

# Design and Implementation of a Model-driven XML-based Integrated System Architecture for Assisting Analysis, Understanding, and Retention of Religious Texts: The Case of The Quran

Mohamadou Nassourou  
Department of Computer Philology & Modern German Literature  
University of Würzburg Am Hubland D - 97074 Würzburg  
mohamadou.nassourou@uni-wuerzburg.de

## Abstract

Learning a book in general involves reading it, underlining important words, adding comments, summarizing some passages, and marking up some text or concepts. Once deeper understanding is achieved, one would like to organize and manage her/his knowledge in such a way that, it could be easily remembered and efficiently transmitted to others.

This paper discusses about modeling religious texts using semantic XML markup based on frame-based knowledge representation, with the purpose of assisting understanding, retention, and sharing of knowledge they contain.

In this study, books organized in terms of chapters made up of verses are considered as the source of knowledge to model. Some metadata representing the multiple perspectives of knowledge modeling are assigned to each chapter and verse. Chapters and verses with their metadata form a meta-model, which is represented using frames, and published on a web mashup.

An XML-based annotation and visualization system equipped with user interfaces for creating static and dynamic metadata, annotating chapters' contents according to user selected semantics, and templates for publishing generated knowledge on the Internet, has been developed.

The system has been applied to the Quran, and the result obtained shows that multiple perspectives of information modeling can be successfully applied to religious texts, in order to support analysis, understanding, and retention of the texts.

**Keywords:** Knowledge representation, Meta-model, Frames, XML model, Knowledge Management, Content Management, Quran

## 1. Introduction

Knowledge Management is defined as a collection of processes for creating, retaining, disseminating, and utilizing knowledge. Knowledge is gained through learning process. There are different types of learning theories among them behaviorism, cognitivism, and constructivism. This paper discusses mainly constructivism because learners interactively build new concepts from information contained in books.

The purpose of learning is to gain knowledge. But there is general knowledge and specific knowledge. In this paper knowledge within a defined domain is discussed. Books under consideration are those whose logical structures can be described in terms of chapters. And each chapter is made of a header and a body section. The header comprises a title, an edition place,

number of verses, and some optional fields such as author or writer. The body is mainly a collection of verses. Reading and understanding this kind of books is not a trivial task, because they usually encompass a lot of hidden knowledge, unusual

expressions, and a peculiar style of narration. As consequence, a well thought strategy is required in order to effectively understand them.

It is commonly known in chemistry or physics that students understand better when they model processes and reactions that they have learned [7]. Breaking down a text into smallest meaningful parts, enumerating all possible required details about each part, and selectively visualizing each part, are reasonably effective ways of understanding and retaining the text.

This paper presents an integrated system architecture that covers all steps susceptible for supporting learning religious books, and applies that to the Quran. The system comprises Dynamic and Static Annotators, an XML Editor, a Knowledge Tester, an Expert System, and a Publisher to a Web Server.

The purpose of having two separate annotators is linked to the fact that religious books could be assigned dynamic as well as static metadata.

Dynamic metadata are those that act directly on the text such as part-of-speech and named entities tagging, and general comments about any word. The metadata are dynamic in the sense that they are attached directly to the texts, and are therefore subject to alteration at any time, as soon as new concepts or techniques are discovered.

Dynamic annotations such as part-of-speech, named entities tagging help understanding the texts themselves, but they do not tell learners about the purpose, application, and reasons of existence of the texts. It is the static annotations as described in this study that provide this information.

Static metadata are external to the text. For instance place, date, causes, and methods of coming into existence of a verse do not usually changed as far as religious texts such as the Quran are concerned. Static metadata are viewed as a representation of the multiple perspectives of knowledge modeling, which by extension correspond to the basic questions of genetic edition. Genetic edition is concerned with the coming into existing of texts. The annotator helps users to insert and revise questions about each verse. Usually genetic edition questions are of the following forms:

Where was the verse written? When was the verse written? How was the verse written? What are the ways of reading the verse? Who created and wrote the verse? Who does the verse refer to?

Additionally for the purpose of this research, following questions are included: Why was the verse created or written? Which message does the verse convey? What are other similar verses?

## 2. Problem Statement

Understanding and memorizing religious books is a challenging task. Can semantic XML markup based on multiple perspectives of information modeling, and visualization techniques alleviate this challenge?

### 3. Proposed Solution

The hypothesis is that, modeling and visualizing structural and terminological dimensions of religious books with the help of computer-aided semantic annotations will improve understanding and retention of the books' contents.

Through intuitive user interfaces, well structured presentations, and detailed explanation with analogies, knowledge modeling and visualization techniques could considerably assist understanding and memorization of religious texts.

### 4. Identifying The Multiple Perspectives of Information Modeling in Religious Texts

The multi-perspective approach proposed in this study is based on the Zachman framework for Information Systems Architecture. The Zachman framework [1] proposes that six perspectives characterized by the phrases what, who, when, where, how and why are necessary for completely representing and understanding an object.

Knowing that religious texts consist principally of collection of verses, the six perspectives with respect to each verse could be identified as follows:

- a) The What perspective  
It refers to the verse itself, and how to read or recite it.
- b) The Who perspective  
It concerns the author, the editor, as well as those who the verse refers to.
- c) The When perspective  
It records the time when the verse was emitted and recorded.
- d) The Where perspective  
The place where the verse was created and recorded is meant here.
- e) The How perspective  
It treats questions such as how and from what the verse was created and recorded.
- f) The Why perspective  
It deals with the causes or occasions that lead to the creation of the verse.

Because the aim of this study is to find out how understanding and retention of religious texts could be achieved through knowledge management, I found it necessary to add some perspectives such as similarity to others (within same religious book, other religious books), and message conveyed (i.e which message does the text convey). Moreover due to the fact that, this study will end up producing a knowledge-based system, the size, location, and topic of verses are included as well.

### 5. Understanding Religious Books

In general it could be said that one understands something if he/she sees how it can be used to achieve some purpose or goal. Understanding is a process whereby sound knowledge is gained. Sound knowledge must be manageable. But knowledge is the result of information modeling. Therefore understanding religious books requires first modeling the information they contain in order to acquire clear knowledge of their contents, and properly manage that knowledge. In other words the first step towards understanding religious books is to create a model of their contents.

### 5.1 Model Definition

The purpose of modeling knowledge contained in religious books is to effectively manage that knowledge. Text Modeling is the process of structuring a text, as well as adding annotations for explaining that text. Books under consideration are those whose logical structures can be described in terms of chapters. And each chapter is made up of a header and a body. The header comprises a list of attributes that could be termed as metadata of the chapters and their contents. The body is mainly a collection of numbered blocks of texts that could be subdivided into words.

The structure of the books could be easily apprehended through XML representation. Usually for textual documents markup TEI [15] seems to be a good candidate. However knowing that one of the aims of this study is to come up with an XML schema specifically for religious books that are complying with the Quran's structure, it has been judged necessary to new define tags for each element of the books. Moreover TEI allow customization whereby an application could add its own tags in order to extend existing TEI tag sets.

In short, the structure of the considered books could be described with the following XML fragment:

```
<book>
  <chapter n="1">
    <header>
      <attr1></attr1>
      <attr2></attr2>
      ...
    </header>
    <body>
      <text_block1 n="1"><word1></word1>...</text_block1>
      ...
    </body>
  </chapter>
  ...
</book>
```

This XML fragment describes the logical structure of religious books under consideration, which would be studied and annotated. This study tries to define some metadata for "book", "chapter", "text\_block", and "word" elements in the above XML fragment.

Annotating is the process of defining some attributes termed as metadata for describing the book itself, its chapters and chapters' contents. The purpose of the metadata is to provide detailed information about the book and its content for facilitating understanding, retention, and information retrieval. There are static and dynamic metadata, which will be assigned to the book using static annotator and dynamic annotator components respectively. Metadata assignment refers to the creation of enclosing tags as well as data they enclose.

Static annotations are those that the encoding tags are created once for each sentence of the document. The encoding tags are made before reading the document. For instance for each sentence one would like to know where, when, how, and why the sentence was included in the document.

Dynamic annotations are annotations made for any text in the book. The encoding tags for this type of annotations are the text themselves or any other word chosen by the user.

Referring to religious texts particularly, dynamic annotations such as simple comments, part-of-speech and named entities tagging help understanding the texts themselves, but they do not tell learners about the purpose and reasons of existence of the texts. It is the static annotation as described in this study that could provide this information.

Once the book is annotated, it could be simply viewed as another model that might be called a meta-model of the book. It is this meta-model that the current study tries to devise, implement, and test.

## 5.2 Metadata Definition

First of all it is worthwhile mentioning that, dynamic metadata are those that act directly on the text such as part-of-speech and named entities tagging, and general comments about any word. The metadata are dynamic in the sense that they are attached directly to the texts, and are therefore subject to alteration at any time, as soon as new concepts or techniques are discovered. Dynamic annotations such as part-of-speech, named entities tagging help understanding the texts themselves, but they do not tell learners about the purpose, applications, and reasons of existence of the texts. It is the static annotations as described in this study that provide this information.

Static metadata are external to the text, and normally expected to suggest the purpose, application, and reason of existence of the text. For instance place, date, causes, and methods of coming into existence of a verse do not usually changed as far as religious texts such as the Quran are concerned.

Modeling a religious book requires viewing the book's content in several perspectives. Since the content is mainly a collection of numbered verses, it is therefore necessary to analyze each verse in all possible perspectives. The multi-perspective approach that I have considered in this study is principally based on the Zachman framework for Information Systems Architecture [1]. The Zachman framework defines six perspectives on information characterized by the phrases who, what, how, when, where and why.

A book could be understood through its internal and external evidences. Internal evidence refers to the content and meaning of the book, and external evidence points to data surrounding the book such as place and date of publication, author, and so on. To some extent one could argue that, these evidences correspond basically to the Zachman's proposed perspectives on information systems architecture.

- **Selection of static metadata tag names**

Based on these evidences following terms have been selected as static metadata for each verse:

- 1) place of creation/revelation
- 2) date of creation/revelation
- 3) method of creation/revelation
- 4) occasions/causes of creation/revelation
- 5) ways of reading
- 6) explanation
- 7) similarity
- 8) writer/editor
- 9) size
- 10) location
- 11) topic

The metadata of the book itself could consist of:

1. chapters' names with number of verses
2. chapters' order of creation/edition
3. total number of verses
4. place of compilation
5. date of compilation
6. method of compilation
7. causes of compilation

- **Design of dynamic metadata**

It must be noticed that, books that comply with this way of structuring their contents could also be viewed as an aggregation of chapters consisting of collection of words. Using this view it is possible to annotate each word or group of words independently of the static metadata. Therefore the metadata of the annotated words are termed as dynamic metadata, since they are attached directly to the texts, and are consequently subject to alteration at any time as soon as new concepts or techniques are discovered.

Having defined the model and its metadata, it is possible to generate a meta-model that could be based on synthesizing the model and its metadata. The meta-model could be considered as a model of domain of interest for religious books.

## 5.3 Meta-Model

In this study a meta-model is considered as a model of domain of interest. In other words the meta-model is just a model of religious books consisting of the book's structure and its metadata.

Figure 1 is diagram of the meta-model. This diagram is a kind of frame-based knowledge representation system, whereby book, chapter, text block, text, and word elements could be viewed as frames with their metadata as slots and facets. Each element can inherit metadata of its parent if its metadata are unavailable.

### Description of the Meta-Model shown in figure 1

The meta-model consists of a book comprising some chapters made up of block of texts such as verses. Chapters and their contents are described with some metadata.

It is important to mention that if static metadata of a child do not exist, then its parent metadata are automatically assigned to it. For example if metadata of a verse are not available, then metadata of its chapter are assigned to it.

The metadata convey clear semantic about what they describe. For instance

- a) `book_attr1` element stands for attribute 1 of the book
- b) `chap_attr1` element stands for attribute 1 of the chapter
- c) `text_block_attr1` element stands for attribute 1 of the text block

`book_attr1`, `chap_attr1`, and `text_block_attr1` elements represent enclosing tags of the metadata.

Considering chapters as collections of words gives the possibility to annotate each word by associating comments and markups, whereby each word has got three attributes: chapter number "`chap_no`", word position "`offset`", and word importance "`weight`" in the book. The "`comment`" and "`markup`" elements stand for tags' names of comment and markup data respectively. The markup element represents all possible kind of markups such as part-of-speech, named entities tagging ... etc.

Representing the meta-model using XML requires dissociating the static annotations from the dynamic ones. The reason why I separate them is a practical one. Static metadata are written together with the text blocks of the book in the same file, while dynamic metadata are not. For the dynamic metadata stand-off markup is used, whereby the metadata are stored in a separate file but linked to the text blocs respectively. For instance static metadata of the Quranic

verses are written together with the verses in the same file, while dynamic metadata are written in a different file.

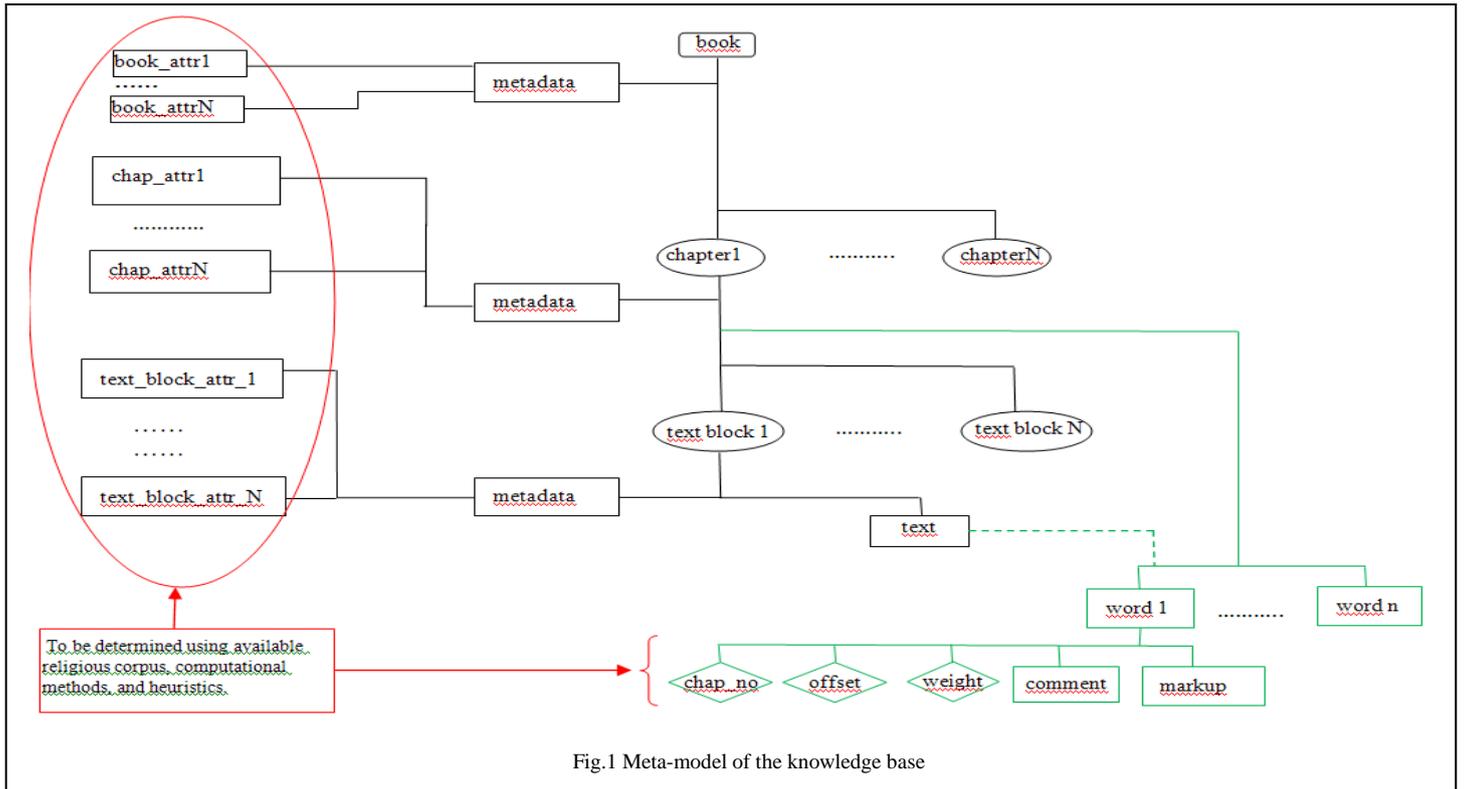


Fig.1 Meta-model of the knowledge base

```

<book>
  <metadata>
    <book_attr1> </book_attr1>
    ...
  </metadata>
  <chapter1>
    <metadata>
      <chap_attr1></chap_attr1>
      ...
    </metadata>
    <text_block1>
      <metadata>
        <text_block_attr_1></text_block_attr_1>
        ...
      </metadata>
      <text>
        <word1 chap="chapter number" pos="offset" weight="x">
          <comment>comments</comment>
          <markup>word1 </markup>
          ...
        </word1>
        ...
      </text>
    </text_block1>
    ...
  </chapter1>
  ...
</book>

```

Figure 2. XML representation of static part of the meta-model

```

<book>
  <metadata>
    <book_attr1></book_attr1>
    ...
  </metadata>
  <chapter1>
    <metadata>
      <chap_attr1></chap_attr1>
      ...
    </metadata>
    <word1 chap_no="chapter number" offset="offset" weight="x">
      <comment>comments</comment>
      <markup>word1 </markup>
      ...
    </word1>
    ...
  </chapter1>
  ...
</book>

```

Figure 3. XML representation of dynamic part of the meta-model

XML representations of the static and dynamic meta-models are shown in figure 2 and 3 respectively.

The meta-model serves as a knowledge base structure which needs to be created and collectively used.

If data for the metadata are not available, text mining methods should be applied to derive those data from the content of the book itself.

The “<text>” element of a <text\_block> element could be split into words. However this part is optional since a chapter could be directly described as a collation of words.

## 9.4 Frame-based Representation of the Meta-Model

A frame is a data structure that looks like modern classes, without methods in object oriented programming languages. It consists of multiple slots used to define various attributes of an object. Some slots hold static data, while others variable ones, as well as procedures. Multiple frames are linked together in a hierarchy so that children frames can inherit from parent ones. Fig 4 shows

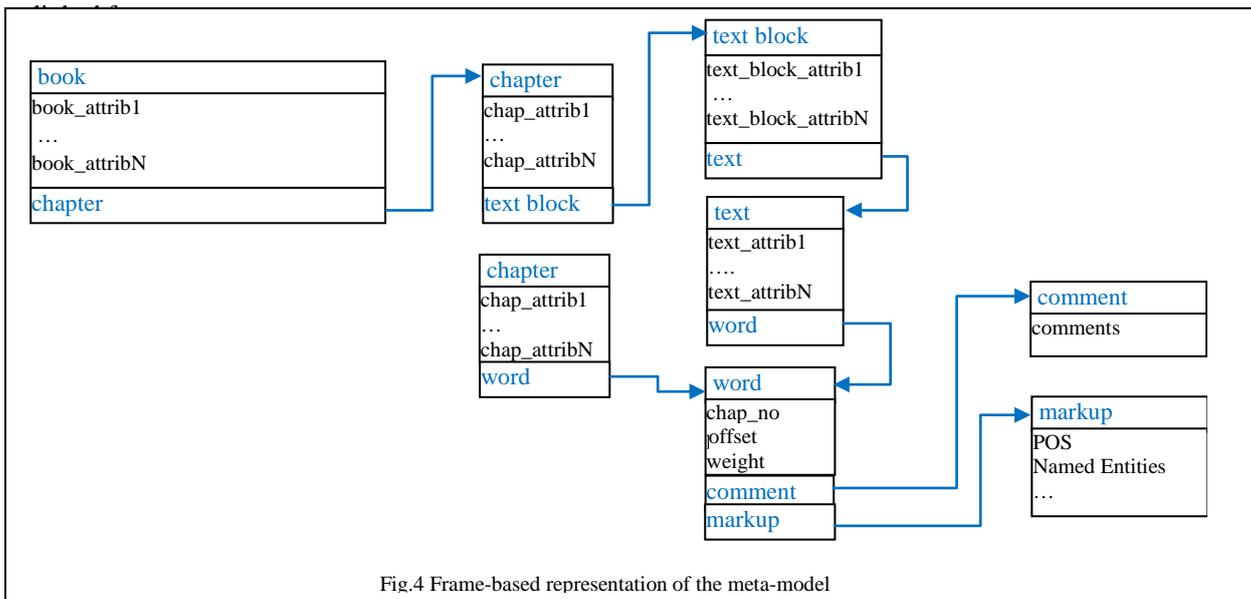


Fig.4 Frame-based representation of the meta-model

The elements of the frames represent the enclosing tags of the metadata whose data need to be collected or computed.

The “book” frame has got some static data (facets) and a slot called “chapter” that stands for chapters of the book.

The “chapter” slot is in turn a frame on its own consisting of some facets that are inherited and could be overridden, and a slot called “text block”.

The “text block” slot is also in turn a frame on its own consisting of some facets that are inherited and could be overridden, and a slot called “text”.

The “text” slot is also in turn a frame on its own consisting of some facets that are inherited and could be overridden, and a slot called “word”.

The “word” slot is also in turn a frame on its own consisting of “chap\_no”, “offset”, and “weight” facets, and “comment” and “markup” slots.

A chapter could be viewed as either a collection of numbered verses or as a collection of words. So the element “chapter” could be represented with two different frames as shown in the diagram above. The first representation using verses is the static part of the meta-model, while the second one based on words is the dynamic representation one. For this reason two XML models of the frames based on separating the two views of the chapter frame are created. Moreover a chapter viewed as collection of words was marked up using stand-off markup, this is referred as the dynamic representation of the meta-model, and the other view is considered as the static representation of the meta-model.

Now that the structure of the knowledge base is defined, I would like to introduce the manager of the knowledge base, which is actually a semi-automatic system for creating, storing, and disseminating knowledge from these types of books.

## 6. Knowledge Management System

A Knowledge Management System (KM System) is defined as any system that offers possibilities to manage knowledge in

organizations, in order to support creation, capture, storage and dissemination of information. In this study, the system that has to be developed must be able to capture every information for each verse of the religious book under consideration, and make this information accessible to any interested person.

The system consists basically of a Mashup website and a Content Management System. The Mashup website integrates a Publisher, Tester, and an Expert System. And the Content Management System is based on an XML Generator, XML Editor, Static and Dynamic Annotators programs.

### 6.1 Architecture of the System

The Architecture is made up of layered architectural diagrams with each layer depending on the layer below it. It is a component-based software system consisting of six major modules interacting with each other.

Figure 5 shows the diagrammatical view of the architecture.

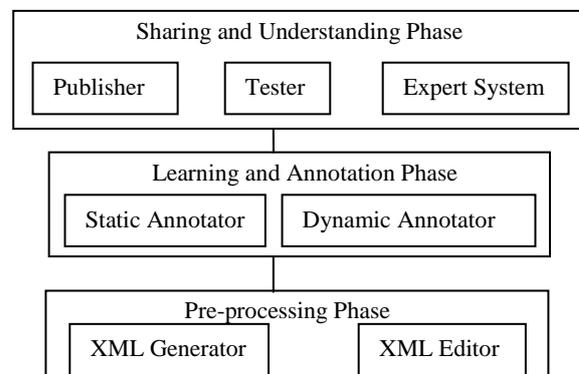


Fig.5. Architecture of the system

#### Components Description:

First of all, it might be important to mention that, including a screenshot for each component would make the paper considerably big, therefore no screenshot is added.

### 1. XML Generator

It is a program that could be used to automatically convert a single text file or a collection of text documents into XML.

### 2. Smart XML Editor

The editor is equipped with an interface for creating an XML-based ontology, which could be used to automatically mark up some well known features such as persons, animals, planets...etc. This editor is capable of eliminating overlapping using fragmentation technique described in [12]. A content assistant provides users with the possibility to choose tag to insert, and in case non well-formedness arises, it rectifies the illness by fragmenting the overlapping elements, and inserts corresponding missing tags. If none of the options is selected, or a text is selected, the content assistant inserts both closing and opening tags. Currently the editor is still under development to include the possibility of fragmenting overlapping elements without the need of a content assistant.

### 3. Static Annotator

It is a program for entering metadata names and their data. The annotator provides interfaces for properly and efficiently creating and managing metadata of chapters and verses. This program works like a relational database system such as MySQL. The creation of metadata names is equivalent to the creation of field or column names, and the entering and maintenance of the entered data similar to insert, update...etc functions. The advantage of this annotator is that, it generates automatically XML data that could be offered as webservice to other applications.

### 4. Dynamic Annotator

It is a program that displays the original text and its translation side by side. It offers the possibility to annotate a text in the translation version by selecting a label or concept for the annotated text. An annotated text is automatically highlighted, hyperlinked to its annotation, and its corresponding text in the original text is also highlighted whenever possible, thereby assisting reading and analysis. A user can at any time click on the link to display or update the annotations. Moreover searching the annotations by using the selected labels or concepts is also possible.

Due to religious reasons, the system does not allow annotation of the original text such as the Quran in Arabic. However this might change in the future.

### 5. Expert System

It is a type of instructional expert system that users can consult for learning and testing their knowledge. It helps gaining knowledge, and checking through questions and answers interaction, what has been studied. The system could be viewed as a kind of dialog based learning or query program. Following is a simple example based on the Quran that shows how the system works:

A user interface allows a client to enter a verse or some words, and get the metadata of all the matching verses. However this is not a straight forward answer. The problem is that, almost half of the verses are similar, where 98 verses are repeated 181 times, with some in different chapters. Which metadata of which verse to select? Here the Expert System would ask some questions such as topic, location, and size of the verse, and then infer the right answer from those questions. For instance the Expert System could request the chapter number or name where the verse is located. If it is not

known then it asks for a verse that might be in the same chapter. If again the client does not know, then it asks an approximate location of the chapter that contains the verse, such as: is the chapter at the beginning, end...?

The Expert system is developed using PHP/Javascript/HTML/CSS and MySql Database. It is currently a simple one, and will be updated regularly as soon the whole project is completed and published.

### 6. Tester

Different researchers use different methods for measuring understanding. However majority of them have used: true-false questions, multiple choice questions, paraphrasing documents, and answer questions.

The Tester in this research consists of multiple choice questions, and a textual documents collation system. The collation system can compare two texts in order to identify their similarities and differences.

A collation system could ease memorization of religious text by highlighting similarities and differences, for more information on textual documents collation see [13, 14].

### 7. Publisher on Web Servers

This is a website for sharing and communicating with others. It consists of several web pages linked through the verses and metadata labels.

The publisher offers the possibility to search in different languages, and interactively visualize the search results as described in [4]. The original book might have been translated in several languages; hence annotations and searching could be made in any of the translation languages.

Having briefly described the system, I would like to explain how the components are used to create and manage a desired knowledge base.

## 6.2 Generating The Knowledge Base

In this section I am going to show how the system developed in this study could be used to implement knowledge management activities.

a. **Knowledge Creation** involves discovering, capturing, and registering the knowledge into the system. The book from which the knowledge has to be generated could be locally stored or retrievable from the Internet.

The following functional components implement the knowledge creation activities. Interactively create metadata of verses:

1. Manual and automatic annotations (meta-data assignment) of verses are done by the XML Generator, the XML Editor, and the Static and Dynamic Annotator components.

b. **Knowledge Dissemination** or knowledge transfer refers to communicating the knowledge in a well-defined format so that others can effectively use it. Following is the component which takes care of this process.

1. The Publisher component which is a Mashup website.

c. **Knowledge Utilization** is concerned with the application of acquired knowledge wherever needed. It is realized with the following systems:

1. Test unit
2. Expert system

3. The Publisher component which is a Mashup website.
- d. **Knowledge Preservation** is achieved with the help of XML. The created knowledge base and the source documents are stored in XML format.

## 7. Working Principle of the System

The ontological diagram in figure 6 describes the overall working principle of the system.

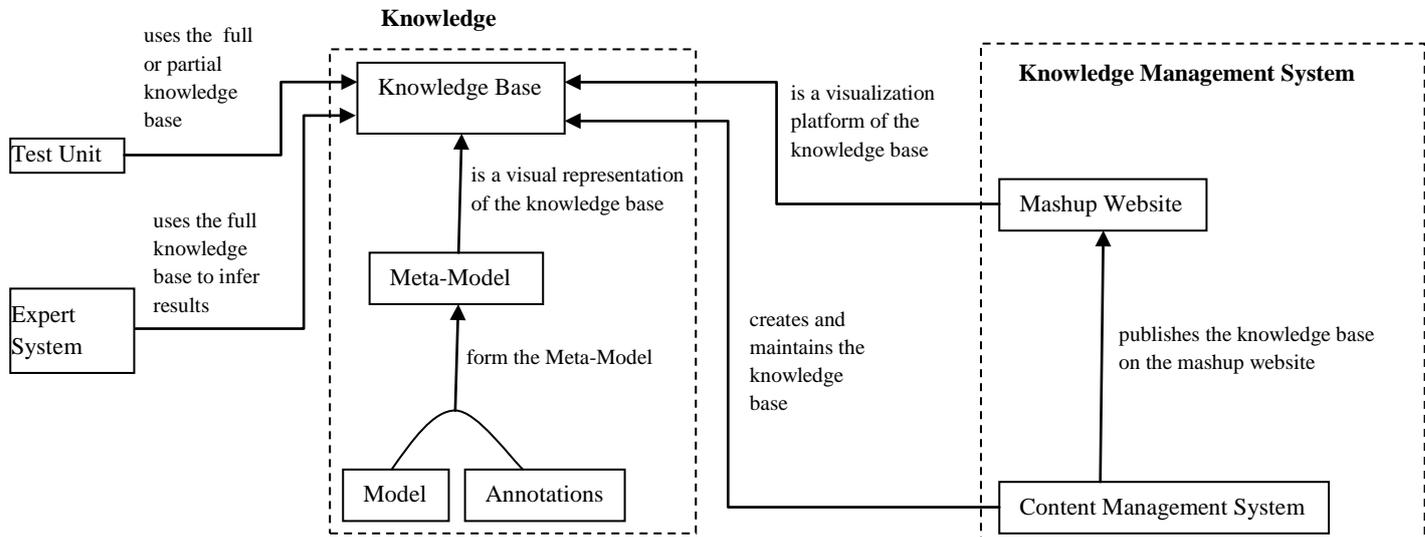


Fig. 6 Ontology of the system

### Description of the system

The diagram in fig. 5 shows the major components of the overall system. The knowledge block consists of the knowledge base represented by the meta-model that comprises the model with its annotations. The knowledge base is created, maintained, and visualized by the Content Management System and the Mashup website. The Content Management System is also responsible for publishing the knowledge base on the Mashup website.

The Test Unit and the Expert System are full or partial consumers of the knowledge base through request response interactions, as well as dialog base communication.

## 8. Implementation

The system was implemented in Java and PHP. The Content Management System was first developed in Java as RCP eclipse plugin, and will be converted into web based one using PHP, Javascript, and HTML/CSS.

## 9. Applications of the Meta-Model

The meta-model could be applied to any book or document whose schema or organization is compliant with the above described model. Religious books particularly would definitely fit into this category, and are therefore excellent examples for implementing this meta-model. Following is a use case based on the Quran.

### 9.1 Case of the Quran

The holy Quran contains 6236 numbered verses divided into 114 chapters, whereby opening verse is considered for the opening chapter only. Some chapters were revealed in Mecca while others in Medina (cities in Saudi Arabia). According to Muslims each verse was revealed to Prophet Mohammad by

God through Angel Gabriel at a specific place, in a given time, in a particular way, and for a well needed purpose. Understanding the Quran is not an easy task because each verse is a chapter on its own, and each chapter is a Quran (i.e a book) on its own. This implies that understanding the Quran requires understanding each individual verse. Knowledge contained in each verse can be acquired by analyzing and comprehending detailed information about that verse.

### 9.2 Modeling the Quran's Content

The Quran has got a structure that could be described in terms of chapters. And each chapter is made up of a header and a body section. The header comprises a title, a revelation place, and number of verses. The body is mainly a collection of numbered verses. For the purpose of memorization and recitation, the Quranic verses have also been divided into groups of 7 (each group called manzil), 30 (each group called juz), 60 (each group called hizb). These divisions are rather not inherent to the revelation.

An XML representation of each chapter is shown below:

```
<quran>
  <chapter n="1">
    <header>
      <chaptername>chapter name</chaptername>
      <placeofrevelation>place of revelation</placeofrevelation >
      <numberofverses>number of verses</numberofverses >
    </header>
    <body>
      <verse n="1"> <word1>word 1</word1>...<wordn>word n</wordn></verse>
      ...
    </body>
  </chapter>
  ...
</quran>
```

To define some metadata for the Quran's chapters and verses, there is need to have basic knowledge about its origin. According to Muslims each verse was revealed to Prophet Mohammad at a specific place, in a given time,

in a particular way, and for a well needed purpose. These characteristics of each chapter and verse are actually a hint for defining a multi-perspective representation of the chapters and verses. The multiple perspectives could be mapped to some static metadata with the purpose of providing detailed information about each one of them. Without going into details, the multiple perspectives could be represented with the following terms.

1. place of revelation
2. date of revelation
3. method of revelation
4. occasions/causes of revelation
5. ways of reading
6. explanation
7. writer/editor
8. similarity to others
9. size
10. location
11. topic

The following terms have been chosen for the metadata of the Quran itself:

1. chapters' names with number of verses
2. chapters' order of revelation
3. total number of verses

### 9.4 Frame-based Representation of the Quran's Meta-Model

Fig 7 shows a diagram for the frame-based representation of the Quran's meta-model.

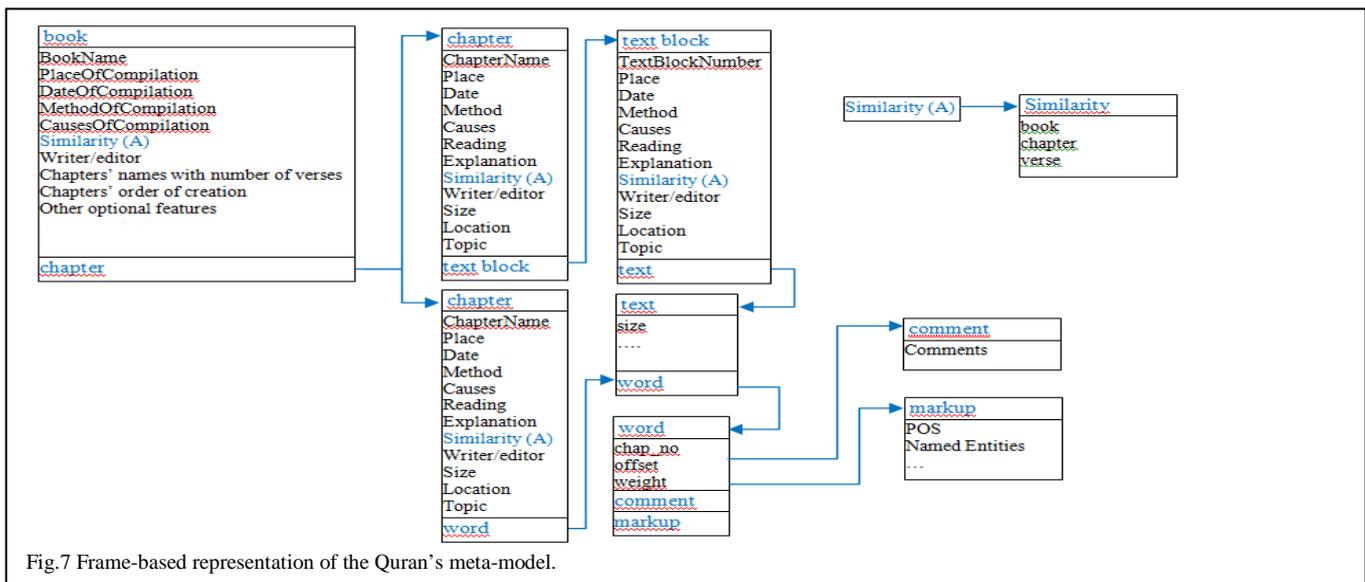


Fig.7 Frame-based representation of the Quran's meta-model.

For chapters and verses “Place”, “Date”, “Method”, “Causes” relate to revelation instead of compilation. A “similarity” slot got three attributes “book”, “chapter”, and “verse”. For the “book” frame, values of “chapter” and “verse” attributes are empty; and for the “chapter” frame the value of “verse” attribute is kept empty.

### 9.5 XML Model of the Frame Representation Schema

XML representation of the static part of the meta-model:

```
<quran>
  <metadata>
    <PlaceOfCompilation>Place where it was compiled</PlaceOfCompilation>
    <DateOfCompilation>Date when it was compiled</DateOfCompilation>
    <MethodOfCompilation>Through which means it was compiled</MethodOfCompilation>
    <CauseOfCompilation>Reasons of its compilation</CauseOfCompilation>
    <ChaptersOrderOfRevelation>number of chapters</ChaptersOrderOfRevelation>
    <ChaptersOrderOfNumbering>chapters' names</ChaptersOrderOfNumbering>
    <Similarity>
      <SimilarTo book="book name" chapter="chapter number"/>
      ...
    </Similarity>
  </metadata>
</quran>
```

4. place of compilation
5. date of compilation
6. method of compilation
7. causes of compilation

The Quran could also be viewed as an aggregation of chapters consisting of collection of words. Using this view it is

possible to annotate a word or group of words independently of the static metadata. Therefore each word could be associated with comments and markup tags representing the dynamic metadata for improving understanding and retention.

### 9.3 The Quran's Meta-Model

We need to substitute the names of attributes in the meta-model described in fig. 1 with our defined metadata. The meta-model is a representation of the knowledge base that users or learners need to keep in mind.

Moreover I suggest that any exegesis of the Quran to include all the metadata particularly the static ones defined in the above proposed meta-model. In fact if each verse is explained through its metadata, then understanding and retention would be easier and more effective.

```
<metadata>
  <ChapterName>chapter name</ChapterName>
  <PlaceOfRevelation>place of revelation</PlaceOfRevelation>
  <DateOfRevelation>place of revelation</DateOfRevelation>
  <MethodOfRevelation>Through which means it was revealed</MethodOfRevelation>
  <CauseOfRevelation>Reasons of its revelation</CauseOfRevelation>
  <NumberOfVerses>number of verses</NumberOfVerses>
  <Similarity>
    <SimilarTo book="book name" chapter="chapter number"/>
    ...
  </Similarity>
  <Size>Whether the chapter is long/medium/short</Size>
  <Location>Whether the chapter is at the beginning/middle/end of the Quran</Location>
  <Topic>To which topic does the chapter belong to</Topic>
</metadata>
<verse n="verse number">
  <Text length="size of the text"><word1>first word</word1>...<wordN>wordN
  </Text>
  <metadata>
    <PlaceOfRevelation>Place where it was revealed</PlaceOfRevelation>
    <DateOfRevelation>Date when it was revealed</DateOfRevelation>
    <MethodOfRevelation>Through which means it was revealed</MethodOfRevelation>
    <CauseOfRevelation>Reasons of its revelation</CauseOfRevelation>
    <ReadingWays>Ways of reading it</ReadingWays>
    <Explanation>What is the meaning or message it contains</Explanation>
  </metadata>
</verse>
```

```

<Similarity>
  <SimilarTo book="book name" chapter="chapter number" verse="verse number"/>
  ...
</Similarity>
<Size>Whether the verse is long/medium/short</Size>
<Location>Whether the verse is at the beginning/middle/end of the Quran</Location>
<Topic>To which topic does the verse belong to</Topic>
<Editor>The Writer or compiler of the verse</Editor>
</metadata>
</verse>
...
</chapter>
...
</quran>

```

There might be several <SimilarTo> elements with each one having three attributes book name, chapter number, and verse number.

An example of an XML representation of static annotation is shown below.

```

<quran>
  <metadata>
    <PlaceOfCompilation>Medina</PlaceOfCompilation>
    <DateOfCompilation>653 AD</DateOfCompilation>
    <MethodOfCompilation>Manual</MethodOfCompilation>
    <CauseOfCompilation>Some muslims were not able to memorize the full Quran, many of the
      those who memorised died in some battles
    </CauseOfCompilation>
    <ChaptersOrderOfRevelation>96, 74, ... 110</ChaptersOrderOfRevelation>
    <ChaptersOrderOfNumbering>1, 2, ...114</ChaptersOrderOfNumbering>
    <Similarity/>
  </metadata>
  <chapter n="1">
    <metadata>
      <ChapterName>Al-Fatiha</ChapterName>
      <PlaceOfRevelation>Mecca</PlaceOfRevelation>
      <DateOfRevelation>615 AD (approximately)</DateOfRevelation>
      <MethodOfRevelation>Recitation from Angel Gabriel</MethodOfRevelation>
      <CauseOfRevelation>What to recite when worshipping (offering compulsory prayers...)
        Allah
      </CauseOfRevelation>
      <NumberOfVerses>7</NumberOfVerses >
      <Similarity/>
      <Size>Short</Size>
      <Location>Beginning</Location>
      <Topic>Prayer</Topic>
    </metadata>
    <verse n="2">
      <Text length="17"><word4>ٱلْقَانِثِ</word4><word3>رَبِّ</word3><word2>ٱللَّهِ
        </word2><word1>ٱلْعَزْمَةِ</word1>
      </Text>
      <metadata>
        <PlaceOfRevelation>Mecca</PlaceOfRevelation>
        <DateOfRevelation>615 AD (approximately)</DateOfRevelation>
        <MethodOfRevelation>Recitation from Angel Gabriel</MethodOfRevelation>
        <CauseOfRevelation>What to recite when worshipping (offering compulsory prayers)
          Allah
        </CauseOfRevelation>
        <ReadingWays>As written</ReadingWays>
        <Explanation>Allah is the only owner of the worlds; therefore worship Him alone by
          reciting this verse
        </Explanation>
        <Similarity>
          <SimilarTo book="Quran" chapter="6" verse="45" />
          <SimilarTo book="Quran" chapter="10" verse="10" />
          <SimilarTo book="Quran" chapter="37" verse="182" />
          <SimilarTo book="Quran" chapter="39" verse="75" />
          <SimilarTo book="Quran" chapter="40" verse="65" />
        </Similarity>
        <Size>Short</Size>
        <Location>Beginning</Location>
        <Topic>Prayer</Topic>
        <Editor>The Writer or compiler of the verse (Uthman third Caliph)</Editor>
      </metadata>
    </verse>
    ...
  </chapter>
</quran>

```

In the above XML fragment, the <SimilarTo> element is confined to verses that contain the displayed verse. However it is possible to include all verses that share some percentage of similarity with it, for more details on this issue see [4].

The translation of the verse in English is: "Praise be to Allah, the Cherisher and Sustainer of the worlds".

As mentioned previously, the metadata would be derived from the Quran itself, if they are not available in other Islamic resources such as narrations (hadiths) of the prophet.

Some text mining techniques explained in [5] have been used to derive chapters' dates and places of revelation. In fact one of the main objectives of this study is to apply computational methods to the text of the Quran in order to derive the metadata and explanation of each verse.

XML representation of the dynamic part of the meta-model:

```

<quran>

```

```

<metadata>
  <PlaceOfCompilation>Place where it was compiled</PlaceOfCompilation>
  <DateOfCompilation>Date when it was compiled</DateOfCompilation>
  <MethodOfCompilation>Through which means it was compiled</MethodOfCompilation>
  <CauseOfCompilation>Reasons of its compilation</CauseOfCompilation>
  <ChaptersOrderOfRevelation>number of chapters</ChaptersOrderOfRevelation>
  <ChaptersOrderOfNumbering>chapters' names</ChaptersOrderOfNumbering>
  <Similarity>
    <SimilarTo book="book name" chapter="chapter number" verse="verse number"/>
    ...
  </Similarity>
</metadata>
<chapter n="chapter number">
  <metadata>
    <ChapterName>chapter name</ChapterName>
    <PlaceOfRevelation>place of revelation</PlaceOfRevelation>
    <DateOfRevelation>place of revelation</DateOfRevelation>
    <MethodOfRevelation>Through which means it was revealed</MethodOfRevelation>
    <CauseOfRevelation>Reasons of its revelation</CauseOfRevelation>
    <NumberOfVerses>number of verses</NumberOfVerses>
    <Similarity>
      <SimilarTo book="book name" chapter="chapter number" verse="verse number"/>
      ...
    </Similarity>
    <Size>Whether the chapter is long/medium/short</Size>
    <Location>Whether the chapter is at the beginning/middle/end of the Quran</Location>
    <Topic>To which topic does the chapter belong to</Topic>
  </metadata>
  <word1 chap_no="chapter number" offset="offset" weight="x">
    <comment>comments</comment>
    <markup>word1</markup>
  </word1>
  ...
</chapter>
...
</quran>

```

Dynamic annotations are only made for the translation version of the Quran. An example of an XML representation of dynamic annotation is shown below.

```

<quran>
  <metadata>
    <PlaceOfCompilation>Medina</PlaceOfCompilation>
    <DateOfCompilation>653 AD</DateOfCompilation>
    <MethodOfCompilation>Manual</MethodOfCompilation>
    <CauseOfCompilation>Some muslims were not able to memorize the full Quran, many of the
      those who memorised died in some battles
    </CauseOfCompilation>
    <ChaptersOrderOfRevelation>96, 74, ... 110</ChaptersOrderOfRevelation>
    <ChaptersOrderOfNumbering>1, 2, ...114</ChaptersOrderOfNumbering>
    <Similarity/>
  </metadata>
  <chapter n="1">
    <metadata>
      <ChapterName>Al-Fatiha</ChapterName>
      <PlaceOfRevelation>Mecca</PlaceOfRevelation>
      <DateOfRevelation>615 AD (approximately)</DateOfRevelation>
      <MethodOfRevelation>Recitation from Angel Gabriel</MethodOfRevelation>
      <CauseOfRevelation>What to recite when worshipping (offering compulsory prayers...)
        Allah
      </CauseOfRevelation>
      <NumberOfVerses>7</NumberOfVerses>
      <Similarity/>
      <Size>Short</Size>
      <Location>Beginning</Location>
      <Topic>Prayer</Topic>
    </metadata>
    <Allah chap_no="1" offset="19" weight="1.0">
      <comment>Allah is the Creator and Sustainer of the univers</comment>
      <enamelx>transcendant existing being</enamelx>
    </Allah>
    ...
  </chapter>
  ...
</quran>

```

## 10. Results and Discussion

The system has been applied to the Quran, and the results obtained show that multiple perspectives of information modeling can be successfully applied to religious texts. Five people (including myself) among them some with little computer literacy have tested the system, and were positively impressed. Even though the system is not yet complete, all the testers acknowledged that the system can effectively improve understanding and memorization of the Quran's verses. Testing the proposed system was carried out in three steps.

The first step was to evaluate the usability aspect of the system, the second and third steps were measuring understanding and retention respectively.

Concerning the usability of the system, all the users found the designed user interfaces adequate and quite simple to use. In fact the user interfaces were mainly consisting of few buttons, labels, text boxes, and browser components that users could

customize according to their desired models. Moreover the process of creating metadata tags, inserting and updating data for each metadata was very similar to that of creating tables, tables' fields, and inserting, updating, browsing data in a database systems such as MySQL. Therefore among the testers three of them were familiar with databases, and found the system quite intuitive and effective.

As for the measurement of assisting understanding, which is actually one of the main objectives of this study, chapter 112 of the Quran along with English translation was displayed, and users were requested to study it, and ask questions. They all said they understood what the chapter was about, but they would like to know the purpose of the chapter, and when, how to apply it in their daily life. So the questions were answered by supplying causes, place, date, and method of revelation of the chapter. Through the supplied answers they were able to figure out the purpose and application of the chapter in daily life.

However they suggested a short summary of the chapter based on the four supplied information would further improve and hasten understanding of the chapter.

The summarization of a chapter and search results are actually part of the ongoing activities. Hopefully they will be soon available.

Now the whole system was presented to them, they entered the information about the chapter on their own, and annotated some words, and finally visualized the whole on a website. They acknowledged that this exercise further improved their understanding as well as retention of the text, since each one considered himself as the author of the content of the website.

It is commonly known in chemistry or physics that students understand better when they model processes and reactions that they have learned [7].

Breaking down a text into smallest meaningful parts, enumerating all possible required details about each part, and selectively visualizing each part, are reasonably effective ways of understanding and retaining the text.

In fact nowadays one is attempted to argue that correct understanding of any religious text requires meaningful modeling and visualization of that text. This study is an attempt to perform such a task, and hopefully the system will be improved from both theoretical and practical aspects.

Of course I do not deny that numerous studies in this direction have been conducted, for instance the multitude existing software for learning religions (could be found using google search engine...) are definitely results of such studies. However those software programs do not offer users or learners the chance to create the knowledge on their own. Moreover the programs are designed for particular religious books; hence they cannot be easily adapted to learn other religious texts.

So I hope this short ongoing study would motivate others to engage in devising and offering software systems for cross-religions learning.

## 11. Conclusion and Future Work

In this study a modeling and visualization system for assisting understanding, memorization, and sharing of knowledge obtained from religious books has been presented. The system is based on books complying with religious books' schema namely the Quran. The modeling is achieved with the Content Management System, and the visualization is based on a Mashup website. A meta-model representing knowledge base

created from a considered religious text has also been shown. The meta-model is an attempt to devise a common architecture for structural and terminological dimensions of religious texts. Even though incomplete some people have tested the system, and the feedback was encouraging.

The next steps in this study will be the completion of the system by offering a fully functional Expert System, which could allow distance users to interactively consult and learn religious texts such as the Quran. The Content Management System has been developed as an eclipse RCP plugin, I am working on converting it into a web based management system in order to offer online maintenance of the Mashup website.

Refining the meta-model by further extending the multi-perspective approach of knowledge modeling could considerably improve the goal of this study, which is to assist understanding and retention of religious texts.

## References

- [1] J. Zachman. "A Framework for Information Systems Architecture" IBM Systems Journal, 26(3):276–292, 1987
- [2] <http://www.openify.com/files/Designing%20A%20Knowledge%20Management%20System2.pdf>
- [3] J. Claudio, T. Angeloni, "Understanding the difference between Information Management and Knowledge Management", TerraForum Consultores, Toronto, ON, Canada
- [4] Mohamadou Nassourou, "Assisting Analysis and Understanding of Quran Search Results with Interactive Scatter Plots and Tables", accepted in IV2011 15th International Conference Information Visualisation, London, UK  
<http://nbn-resolving.de/urn:nbn:de:bvb:20-opus-55840>
- [5] Mohamadou Nassourou, "A Knowledge-based Hybrid Statistical Classifier for Reconstructing the Chronology of the Quran", accepted in WEBIST/WTM 2011, The Netherlands  
<http://nbn-resolving.de/urn:nbn:de:bvb:20-opus-54712>
- [6] M. Lansdale, "The psychology of personal information management" University of Technology, Loughborough, UK
- [7] J. Heitz, "Modeling exercises improve understanding", university of Wisconsin
- [8] Y. Chali, E. Pascual, J. Virbel, "Text structure modeling and language comprehension processes", Université Paul Sabatier 118, Toulouse, France
- [9] John R. Kirby, "What have we learned about reading comprehension?," Faculty of Education, Queen's University
- [10] Yulong Gu, James Warren, Jan Stanek, Graeme Suthers, "A System Architecture Design for Knowledge Management (KM) in Medical Genetic Testing (MGT) Laboratories", Proceedings of the 10th International Conference on Computer Supported Cooperative Work in Design
- [11] Roelof P. uit Beijerse, "Questions in knowledge management: defining and conceptualising a phenomenon"
- [12] Mohamadou Nassourou, "Markup Overlap: Improving Fragmentation Method", accepted for nocturne presentation in Balisage 2010, Montreal, Canada,  
<http://nbn-resolving.de/urn:nbn:de:bvb:20-opus-49084>
- [13] Mohamadou Nassourou, "Design and Implementation of Textual Documents Collation Systems", accepted (poster presentation) in International Conference on Software, Services and Semantic Technologies (S3T 2011), Bourgas, Bulgaria, 2011,  
<http://nbn-resolving.de/urn:nbn:de:bvb:20-opus-56601>
- [14] Mohamadou Nassourou, "Computer-based Textual Documents Collation System for Reconstructing the Original Text from Automatically Identified Base Text and Ranked Witnesses", accepted in SDH 2011 Supporting Digital Humanities, 17-18 November 2011, Copenhagen, Denmark
- [15] <http://www.tei-c.org/index.xml>