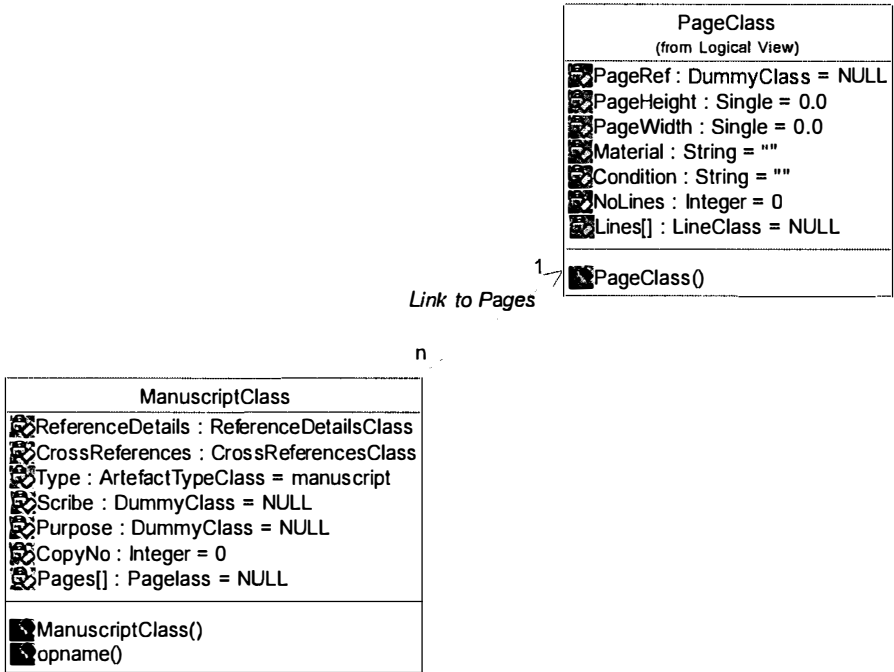
Use Cases

Diagrams called *Use Cases* can show the basic methodology in terms of people and machines like computers doing something. In our case there is one person called a user and one computer program called the index generator. In the diagram, these are represented by stick figures. Ellipses represent what the stick figures do or how they change the environment. In the diagram, the user reads the manuscript, writes a transcript and also tells the index generator what to do. The index generator generates an index. Also shown in this diagram are associations or relationships. Each relationship connects our stick figures to an object not shown in the diagram. Labels show the roles of the two players in each relationship relative to the stick figure.

The diagram shows that the user reads a manuscript and writes a transcript. The user takes control of the index generator which reads the transcript and writes

a grammatical index. The diagram also shows that the user acts as a point where knowledge is shared with the world. The operations that the stick figures do are listed underneath the icons. Even though the diagram carries a lot of information, it has nothing about the documents or text being studied or the timing of events.

Class Diagram

```
┌─────────────────────────────────────────┐
│               PageClass                  │
│            (from Logical View)           │
├─────────────────────────────────────────┤
│  PageRef : DummyClass = NULL             │
│  PageHeight : Single = 0.0               │
│  PageWidth : Single = 0.0                │
│  Material : String = ""                  │
│  Condition : String = ""                 │
│  NoLines : Integer = 0                   │
│  Lines[] : LineClass = NULL              │
├─────────────────────────────────────────┤
│  PageClass()                             │
└─────────────────────────────────────────┘
```

1

Link to Pages

n

```
┌─────────────────────────────────────────┐
│             ManuscriptClass              │
├─────────────────────────────────────────┤
│  ReferenceDetails : ReferenceDetailsClass│
│  CrossReferences : CrossReferencesClass  │
│  Type : ArtefactTypeClass = manuscript   │
│  Scribe : DummyClass = NULL              │
│  Purpose : DummyClass = NULL             │
│  CopyNo : Integer = 0                    │
│  Pages[] : Pagelass = NULL               │
├─────────────────────────────────────────┤
│  ManuscriptClass()                       │
│  opname()                                │
└─────────────────────────────────────────┘
```

Class Diagrams show the details of classes. A class describes the structure of something such as a manuscript; it is a 'blue print', a plan or genome. The class diagram for a manuscript contains classes for other objects which contain other classes. The arrows on the diagram indicate that one class is dependent on another class.

As far as a methodology to study Turkic texts is concerned the text is the most basic class that is it is not dependent on other classes. In the diagram higher level classes define the alphabet that will enable reading the text. The diagram argues that a text can exist even though nobody can read it. On the other hand in

order to understand a text knowledge of the alphabet and language is important. The methodology outlined in this paper acknowledges and facilitates opposition. In the diagram I have tried to make allowances for text such as *the Orkhon Monuments* which are carved in stone. In this case the size, weight and orientation are important.

UML has many other diagrams. Activity, State and Sequence Diagrams represent the timing aspects of the model. Object and Instance Diagrams explain the data. There are diagrams that cover collaboration and implementation. Each diagram has its own visual grammar and semantics. For the last few years, UML has been developing extensively and now it is felt that most scenarios have been covered and a stable version of the language has been reached.

As far as Turkic texts are concerned, it is a good environment in which to model knowledge. Implementation of UML has become very sophisticated (and complex) but I feel that it can act as a bridge between conventional methods of studying old Turkic texts and the very dynamic range of techniques and facilities that Information Technology can offer. I hope that a methodology in which to study old Turkic texts should not only implement what is currently available but should embody the ability to absorb change.

Knowledge Models

Artificial Intelligence and Knowledge Engineering (Juristo, Natalia, Ana M. Moreno: Introductory paper: Reflections on Conceptual Modelling) have been using similar techniques to UML to represent knowledge. The methodology to study Turkic texts, I hope, will reflect knowledge of the subject. A lot of facts and figures will remain just that, without the organisation that will turn it into knowledge.

Knowledge is an amorphous term, it does not mean intelligence and it is more than just facts and figures. Information gets close but knowledge suggests something more. (Gruber, Thomas R: Toward Principles for the Design of Ontologies Used for Knowledge Sharing) What is important is the organisation or the relationship between pieces of information. Organisation not only helps find the information you are looking for but holds knowledge, similar to the human brain it is not just the synapses that are important it is what happens between them.

Concept Tree
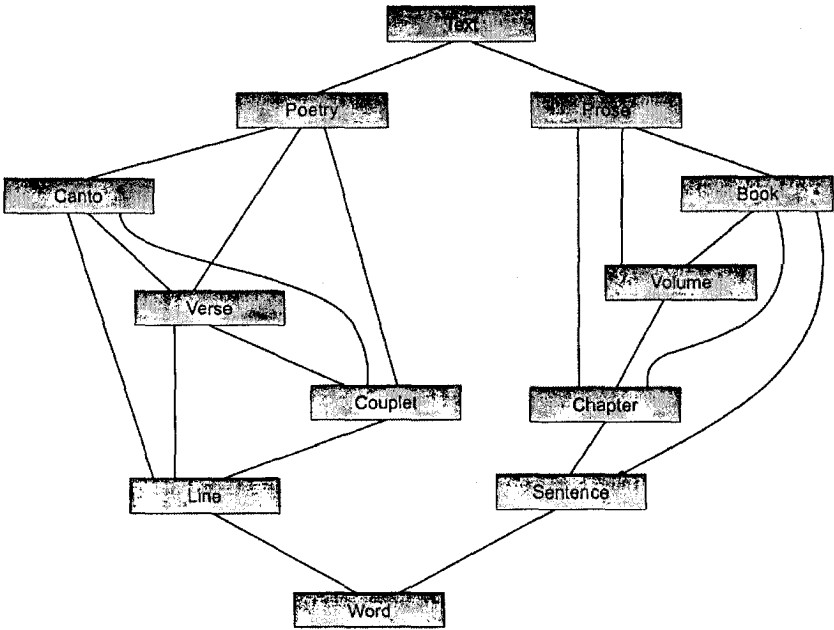
Concept Trees

A concept tree represents the knowledge that is embedded in the relationships between different concepts. A piece of text breaks down to the concept of poetry and the concept of prose. Poetry can be further broken down into canto, verse, and couplet and so on. Poetry and prose each have their own concept breakdowns but at certain stage there is a break in their separate linearity and the two treads share commonality of words. Individual words are common to both poetry and prose. They share some common attributes which can be identified in a grammatical index.
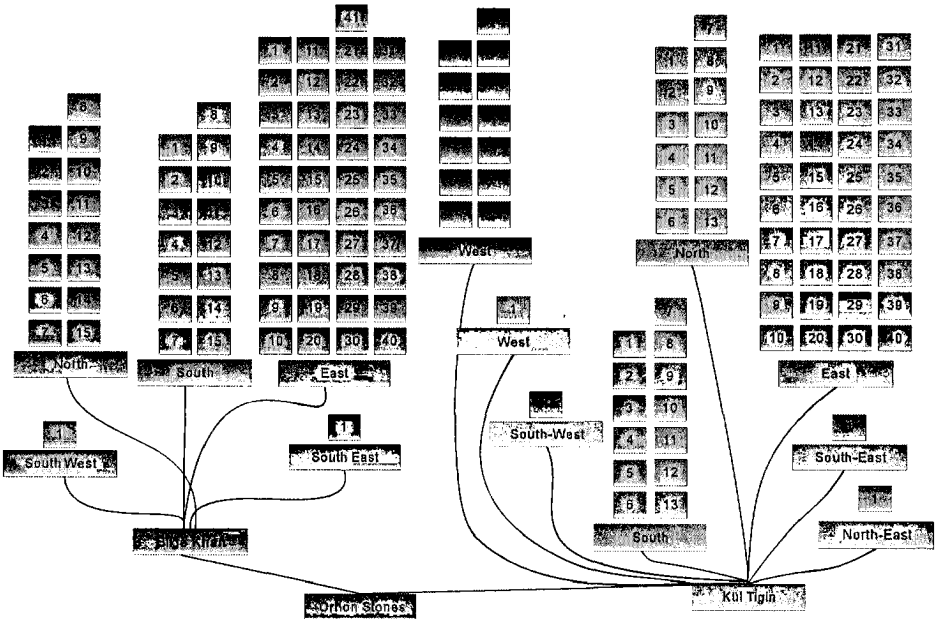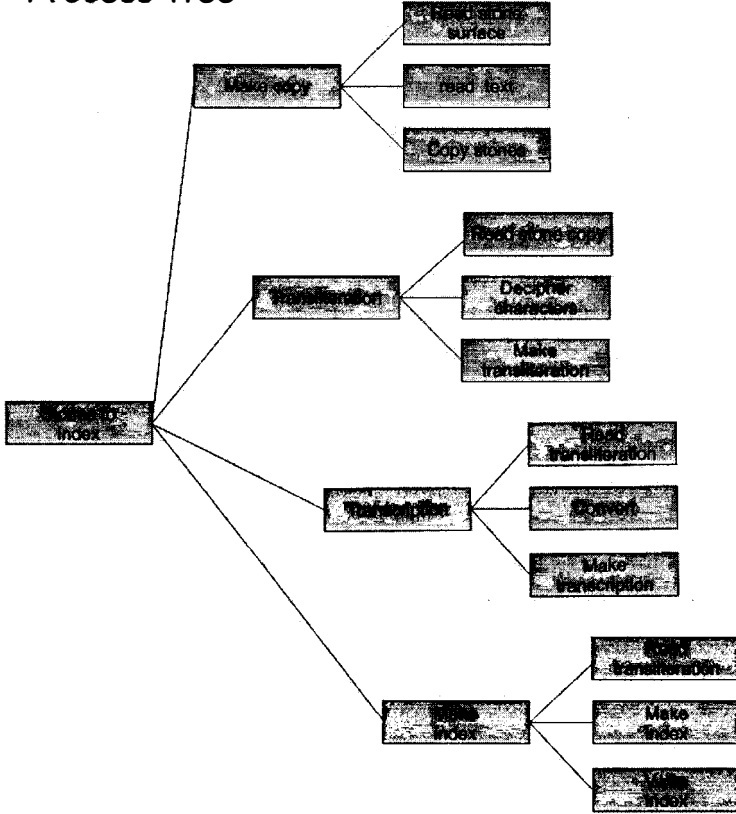
Composition Trees

A composition tree or ladder organises information in terms of attributes of objects. This diagram shows a composition tree for *the Orkhon Monuments*. The diagram shows that there are two stones one for *Köl Tigin* and the other stone for *Bilge Khan*. The stone for *Köl Tigin* has seven surfaces of text of which the east face has the largest number of lines. The west surface has Chinese characters.

More information could be put on the diagram, or more importantly different diagrams could be used to show the relationships different elements. The composition diagram characterises the stones but says nothing about the text. The text itself will have attributes of language, grammar and semantics. The methodology to study Turkic texts should accommodate how different attributes relate to different linguistic concepts on the text. UML provides the concept of stereotypes where an object such as a poem can be examined as a piece of grammar, a theological eulogy or as a simple love poem.

Process Tree

Read the
surface

Read text

Copy stone

Read stone copy

Decipher
characters

Make
transliteration

Read
transliteration

Convert

Make
transcription

Read
transliteration

Make
index

View
index

Process Trees

These diagrams show the sequence of events and the decisions that have to be made to archive a task. Therefore to in order to generate a grammatical index of what is written on the *Orkhon Monuments* it has to be read. This task is broken down into trying to extract what was inscribed on the stone. After over a thousand years this is not easy as the stones are not complete and the surface details have worn away. More detailed process trees could represent the knowledge of what can be done to extract the individual characters on the stones such as ultrasound and magnetic resonance techniques. This process tree shows four main stages of

producing a grammatical index; first reading the text, second transliteration, then transcription and finally the index generation itself.

Dynamic Knowledge Modelling

The concepts, structures and technology associated with a methodology should be dynamic and organic (Motta, E., D. Fensel, M. Gaspari, and R. Benjamins: Specifications of Knowledge Components for Reuse) in the sense that it should be able to accommodate new ideas, should be able to hold two or more opposing viewpoints and it should be able to be self analytic and make value decisions based on collective experience. Not all these have been realised but I will outline how they can be achieved. I will deal with the first requirement last as it is the most difficult.

♦ Hold two or more opposing viewpoints

If I can build dynamic concept maps it should be possible to build opposing points of view. They are just points in the tree that branch in differing directions. Theories can develop, die or even sub divide and grow organically. Together with the next requirement dynamic concept modelling could enable us to forecast how ideas could develop in the future.

♦ Make value decisions on collective experience

It should be relatively easy to monitor how the object structures are used. I suggest that the most popular interactions lead to the most successful ideas. Interactions that lead nowhere are not repeated, or at least, not repeated by the same user and will eventually die. Using how the structure is used could influence how the structure grows and develops.

♦ Accommodate new ideas

Everything is dependent on being able to accommodate new ideas and grow under the control of the structure's users. This is the most important and the most difficult of my requirements. Users have to have the same facilities as system designers as well as having an invisible interface. The methodology should not turn Turkologs into systems analysts.

In order to accommodate new ideas the methodology object structure has to be able to introduce new attributes and new actions, to create new objects and make new relationships. All this has to be done in the least intrusive way possible.

Dynamic structures have been built by other people working in other fields and it is still an open question if these techniques can be used in the study of old Turkic texts.

It is not certain what the final interface will look like. Good interfaces are difficult to design and a bad interface may bring the project to an abrupt end. Not only has the interface got to look and feel good it has to accommodate dynamic data structures.

The Wikipedia web site provides encyclopaedic type information, it is easy to use, provides references, citations, and bibliographies. Readers are able to add their knowledge on the subject and if they have references so much the better. It is a good example of a dynamic knowledge resource. The interface to the methodology should not be solely computer based and should be accessible through different media and contexts.

Conclusion

I hope that this short paper is relevant to what people are actually doing. My research started with finding a method to produce grammatical indexes more quickly and accurately than putting everything down on small pieces of paper. This work is still continuing and developing links to new disciplines. However, as I am putting the final changes to my first objective a new one is on the horizon. What used to be work that focused inward on the written word and its linguistic features is now looking further a field. I have made a start to formalise my thoughts by using UML which in itself has focused on the fundamentals of studying the documents of Turkic culture.

Language is fundamental to communication. To study language, its roots and development is to study the roots and development of communication. Without communication nothing can be passed on to the next generation. Studying old manuscripts should tell us what was felt to be important and should be communicated through history. Studying history should tell us if this was successful.

References:

GRUBER, Thomas R. (1993). "Toward Principles for the Design of Ontologies Used for Knowledge Sharing", *Stanford Knowledge Systems Laboratory.* August 23.

JURISTO, Natalia and Ana M. Moreno. (2000). "Introductory paper: Reflections on Conceptual Modelling", *Data & Knowledge Engineering.* 33, pp. 103-117.

MINGANA, Alphonse, D.D. (1914-15). "An Important Old Turkic Manuscript in the John Rylands Library", *John Rylands Library Bulletin.* Manchester, volume 2, pp. 129-138.

MOTTA, E., D. Fensel, M. Gaspari, and R. Benjamins. (1999). "Specifications of Knowledge Components for Reuse", *Eleventh International*

Conference on Software Engineering and Knowledge Engineering (SEKE '99). January.

RUMBAUGH, James, Ivar Jacobson, Grady Booch. (1998). *The Unified Modeling Language reference manual.* Longman.

TEKCAN, Münevver 2006: "Grammatical Index Generation For Turkish Texts", *Journal of Turkish Studies* (Türklük Arastirmalari Dergisi). Festscrift In Honor of Orhan Okay III / Harvard USA, vol. 30/III, pp. 181-208.

_____. (2008). "An Object Based Methodology of Studying Turkic Texts", *International Journal of Turcologia,* vol. III, SPRING pp. 75-92.

Tekin, Talat. (1968). *A Grammar of Orkhon Turkic.* Indiana University Publications, Bloomington.

_____. (1988). *Orhon Yazıtları.* Ankara.

The British Library – The world's knowledge (http://www.bl.uk/)

VÖLKEL, Max 2007: "From Documents to Knowledge Models", *Prof. Dr.-Ing. Norbert Gronau, Proc. of the 4th Conference on Professional Knowledge Management.* GITO mbh, Berlin, volume 2, March pp. 209-216.