# Arabic Database for Automatic Printed Arabic Text Recognition Research and Benchmarking

BY

Amin Ghalib Al-Hashim

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

**KING FAHD UNIVERSITY OF PETROLEUM & MINERALS**

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

**MASTER OF SCIENCE**

**In**

COMPUTER SCIENCE

June 2009

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **AMIN G. AL-HASHIM** under the direction of his thesis advisor and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER SCIENCE**.

Thesis Committee

Mahmoud 28/6/200
Dr. Sabri Mahmoud (Thesis Advisor)

Alshayb 29/6/2009
Dr. Mohammad Alshayeb (Member)

29-6-200
Dr. Salahadin Mohammed (Member)

Dr. Kanaan Faisal
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

10/8/09
Date

# DEDICATION

I dedicate this work to all my family members, especially my parents, my wife and my newborn daughter, S. Fatima.

# ACKNOWLEDGMENT

In the first place, all thanks are due to ALLAH Almighty for his continuous blessings.

Acknowledgment is due to King Fahd University of Petroleum & Minerals for supporting this research.

I wish to express my appreciation to my major advisor, Dr. Sabri Mahmoud, for all the invaluable help and support he gave me throughout the course of this work. I also wish to thank the other members of my thesis committee Dr. Mohammad Alshayeb and Dr. Salahadin Mohammed.

Further, I would like to thank Dr. Kanaan Faisal, chairman of ICS Department and his secretary staff Mr. Madakkara and Mr. Ansari for their patient during the faxing process.

Finally, I wish to express my gratefulness to my family members, especially my parents, for their moral support. A special gratitude goes to my wife for her word of encouragement, patient and invaluable help in preparing the hard copy material for the database population process.

# TABLE OF CONTENTS

# LIST OF TABLES

Page No.

# LIST OF FIGURES

Page No.

# ABSTRACT

| | | |
|---|---|---|
| NAME | : | AMIN G. AL-HASHIM |
| TITLE | : | ARABIC DATABASE FOR AUTOMATIC PRINTED ARABIC TEXT RECOGNITION RESEARCH AND BENCHMARKING |
| MAJOR FIELD | : | COMPUTER SCIENCE |
| DATE OF DEGREE | : | June 2009 |

One of the major obstacles that face researchers in the automatic Arabic text recognition field is the lack of a public large-scale comprehensive Arabic text database. Such database saves the researcher's time and effort since he will not be enforced to create a private database that most probably will not cover most of the aspects of real life written communications. Moreover, the public large-scale comprehensive Arabic text database can act as a benchmark database. Through this benchmark database, the research results of different researchers can be tested and verified. In addition, the different techniques and researches can be compared. The aim of this work is to construct such database for printed Arabic text with the idea of future extension in mind. This work includes software that makes the manipulation of the created database easier.

# ملخص الرسالة

الإســــــــم : أمين غالب الهاشم

عنوان الدراسة : قاعدة بيانات لبحوث التعرف الآلي على النصوص العربية المطبوعة وأيضا وسيلة مقارنة

التخصــــــص : علوم الحاسب الآلي

تاريخ التخــرج : يونيو ٢٠٠٩

احدى العقبات الرئيسية اللتي تواجه الباحثين في مجال التعرف الآلي على النصوص العربية هو عدم توفر قاعدة ببانات نصوص عربية تكون عامة وكبيرة وشاملة. هذا النوع من قاعدة البيانات يحفظ وقت وجهد الباحث المبذول في عملية إنشاء قاعدة بيانات خاصة عادةً ما تكون غير شاملة لجميع نواحي إتصالات الحياة اليومية المكتوبة. كما يمكن استخدام قاعدة بيانات النصوص العربية العامة والكبيرة والشاملة كقاعدة بيانات معيارية. من خلال قاعدة البيانات المعيارية هذه، يمكن مقارنة البحوث والتقنيات المختلفة. هدف هذا العمل هو إنشاء قاعدة بيانات للنصوص العربية المطبوعة مع الأخذ في عين الإعتبار فكرة التمديد المستقبلي.

بالإضافة الى قاعدة البيانات، سوف يحتوي هذا العمل على البرمجيات الازمة لجعل عملية إدارة قاعدة البيانات المنشئة أكثر سهولة.

# CHAPTER 1

# INTRODUCTION

Nowadays, we live a data explosion in all aspects of life.  This data varies from electronic form to machine-printed and handwritten forms. Although, the electronic form is the most dominant, the other two are still present in some aspects of life such as traditional mails, bank checks, forms, manuscripts, etc.  Most probably, this type of printed forms needs to be presented in an electronic form for processing (e.g., searching).  Either this can be done manually or automatically through computers using the Automatic Text Recognition (ATR) techniques (referred to as Automatic Arabic Text Recognition (AATR) techniques in the case of Arabic language) or image search and retrieval techniques.

The manual and the automatic methods that are used to convert a printed document page to its corresponding electronic version cannot always ensure the exact correspondence between the printed version and the produced electronic version.  So some reviewers are needed to ensure near equivalent correspondence. On the other hand, the automatic method needs to be trained and tested (within the development process of the application) on a relatively large set of data. The data should cover a variety of document types and forms faced in real life in order to achieve the near equivalent correspondence of the printed version. For this reason and others, we aim at addressing this problem by coming up with a standard structure database that can handle large amount of documents.

AATR techniques can be used in different fields of our daily life. For example, instead of having a large number of personnel to sort the letters coming to a post office into their different destinations, AATR techniques can be used to make the process automatic and faster. Another example is the processing of the bank checks. Manually processing the data written in a check is considered a tedious task. This manual data processing can be replaced by an appropriate AATR application. In both examples, the ultimate goal is to come up with an AATR application that achieves near 100% accuracy. This goal is very difficult to achieve without having a relatively large amount of data on which the application is trained and tested. The goal of this work is to create such database.

The proposed database for AATR needs to have two very important features: simplicity and extensibility. It should be simple in the sense that it is easy to use, easy to read and operating system (OS) independent. A new OS independent format called JSON (JavaScript Object Notation) [JSON-a] may be used as the standard format for representing the database text files. In addition, it should be extensible in the since that different types of documents can be added to it without any modification (or with very simple modifications) to the original database.

The goal of this work is to construct a public large-scale comprehensive database specialized for printed Arabic text. Along with the database, software will be developed. This software will enable the database users to manipulate the database easily. In addition, the software will provide a number of image processing functions that can help the researchers of AATR field.

The rest of this thesis is organized as follows. Chapter 2 presents the motivation toward constructing a comprehensive public large-scale database for AATR and benchmark. A literature review of the available Arabic databases for AATR and their limitations is presented in Chapter 3. Chapter

4 addresses the list of attributes needed by any comprehensive public large-scale database developed for AATR applications and benchmark followed by the construction process of the PATDB (PATDB). Chapter 5 expresses in details the specifications and the definition of the PATDB. A discussion of the features and functions of the software provided with the database appears in Chapter 6. Chapter 7 gives the conclusion and suggests future work.

# CHAPTER 2

# MOTIVATION

For the best of our knowledge, up to the moment of writing this document, there is no comprehensive public large-scale printed Arabic text database (PATDB) that is freely available for AATR researchers and developers. On the other hand, other languages have many freely available databases. For example, English language has several databases such as CEDAR and NIST. The absence of such database is one of the major problems encountered in AATR field [Srih07][Lori06][Alba95]. Currently, researchers use their own data in their research. Hence, this makes the results of different researchers incomparable.

The reasons behind the need of a comprehensive public large-scale database for AATR can be summarized as follows:

First, it will remove from the AATR researchers and developers the burden of acquiring a suitable data for each task in the recognition process [Bipp95][Phil93a]. This is a genuine hindrance for many researchers in the

AATR field [Khar99][Märg01]. Building a database with its ground-truth value for Arabic is more difficult, time consuming and error prone than building it for English [Märg01].

Second, it will enable the researchers and developers to compare and contrast the effectiveness and the performance of their techniques [Hull94] [Phil93a] [Khar99][Märg01][Alba95].

Third, the algorithm results can be reported, verified and compared [Hull94][Phil93a].

Fourth, experiments with combined algorithms can be achieved [Hull94].

Fifth, the entire statically based recognition methods need large database for training [Bipp95][Märg01][Alba95].

Sixth, researchers and developers will not find themselves enforced to tune their algorithms and applications to suite others' customized databases. This way of activity does not promote a good research environment [Phil93a][Märg01].

Finally, the AATR is likely to be a complex process involving many steps that are independent and may need to be undone using backtracking algorithm.  Therefore, a standard and suitable representation scheme of the database is needed.  [Alma02]

# CHAPTER 3

# LITERATURE REVIEW

## 3.1 INTRODUCTION

Currently, there is no comprehensive public large-scale printed Arabic text database (PATDB) that is freely available for AATR researchers and developers. This is according to the best of our knowledge. On the other hand, other languages have many freely available databases. For example, English language has several databases such as CEDAR and NIST. On the contrary, Arabic language has only a small number of bounded, commercial and/or special purpose Arabic text databases.

A list of printed Arabic text databases followed by a list of Arabic handwritten text databases are presented in Section 2 and 3, respectively. These two lists represent the available databases in the literature of which the writer is aware. Arabic handwritten text databases are outside the scope of this work. They are surveyed here just to highlight the shortage of the available Arabic database for researchers in AATR field.

## 3.2 PRINTED ARABIC TEXT DATABASES

Following is a list of the available printed Arabic text databases in the literature of which the writer is aware.

### 3.2.1 ERIM DATABASE

The Environmental Research Institute of Michigan (ERIM) has created a database of machine-printed Arabic documents. The database is extracted from typewritten and typeset books and magazines. ERIM contains over 750 pages that consists of approximately 1,000,000 characters and over 200 distinct ligatures. Pages were scanned with a resolution of 300 dots per inch (dpi). The database is divided into 3 distinct sets, namely, training, statistics and testing set. It is available on a CD ROM for US$40.

ERIM database covers only two aspects of real life written communications, namely, books and magazines. However, many other aspects of real life written communications exist such as letters and newspapers. This coverage limitation is considered a disadvantage when developing a general-purpose AATR application. In addition, one may consider the commercial aspect of ERIM database as another disadvantage.

### 3.2.2 DARPA CORPUS

DARPA (Defense Advanced Research Projects Agency) Arabic corpus was created by Scientific Application International Company (SAIC) for the US Department of Defense [Davi97]. DARPA contains 345 pages of images (around 670,000 characters) with ground-truth. Images were produced by scanning the pages with a resolution of 600 dpi. Images are zones of a single column of text and they vary in quality. The corpus was collected from book chapters, magazine articles, newspapers and computer generated documents having only 4-fonts.

DARPA corpus has the same disadvantage of ERIM database. DARPA corpus does not cover all the aspect of real life written communications such as letters and advertisements.

## 3.3 ARABIC HANDWRITTEN TEXT DATABASES

Following is a list of the available handwritten Arabic text database in the literature of which the writer is aware.

### 3.3.1 IFN/ENIT DATABASE

IFN/ENIT database was created by the Institute of Communications Technology (IFN) at Technical University Braunschweig in Germany and the Ecole Nationale d'Ingénieurs de Tunis (ENIT) in Tunisia [Pech02]. It was made freely available for non-commercial research groups. The IFN/ENIT database consists of 26,459 images of 946 Tunisia town/village names written by 411 different writers. The database is partitioned into a number of sets. The partition is done to enable researchers to use these sets as training and testing data. [Märg05] gives the most important statistics of the IFN/ENIT database. The IFN/ENIT database was used as a basis for the "ICDAR 2005 Arabic Handwritten Recognition Competition." More than 30 groups were working on it worldwide by the time of the competition launch in 2005 [Märg05].

One of the limitations that can be pointed out for the IFN/ENIT database is that, it was captured in laboratory environment in which writers prepared samples on standard forms, which are then digitized. The awareness by writers that their handwriting would be used to develop AATR algorithms could have introduced biases into the data. The desire to perform well, because of the similarity to a classroom-testing environment, may yield abnormally neat samples. Alternatively, writers may try to fool the computer by making their samples unusually sloppy. Another limitation (some may consider it as an advantage) is the scope focus. The database is limited only to a set of Tunisian town/village names; it is not covering printed documents, articles, journals, etc. The focus of the scope here is considered as a limitation when building a general-purpose AATR application. However, it is considered as an advantage when it is used in developing a special-purpose AATR application for post offices especially in Tunisia. In addition, the North African Arab countries handwriting style used in the database considered as a limitation for the East Arabs. It is hard for them to recognize some of the city names visually.

### 3.3.2 ALMA’ADEED AHDB

Alma’adeed et al. AHDB (Arabic Handwritten Data Base) [Alma02] consists of samples from 100 writers, which includes 10,000 words used for numbers and quantities in checks filling. It also includes the most popular words in Arabic writing, sentences used in writing checks with Arabic words and free handwriting pages in any area of the writer’s choice.

### 3.3.3 CENPARMI DATABASE

In 2003, Al-Ohali et al. of the Centre for Pattern Recognition and Machine Intelligence (CENPARMI) in Montre’al developed a database of images from 3,000 checks out of 7,000 checks provided by Al-Rajhi Banking Corporation. The development was done after removing all personal (private) information from the provided checks. [Aloh03] This database was organized into four sub-databases. These sub-databases are Arabic legal-amounts (2499 legal amount), courtesy amounts (2499 courtesy amount written in Indian digits), Arabic sub-words (29,498 sub-words within the domain of legal amount) and Indian digits (15,175 digits). Each sub-database is divided into training and testing set. The training set includes 66-75% of the available data. A further division of the training and testing sets also exists depending on a specific criterion for each sub-database. The database is available for interested researchers for a nominal fee (a single license costs US$350) upon request to CENPARMI.

The banking scope and more specifically the checks scope that this database focuses on, made it less useful for AATR researchers and developers. The scope focus of CENPARMI database is considered a disadvantage for researches and developers working on general-purpose AATR applications.

### 3.3.4 UBC DATABASE

Kharma N. et al. of University of British Columbia (UBC), Canada, developed a handwritten Arabic database in 1999 [Khar99]. The database contains 37,000 Arabic words, 10,000 digits in 2 types (‘Magharibi’ (North African) and ‘Mashriqi’ (Middle East)), 2,500 signatures and 500 free-form Arabic sentences. A full copy of the database is offered to researchers in Canada and only partially to the rest of the world.

In addition to the databases mentioned above, there are many printed Arabic and handwritten character and text databases generated by researchers for their own researches. [Srih07] listed a group of Arabic databases that has been used as training data or lexicons by different published AATR applications.

Depending on the advantages and disadvantages of the above surveyed Arabic text databases in addition to other languages' text databases, we come up with a set of attributes. This set of attributes is considered when developing the PATDB. The set of attributes is discussed in the next chapter.

# CHAPTER 4

# PATDB ATTRIBUTES AND CONSTRUCTION PROCESS

## 4.1 INTRODUCTION

When a developer builds a service for others to use, he normally comes up with a set of attributes for this service. These attributes is set up in a way that maximizes the usage of the provided service by the target group of users. Similarly, in order to develop a database of printed Arabic text that helps AATR researchers and developers in their work, the database should have a set of attributes. This set of attributes should be set up in a way that maximizes the benefit of the developed printed Arabic text database (PATDB) by the AATR researchers and developers.

The set of attributes that should be considered when building a database for supporting the AATR community is listed in Section 2. The set of attributes mentioned in Section 2 is followed by the construction process of the PATDB, which is discussed in Section 3.

## 4.2 PROPOSED ATTRIBUTES

We propose the use of the following attributes for any database aiming to support the AATR community. These attributes were in mind at the development time of the PATDB. However, they were not assessed at the end of the development process of the PATDB.

The attributes are:

1. Simple, i.e., easy to user, easy to read, and operating system independent;
2. Extensible, i.e., eligible for adding new document types without any modifications;
3. Standard, i.e., any addition to the database will follow a set of pre-defined rules;
4. Public;

5. Comprehensive, i.e., covers different types of documents with different writing styles;

6. Large-scale;

7. Uniform, i.e., allows different sub-tasks to access different types of data in the same way;

8. Reflects the physical document hierarchy as well as the logical structure;

9. Includes explicit training and testing data sets;

10. Stores the scanned images and their ground-truth information (corresponding text, style information, etc.) separately;

11. May acts as a benchmark for evaluating the performance of AATR applications;

12. May include a set of tools for:

    a. Manipulating the database;

b. Simulating the natural paper degradation models such as paper aging, multi-coping, skewing and pepper-and-salt noise.

The construction process of the PATDB, which is addressed in the following section, takes into account the above attributes.

## 4.3 CONSTRUCTION PROCESS

The ultimate goal of this work is to construct a database that can act as a foundation for a standard structure PATDB for AATR. This section addresses in details the construction process of the PATDB. The construction process takes into consideration the attributes mentioned in the previous section.

The construction process of the PATDB is shown by the flow diagram in Figure 1. First, an appropriate page is selected. Second, the selected page is scanned. Third, the scanned page image is saved into the database if it is equivalent to its corresponding hard copy version. Along with the scanned page image, the truth-data of the page is included in the database. The truth-data represents two things: the ground-truth value of the page and the metadata (record) files. The metadata files describe the page in general and the page zones in specific. Appendix A addresses the metadata files in details.

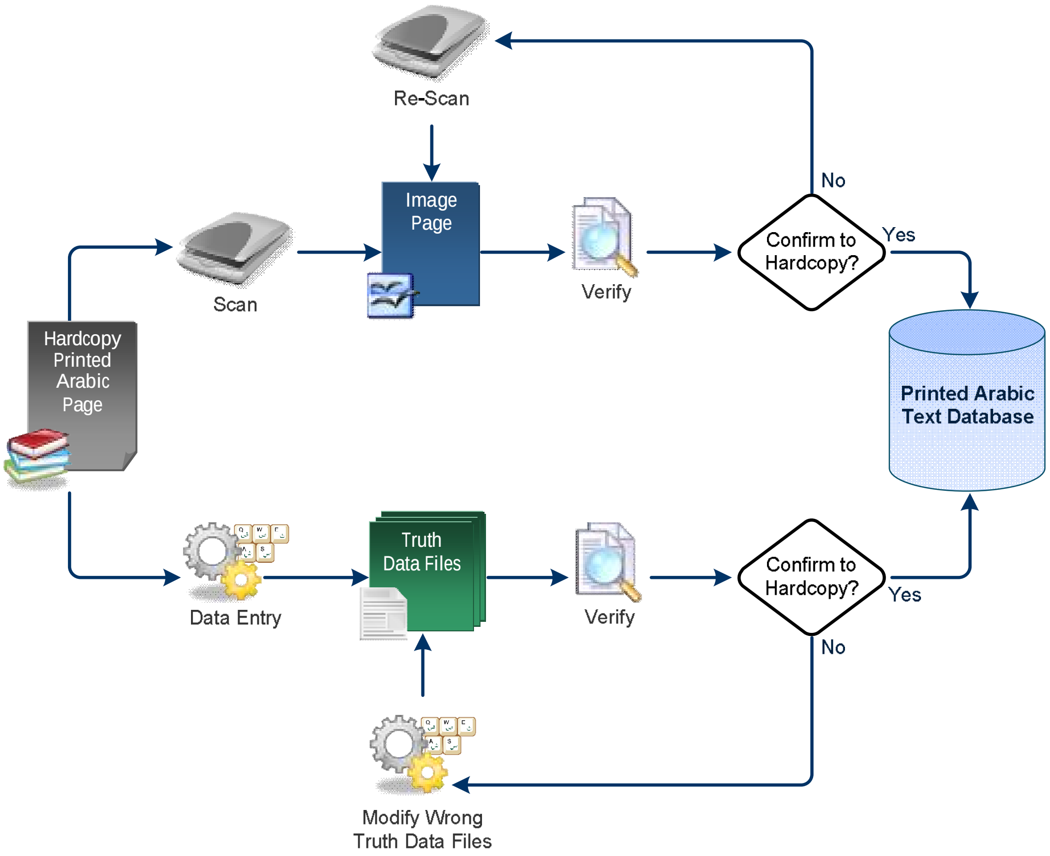


Figure 1: PATDB construction process flow diagram

While constructing the PATDB, the emphasis is put on two main features: simplicity and extensibility. In this manner, the database will be eligible for enhancement and addition. To achieve these two features, the implementation methodology needs to be defined explicitly. Figure 2 outlines the proposed database implementation methodology adopted from [Phil93b]. As shown in the figure, each selected page may be faxed first. After that, the page is scanned and given a unique number that will act as its identifier. This number will appear in all the files that describe this page later. After the selection process, the page will be attributed and zoned. Then, the ground-truth value of the identified zones will be keyed. All the resulting files (page image, attribute and ground-truth value files) will be named according to a pre-defined naming convention before uploading them to the database. The defaults of the naming conventions are given in Chapter 5.

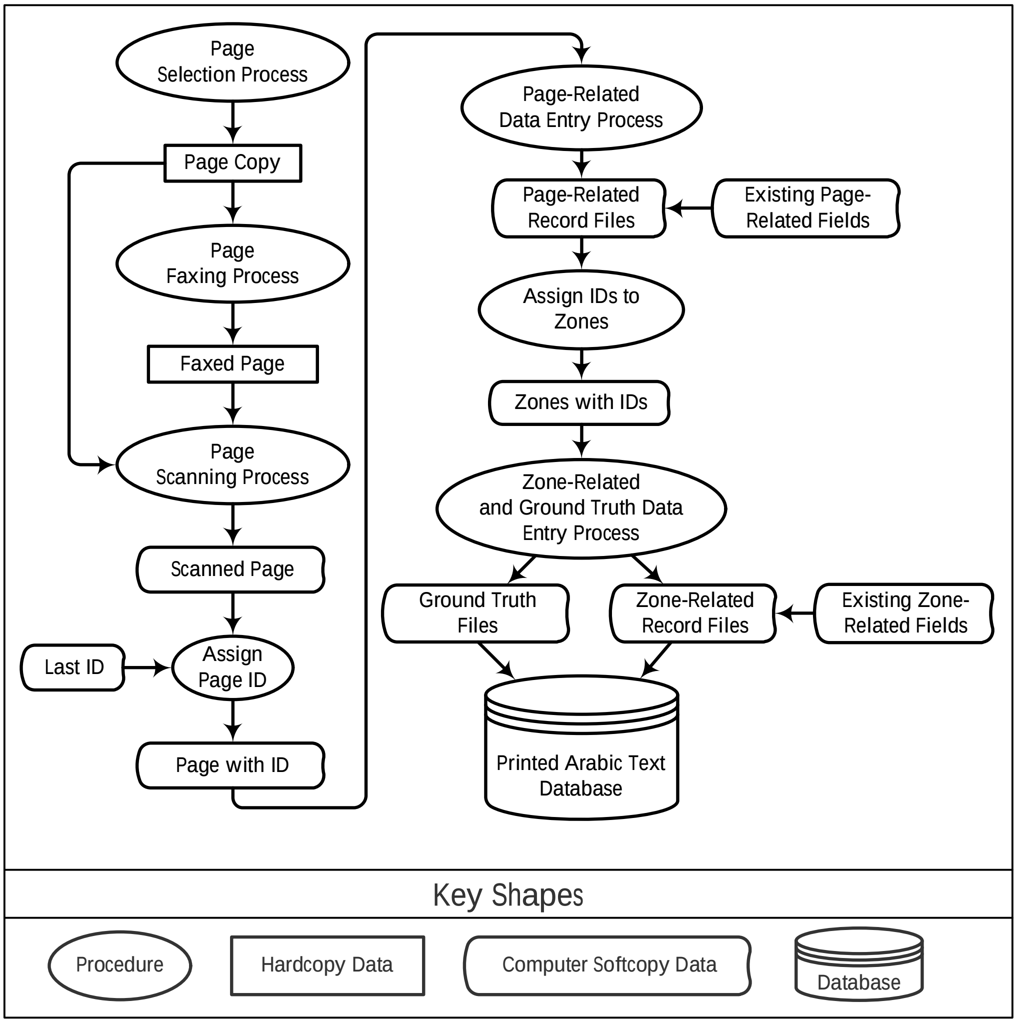


Figure 2: PATDB high-level implementation process

The database includes distinct pages from different sources of real life communications (articles, newspapers, etc.). The weight of each source will be determined by its importance in the current real life communications.

Along with the document part (page images, attribute and ground-truth files), the database will include a software part which is addressed in Chapter 6. The software part will include two basic set of functions. The first set of functions provides the user with the browsing and manipulation capabilities of the database. The other set of functions consists of tools that can assist the ATR systems such as de-skewing and pepper/salt de-noising.

# CHAPTER 5

# PATDB SPECIFICATIONS

## 5.1 INTRODUCTION

This chapter addresses the specifications of the printed Arabic text database (PATDB). The format of the page images stored in the database is expressed in Section 2. Section 3 defines the naming conventions of the files stored in the database. Section 4 presents the database storage requirements. The distribution of the page images across the PATDB is shown in Section 5.

## 5.2 PAGE IMAGE FORMATS

The document pages, that are available in the PATDB, are scanned and stored in three different formats: (1) binary (black & white) format with color depth of 1-bit per pixel; (2) grayscale format with 8-bit (1-byte) per pixel color depth (0 to 255 gray levels); and (3) color format with 24-bit (3-byte) per pixel color depth. The resolutions at which the document pages are scanned are 200, 300 and 600 dots per inch (dpi).

Common file types in bitmap format are JPEG, GIF, TIFF and WMF. The uncompressed TIFF file format (file-extension .tif) is chosen for the PATDB since it can store complex information for the CMYK (Cyan, Magenta, Yellow and Key (black)) color model and can use JPEG compression techniques. This makes TIFF format one of the most robust and well-supported image formats available [Howe00].

To simulate the salt/pepper noise incorporated from faxes, some of the document pages are first faxed then scanned.

All the page images in the PATDB were scanned using HP Scanjet N8400 series scanner. The faxing of the page images was done internally from one Panasonic fax to another Panasonic fax.

## 5.3 FILE NAMING & PATDB HIERARCHY

Table 1 shows the naming conventions that are used for the different types of scanned documents.

The scan-type naming conventions are shown in Table 2.

TABLE 1: CATEGORIES (DOCUMENT TYPES) ABBREVIATIONS

| Abbreviated Name | Full Name |
|---|---|
| ADS | Advertisements |
| BOOK | Book Chapters |
| LTR | Letters |
| MAG | Magazines |
| NEWS | Newspapers |
| OTHR | Others (Posters, thesis, …) |
| REP | Reports |

TABLE 2: SCAN-TYPE ABBREVIATIONS

| Abbreviation | Meaning | Color Depth (bit/pixel) | Number of Colors |
|---|---|---|---|
| BW | Black & White (binary) | 1 | 2 |
| GS | Grayscale | 8 | 256 gray shades |
| CL | Color | 24 | Millions of colors |

The following criteria are adhered to at the time of building the database:

- All the document files that belong to a specific category are stored in their corresponding directory. When creating a new directory, the corresponding (abbreviated) category name is created after recording its name in the 'Categories Abbreviations' table. For example, all the report document files will be stored in the reports' directory named REP (according to the 'Categories Abbreviations' table).

- The name of the document file within its corresponding directory is formatted as follows:

  ***CategoryNamennnn.FileExtension***,

  where ***CategoryName*** is the abbreviated category name according to the 'Categories Abbreviations' table; ***nnnn*** is a 4-digit unique number; and ***FileExtension*** is the extension of the file being stored. This document file acts as the source from which the ground-truth value is taken for the page images. An example of a report document name is REP0092.doc.

- For each added document file, a directory is named after the document file name excluding the file extension. This directory will be created in the corresponding category directory. For example, for the document named REP0092.doc, a directory named REP0092 in the REP directory is created. This directory includes the following items for each page of the document file:

  1. A scanned image is named as follows: ***DocNamePppp_CccFff_RrrrrScanTypelll.tif***, where ***DocName*** is the document name without file extension; ***ppp*** is a 3-digit page number; ***cc*** is a 2-digit copy number; ***ff*** is a 2-digit fax number; ***rrrr*** is a 4-digit resolution level in dot per inch (dpi); ***ScanType*** is an abbreviated san type according to the ‘scan types abbreviations’ table; and ***lll*** is a 3-digit color depth in bits. The copy number, referred to by ***cc***, represents the number of consecutive copies that were done on the page before scanning. For example, if an original printed page is scanned directly, then its copy number is zero. Similarly, if the original page is copied one time and the copy is then scanned, the copy number will be 1 and if this copied page is copied again, its copy number will be

2 and so on.

Similarly, the fax number, referred to by ***ff***, represents how many numbers of times this page was faxed with the restriction that each new faxed version is done on the previous faxed version.

For example, if the first page of REP0092.doc file is copied 5 times (5th generation), faxed 2 times (2nd generation), scanned with a resolution level of 300 dpi (dot per inch) and saved as a grayscale image with 8-bit color depth; then it is named as REP0092P001_C05F02_R0300GS008.tif.

2. A text file representing the page condition record is named as follows:

   ***DocNamePppp_CccFff_RrrrrScanTypelll_PC.txt***,

   where ***DocName***, ***ppp***, ***cc***, ***ff***, ***rrrr***, ***ScanType*** and ***lll*** are as defined earlier.

   For example, the file representing the page condition record of the page in the example of point 1 above is REP0092P001_C05F02_R0300GS008_PC.txt.

3. A text file representing the page attribute record is named as follows:

   ***DocNamePppp_CccFff_RrrrrScanTypelll_PA.txt***,

   where ***DocName***, ***ppp***, ***cc***, ***ff***, ***rrrr***, ***ScanType*** and ***lll*** are as defined earlier.

   For example, the file representing the page attribute record of the page in the example of point 1 above is REP0092P001_C05F02_R0300GS008_PA.txt.

4. A text file representing the page bounding box record is name as follows:

   ***DocNamePppp_CccFff_RrrrrScanTypelll_PBB.txt***,

   where ***DocName***, ***ppp***, ***cc***, ***ff***, ***rrrr***, ***ScanType*** and ***lll*** are as defined earlier.

   For example, the file representing the page bounding box record of the page in the example of point 1 above is REP0092P001_C05F02_R0300GS008_PBB.txt.

5. A directory is named as follows:

   ***DocNamePppp_CccFff_RrrrrScanTypelll***, where ***DocName***, ***ppp***, ***cc***, ***ff***, ***rrrr***, ***ScanType*** and ***lll*** are as defined earlier.

This directory includes the following items for each identified zone in the page:

1. A text file representing the zone bounding box record is named as follows:

   ***DocNamePppp_CccFff_RrrrrScanTypelll.Zzz_ZBB.txt***,

   where ***DocName***, ***ppp***, ***cc***, ***ff***, ***rrrr***, ***ScanType*** and ***lll*** are as defined earlier and ***zz*** is a 2-digit zone number.

   For example, the file representing the zone bounding box record of a zone having number 3 on the page of the example of point 1 above is REP0092P001_C05F02_R0300GS008.Z03_ZBB.txt.

2. A text file representing the zone attributes record is named as follows:

   ***DocNamePppp_CccFff_RrrrrScanTypelll.Zzz_ZA.txt***,

   where ***DocName***, ***ppp***, ***cc***, ***ff***, ***rrrr***, ***ScanType*** and ***lll*** are as defined earlier and ***zz*** is a 2-digit zone number.

For example, the file representing the zone attributes record of a zone having number 3 on the page of the example of point 1 above the is REP0092P001_C05F02_R0300GS008.Z03_ZA.txt.

3. A text file representing the zone ground-truth value is named as follows: ***DocNamePppp_CccFff_RrrrrScanTypelll.Zzz_ZTV.txt***, where ***DocName***, ***ppp***, ***cc***, ***ff***, ***rrrr***, ***ScanType*** and ***lll*** are as defined earlier and ***zz*** is a 2-digit zone number.

   For example, the file representing the ground-truth value of a zone having number 3 on the page of the example of point 1 above is REP0092P001_C05F02_R0300GS008.Z03_ZTV.txt.

Figure 3 represents the above criteria graphically. The figure shows a case of the report category (REP) in details. Other types are similarly constructed.

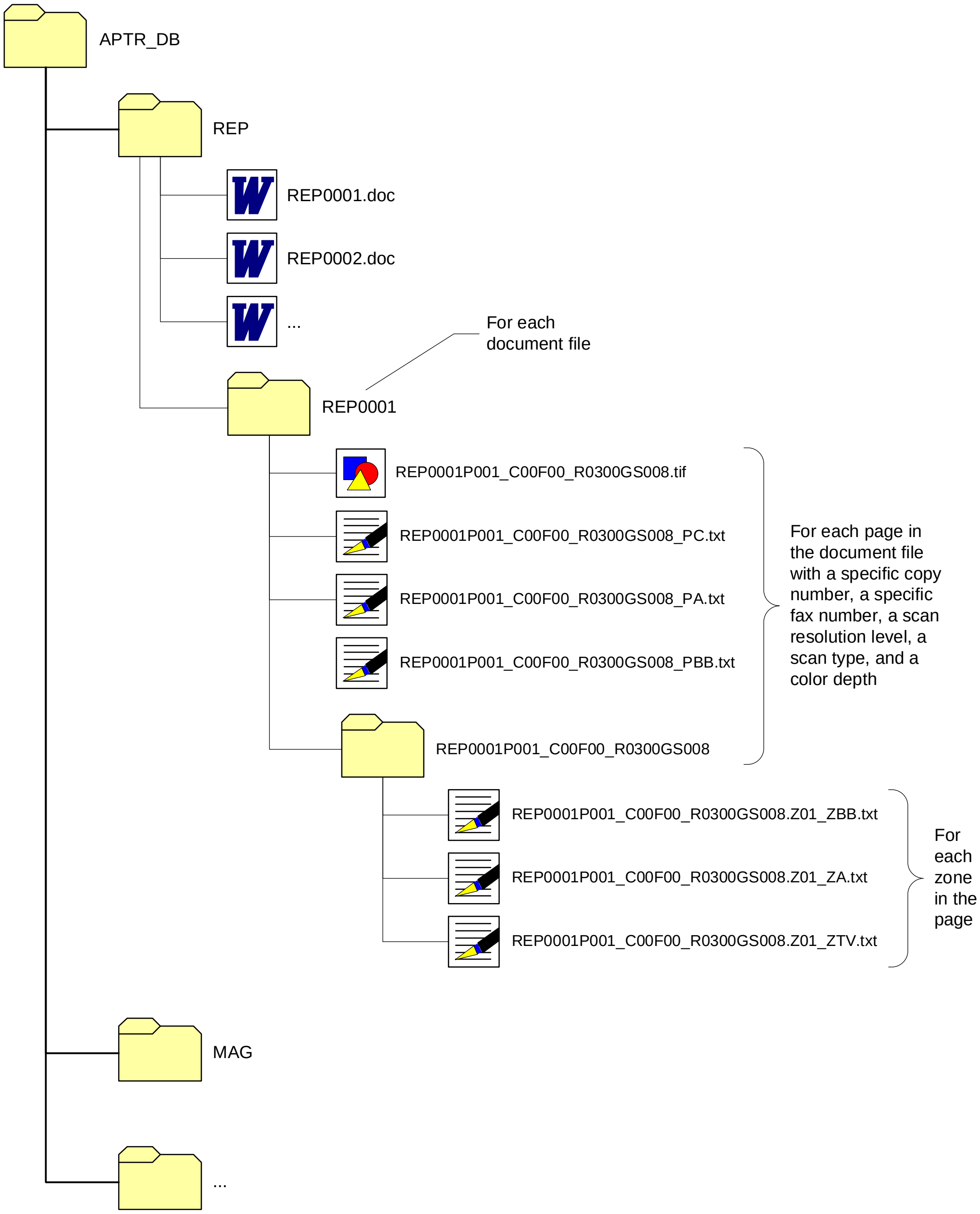

APTR_DB
REP
REP0001.doc
REP0002.doc
...
For each document file
REP0001
REP0001P001_C00F00_R0300GS008.tif
REP0001P001_C00F00_R0300GS008_PC.txt
REP0001P001_C00F00_R0300GS008_PA.txt
REP0001P001_C00F00_R0300GS008_PBB.txt
REP0001P001_C00F00_R0300GS008
For each page in the document file with a specific copy number, a specific fax number, a scan resolution level, a scan type, and a color depth
REP0001P001_C00F00_R0300GS008.Z01_ZBB.txt
REP0001P001_C00F00_R0300GS008.Z01_ZA.txt
REP0001P001_C00F00_R0300GS008.Z01_ZTV.txt
For each zone in the page
MAG
...

Figure 3: PATDB defined specification

## 5.4 STORAGE REQUIREMENTS

The total size of each category in the PATDB, in megabytes (MBs), is shown in Table 3.

TABLE 3: THE SIZE REQUIREMENTS IN MEGABYTES FOR EACH CATEGORY IN PATDB

| Category | Size (MBs) |
|---|---|
| Advertisements | 3050 |
| Book Chapters | 11226 |
| Letters | 67.4 |
| Magazines | 52110 |
| Newspapers | 1736 |
| Others | 0 |
| Reports | 7956 |
| Total | 76145.4 |

## 5.5 PATDB DEFINITION

The PATDB contains 6952 page images with their associated metadata files (records). The record files for a specific page includes three page-related records, namely, a page condition record, a page attribute record and a page bounding box record. In addition, the record files for a specific page includes three zone-related records, namely, a zone bounding box record, a zone attribute record and a zone ground-truth value record for all the indentified zones on the page. Page images can be binary (back & white), grayscale, or color format. Page images may be produced by scanning the document pages directly without faxing it first, or after faxing. Table 4 shows how many page images are available without faxing them first while Table 5 shows the number of page images after faxing them first.

TABLE 4: DISTRIBUTION OF DIRECTLY SCANNED PAGE IMAGE ACROSS PATDB

| Category | Color Format / Resolution | | | | | | | | | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| | Binary | | | Grayscale | | | Color | | | |
| | 200dpi | 300dpi | 600dpi | 200dpi | 300dpi | 600dpi | 200dpi | 300dpi | 600dpi | |
| Advertisements | 22 | 22 | 14 | 22 | 22 | 14 | 22 | 22 | 14 | 174 |
| Book Chapters | 111 | 111 | 111 | 111 | 111 | 111 | 0 | 0 | 0 | 666 |
| Letters | 5 | 5 | 0 | 5 | 5 | 0 | 0 | 0 | 0 | 20 |
| Magazines | 536 | 536 | 536 | 536 | 536 | 536 | 284 | 284 | 336 | 4120 |
| Newspapers | 35 | 35 | 35 | 35 | 35 | 35 | 0 | 0 | 0 | 210 |
| Others | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Reports | 161 | 161 | 161 | 161 | 161 | 161 | 0 | 0 | 0 | 966 |
| Total | 870 | 870 | 857 | 870 | 870 | 857 | 306 | 306 | 350 | 6156 |

TABLE 5: DISTRIBUTION OF FAXED THEN SCANNED PAGE IMAGE ACROSS PATDB

| Category | Color Format / Resolution | | | | | | | | | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| | Binary | | | Grayscale | | | Color | | | |
| | 200dpi | 300dpi | 600dpi | 200dpi | 300dpi | 600dpi | 200dpi | 300dpi | 600dpi | |
| Advertisements | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Book Chapters | 111 | 187 | 111 | 187 | 111 | 111 | 0 | 0 | 0 | 818 |
| Letters | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Magazines | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Newspapers | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Others | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Reports | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Total | 111 | 187 | 111 | 187 | 111 | 111 | 0 | 0 | 0 | 818 |

# CHAPTER 6

# PATDB SOFTWARE SYSTEM

## 6.1 INTRODUCTION

Along with the printed Arabic text database (PATDB), this work contains a software system. This software system works as a management system for the PATDB. The sole purpose of this software system is to make the job of those who would like to upload more documents to the database easier. It enables the users to upload more documents and manipulate the PATDB through an easy-to-use graphical user interface (GUI) forms. In addition, the software system provides the ATR researchers and developers with a number of basic image processing utilities that may be used in their field.

This chapter addresses the different aspects of the designed and implemented software system. Section 2 presents the features of the software system while Section 3 addresses its functions. The environment under which the software system is implemented is discussed in Section 4.

## 6.2 FEATURES

The software system of the PATDB is developed with two main features in mind: usability and extensibility. Following sub-sections address each one of these features alone.

### 6.2.1 USABILITY

System usability is governed by two main factors: the appearance of the user interface and how the system interacts with the user. Usability is characterized by five basic attributes: learnability, efficiency, memorability, error rate and satisfaction. Reference may be made to [Ferr01] for more information about each of the usability attributes. These attributes were in mind at the development cycle of the PATDB software system. However, they were not assessed at the end of the development. These attributes are achieved by the following practices:

- ***Intuitive:*** Making the system simple and self-explaining that will allow the user to carry out the required tasks in an easy and smooth manner. This minimizes the time required for training and becoming familiar with the system.

- ***Minimum Clicks:*** Minimizing the number of inputs and clicks required from the user. This in turn speeds up the time required for conducting the required task.

- ***Fully Featured:*** Making the function workspace fully featured and fully encapsulated (self-contained) that will in turn remove from the user the burden of switching between windows.

- ***Mouse-Keyboard:*** Taking in mind both mouse-oriented users and keyboard-oriented users. Any task that can be carried out by the mouse can be carried out using the keyboard.

- ***Same Look and Feel:*** Following the same look and fell throughout the system. This increases the ease of use, since the user will become familiar with how one function works (looks, act, etc.) and can translate his experience to other functions with the same look and feel.

- ***Auto:*** Automating some of the natural behaviors of users such as changing the system settings depending on the user selection. This will remove the burden of repeating things from the user each time he carry out this specific function.
- ***Flexible:*** Building the system to be controlled by the users and not vice versa. This is adhered to by providing the user with choices whenever appropriate.

### 6.2.2 EXTENSIBILITY

One main factor that governs the long-term use of any system, that continues to grow in scale and scope, is its extensibility feature. The extensibility here means two things: (1) the ability to extend the system without touching the existing source code; (2) the ability to add new functionalities or modify existing ones with minimum impact on the existing system functions [Krish98]. Building a system with extensibility in mind makes the software development process harder. One suggested solution to this issue is to follow well-defined criteria. In the PATDB software system, a set of self-generated criteria has been set up and followed throughout the software development cycle. This set includes the following points:

- ***Dictionary:*** PATDB depends heavily on records to store the information related to any page in the database. These records consist of a set of attributes. Each of these attributes has a name and a value. The dictionary is used to link each record attribute to a control on the system. The name of the control that is linked to a specific attribute depends on the record name that contains this specific attribute. On the other hand, the control type (text box, combo box, check box, etc.) depends on the attribute value type (text, number, logical, etc.). The linkage is done between the attribute name and the control name. For example, if we have an attribute called 'visible page rotation' in the 'page condition' record, then a check box control may be created with a name pc2. After that, an entry in the 'page condition' dictionary is created with the following information: pc2 and 'visible page rotation'.

  This dictionary principle is used to facilitate a single point of change that affects the rest of the software. Whenever the attribute name changes, only the entry where this attribute reside in the dictionary need to be changed. In addition, when a new attribute need to be added, we have only to create a control and name it after the record name. After that, we link the control name with the attribute name in the corresponding dictionary.

- ***Modularity:*** The software is implemented based on the modularity principle. Each module (class in object-oriented terminology) is devoted for implementing a specific function. Each module belongs to one of the following main categories:

  a. Forms Category

     This category represents the modules that have a direct interaction with the user.

  b. Modules Category

     The modules that provide the system with a major function are contained in the modules category. The modules in this category are mainly used by the modules in the forms category.

  c. Algorithms Category

     This category includes the modules that implement any algorithm used by the system.

d. Utilities Category

The modules in the utilities category provide assistant to the rest of the modules. They make the job of the caller module easier. In addition, they increase the code usability across modules.

e. Linkers' Category

The modules of the linkers' category are characterized by their simplicity. They are used to put-to-gather (encapsulate in object-oriented terminology) more than one entity / object into one. Therefore, the caller module will manage only one entity / object instead of multiple ones.

- ***Dynamic:*** The data populating the drop down controls in the system are stored separately from the system and loaded automatically at run time. This is done because the data is expected to change over time. In this way, the system will be able to adapt to the new requirements that target the drop down controls' data without modifying the source code.

- ***User Preferences:*** System settings are not hard coded. They are stored separately from the system in a parameters preferences rule.

In this way, the user will have the choice to modify these settings without accessing the system. The next time the user runs the system, his preferences will be implemented. In addition, the user can change the system settings within the system itself through GUI forms.

- ***Single Call:*** Whenever a module, caller, want to use a function from another module, calee, more than a specific number of times, the call to the function is executed through a single point, i.e., through a function in the caller. The purpose of this implementation will be clear by the following example. Assume that the caller calls a function in the callee 10 times. When we change the name of the function in the callee, we need to change it 10 times in the caller. Using the single point of call idea, we need to change it only once.

- ***Naming Conventions:*** Variables and functions throughout the system are named according to pre-defined criteria. Variables are named by the first letter of each words of the type of the variable followed by one or more words reflecting the variable purpose. Similarly, functions are named according to their intended tasks.

## 6.3 FUNCTIONS

The PATDB software system provides two broad sets of functions: database-related functions and ATR-related functions. Each set of these functions is discussed in the following sub-sections.

### 6.3.1 DATABASE-RELATED FUNCTIONS

PATDB software system offers a set of functions through which the user can browse and manipulate any database following the file naming conventions of PATDB specification shown in Chapter 5. This set of functions enables the user to:

- Traverse through the available document pages one by one with their associated description files (page condition record, page attribute record and page bounding box record) shown aside separately.
- Traverse through the available zones of any selected document page with the zones associated description files (zone bounding box

record, zone attribute record and zone ground-truth value record) presented aside separately.

- Modify any description file associated with any document page or zone through easy-to-use intuitive forms.
- Save the description files associated with any document page or zone as template and then load these saved templates to other document pages and zones.

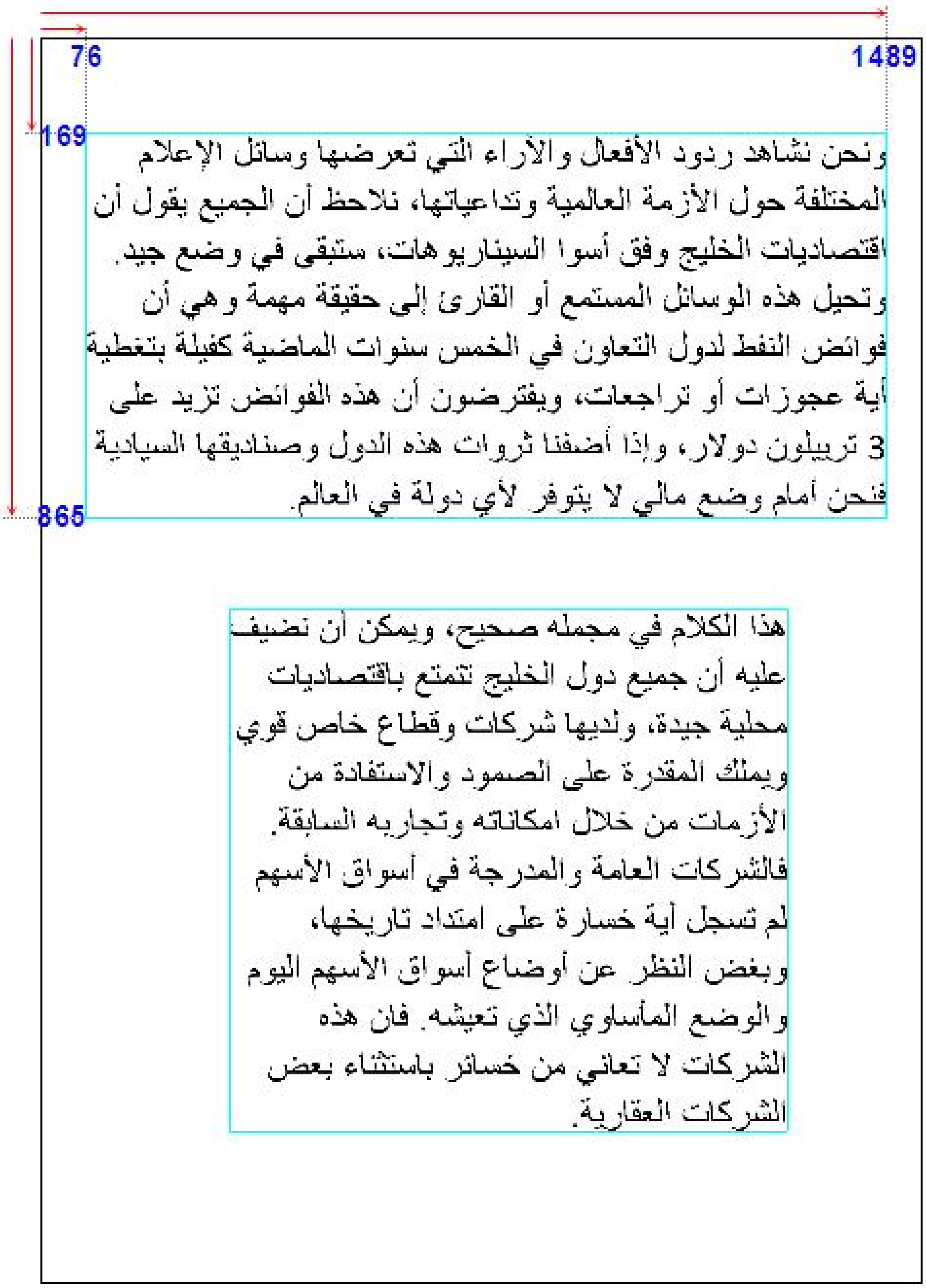
## 6.3.2 ATR-RELATED FUNCTIONS

The PATDB software system can assist the ATR systems by providing a set of ATR-related functions. These functions can be used in the early stages of the processing process of the ATR system in order to enhance its accuracy level. Following sub-sections explain each one of these functions.

### 6.3.2.1 BOUNDING BOX DETECTOR FUNCTION

The bounding box of a zone (image, figure, text, etc.) on a page is the rectangle that surrounds it ignoring the surrounding space. The bounding box is described by four numbers: the x-y coordinates of the upper-left corner and the x-y coordinates of the lower-right corner of the zone. The coordinates are measured, in pixels, from the top-left corner of the page. The user can modify the detected bounding box according to his preference.

The bounding box detector function provided by the PATDB software system is ideal in detecting the bounding boxes of zones on pages with single-column layout. Figure 4 shows the bounding box rectangles detected by the function of the two available text paragraphs on a sample single-column page. In addition, it shows the coordinates of the bounding box of the first paragraph.


76
1489
169
865

ونحن نشاهد ردود الأفعال والآراء التي تعرضها وسائل الإعلام
المختلفة حول الأزمة العالمية وتداعياتها، نلاحظ أن الجميع يقول أن
اقتصاديات الخليج وفق أسوا السيناريوهات، ستبقى في وضع جيد.
وتحيل هذه الوسائل المستمع أو القارئ إلى حقيقة مهمة وهي أن
فوائض النفط لدول التعاون في الخمس سنوات الماضية كفيلة بتغطية
أية عجوزات أو تراجعات، ويفترضون أن هذه الفوائض تزيد على
3 تريليون دولار، وإذا أضفنا ثروات هذه الدول وصناديقها السيادية
فنحن أمام وضع مالي لا يتوفر لأي دولة في العالم.

هذا الكلام في مجمله صحيح، ويمكن أن نضيف
عليه أن جميع دول الخليج تتمتع باقتصاديات
محلية جيدة، ولديها شركات وقطاع خاص قوي
ويملك المقدرة على الصمود والاستفادة من
الأزمات من خلال امكاناته وتجاربه السابقة.
فالشركات العامة والمدرجة في أسواق الأسهم
لم تسجل أية خسارة على امتداد تاريخها،
وبغض النظر عن أوضاع أسواق الأسهم اليوم
والوضع المأساوي الذي تعيشه. فان هذه
الشركات لا تعاني من خسائر باستثناء بعض
الشركات العقارية.

Figure 4: An illustration example of the bounding box rectangles for a sample document image

Algorithm

Detecting the bounding box rectangle of a zone (a text paragraph or a text line) on a single-column page involves the following steps:

1. ***Horizontal Projection:*** Project the page horizontally pixel by pixel. Then, store the result into an array, ***hArray***, of size equals to the page height in pixel.

2. ***Bounding Box Y-Coordinates***

   Find the bounding box y-coordinates of the zone by conducting the following two steps:

   a. ***Top Y-Coordinate***: Traverse forward through ***hArray*** to find the y-coordinate of the upper-left corner of the bounding box, $\boldsymbol{y^{top}}$. If the zone is the top most one in the page, then $\boldsymbol{y^{top}}$ is the zero-value index directly before the first non-zero-value index of ***hArray***. The other zones $\boldsymbol{y^{top}}$ is calculated in the same manner except that the first non-zero-value index must be preceded by at least ***n*** consecutive zero-value indices.

These ***n*** zero-value indices, which are determined by the user, represent the minimum vertical spaces (in pixel) that must separate any two consecutive zones. By having a small value for ***n***, the algorithm detects the non-intersecting text lines of the pages. Similarly, the bounding boxes for each paragraph in the page can be detected if we increase the value of ***n***. Likewise, the bounding box of the whole page content can be detected by setting n to a large number or the page height.

b. ***Bottom Y-Coordinate***: Traverse through the ***hArray*** again starting from $\boldsymbol{y^{top}}$ until a zero-value index is reached. This zero-value index represents the y-coordinate of the lower-right corner of the zone bounding box rectangle, $\boldsymbol{y^{bottom}}$.

### 3. Bounding Box X-Coordinates

Find the bounding box x-coordinates of the zone by carrying out the following three steps:

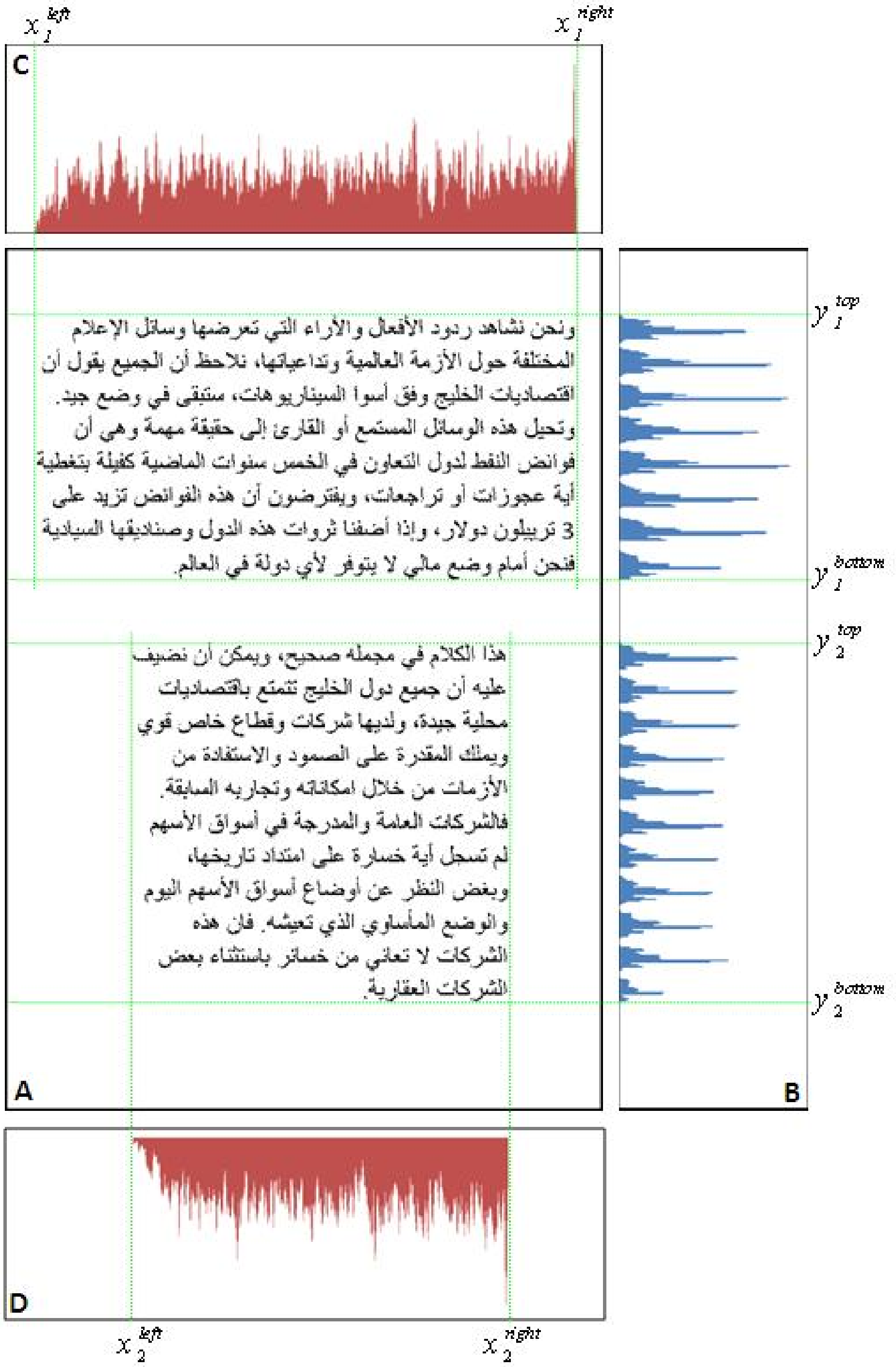
a. ***Vertical Projection***: Project the page vertically pixel by pixel starting from $\boldsymbol{y^{top}}$ to $\boldsymbol{y^{bottom}}$ and store the result into an array, ***vArray***.

b. ***Left X-Coordinate***: Traverse forward through ***vArray*** until a non-zero-value index is reached. The zero-value index directly preceding this non-zero-value index represents the x-coordinate of the upper-left corner of the bounding box rectangle, $\boldsymbol{x^{left}}$.

c. ***Right X-Coordinate***: Traverse backward through ***vArray*** until a non-zero-value index is reached. The zero-value index directly preceding this non-zero-value index is the x-coordinate of the lower-right corner of the bounding box rectangle, $\boldsymbol{x^{right}}$.

4. Repeat steps 2 and 3 until the end of the ***hArray*** is reached.

Figure 5 illustrates the steps carried out by the function to finds the bounding box rectangles of the two paragraphs in a sample page. The minimum vertical spaces, ***n***, were set to 30 in order to detect text paragraph (not text lines). Figure 5 shows a sample text page image (A), its horizontal projection (B) and the vertical projection of the two paragraphs in the page (C and D).



ونحن نشاهد ردود الأفعال والأراء التي تعرضها وسائل الإعلام
المختلفة حول الأزمة العالمية وتداعياتها، نلاحظ أن الجميع يقول أن
اقتصاديات الخليج وفق أسوا السيناريوهات، ستبقى في وضع جيد.
وتحيل هذه الوسائل المستمع أو القارئ إلى حقيقة مهمة وهي أن
فوائض النفط لدول التعاون في الخمس سنوات الماضية كفيلة بتغطية
أية عجوزات أو تراجعات، ويفترضون أن هذه الفوائض تزيد على
3 تريليون دولار، وإذا أضفنا ثروات هذه الدول وصناديقها السيادية
فنحن أمام وضع مالي لا يتوفر لأي دولة في العالم.

هذا الكلام في مجمله صحيح، ويمكن أن نضيف
عليه أن جميع دول الخليج تتمتع باقتصاديات
محلية جيدة، ولديها شركات وقطاع خاص قوي
ويملك المقدرة على الصمود والاستفادة من
الأزمات من خلال امكاناته وتجاربه السابقة.
فالشركات العامة والمدرجة في أسواق الأسهم
لم تسجل أية خسارة على امتداد تاريخها،
وبغض النظر عن أوضاع أسواق الأسهم اليوم
والوضع المأساوي الذي تعيشه. فان هذه
الشركات لا تعاني من خسائر باستثناء بعض
الشركات العقارية.

Figure 5: (A) A sample page with (B) its horizontal projection and (C, D) the vertical projection of its first and second paragraphs, respectively.

### 6.3.2.2 DE-NOISING FUNCTIONS

When a document page is scanned and then digitized, a certain amount of noise may appear. This noise incorporates some difficulties to the recognition process of the ATR systems. It may lower the recognition accuracy level of the system if the system was not developed to address this kind of noise. Different approaches can help in this regards. They can help in removing the incorporated noise from the scanned image before processing it by the ATR system. Following sections presents two algorithms for de-nosing.

#### 6.3.2.2.1 STATISTICAL BASED SMOOTHING FUNCTION

A statistical based smoothing function tackles mainly the noise pixels that add irregularities to the outer boundary of the characters. This function reduces the incorporated noise on the binary images by getting rid of small areas and filling little holes that make the character contour regular [Mahm94]. Filling and deletion depending on the pixel's initial value and its neighbors' initial values. The function can handle only black-&-while (binary) images.

Algorithm

The algorithm of the statistical based smoothing function bases on a statistical decision criterion. Given a binary image, the function modifies (fills or eliminates) each pixel depending on the pixel's initial value and its neighbors' initial values. The rules, taken from [Mahm94], are stated as follows:

If $P_0 = 0$ then

$$P_0' = \begin{cases} 0, & if \ \sum_{i=1}^{8} P_i < T \\ 1, & otherwise \end{cases}$$

else

$$P_0' = \begin{cases} 1, & \text{if } P_i + P_{i+1} = 2 \text{ for at least one } i = 1,\ldots,8 \\ 0, & \text{otherwise} \end{cases}$$

where $P_0$ is the current pixel value, $P_0'$ is the new pixel value and T is the threshold. The zero '0' in the above rules means white pixel while the one '1' means black pixel. The labeling scheme of these pixels is shown in Figure 6.

| | | |
|---|---|---|
| $P_4$ | $P_3$ | $P_2$ |
| $P_5$ | $P_0$ | $P_1$ |
| $P_6$ | $P_7$ | $P_8$ |

Figure 6: The current pixel $P_0$ and its neighbors.

### 6.3.2.2.2 AVERAGE SMOOTHING FUNCTION

The average smoothing function attempts to smooth the image edges and corners by filling small holes or deleting small fills. The filling and deletion is determined by a 3×3 weighted matrix elements. The system gives defaults parameters. However, the user can modify the parameters. The function can handle black-&-white (binary) and grayscale images.

Algorithm

Given a 3×3 weighted matrix elements of a total sum equals to one (see Figure 7-C), the steps of the average smoothing function are as follows:

1. Scan the image pixel by pixel with a window size of 3×3 as shown in Figure 7-B.

2. Multiply each pixel in the 3×3 image window by its corresponding element in the 3×3 weighted matrix elements. Then, sum all of the multiplications into variables called ***total***. Figure 7-D shows the result of multiplying a sample 3×3 image window by the weighted matrix elements.

3. Round the ***total*** variable into its nearest integer value as shown in Figure 7-E.

4. Change the value of the pixel at the center of the 3×3 image window to the new pixel that has its value equals to the value of the ***total*** variable.

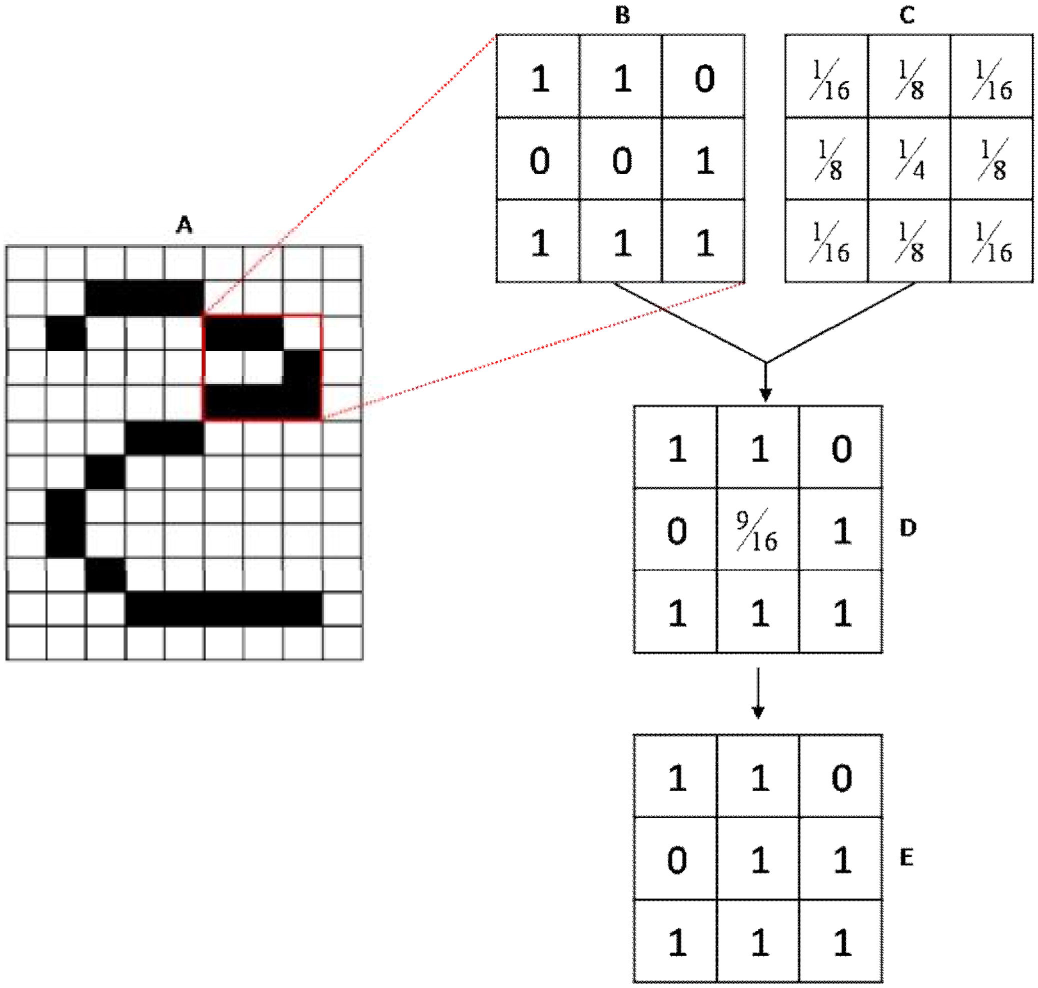


Figure 7: A sample iteration of the average smoothing function on a sample black-&-white page image (A). (B) is the initial 3×3 image window. (C) is the 3×3 weighted matrix elements. (D) is (B) after applying Step 2. (E) is (D) after applying Step 3 then Step 4.

### 6.3.2.3 PAGE DE-SKEW FUNCTION

When a document page is copied or scanned, a few degree of skew may be introduced. In order to achieve good recognition results, this skew need to be corrected before passing the page image into the ATR system. The aim of this function is to enable the user to correct such skew through few mouse-clicks.

Algorithm

Given a page image that contained skewed text, the function de-skews the text by carrying out the following steps:

1. The user is asked to draw a line that simulates the base line of the dominant text on the given page.

2. The function calculates the slope absolute value of the drawn line and finds the slope angle, ***angle***, in degree.

3. The function rotates the page ***angle*** degrees

## 6.4 DEVELOPMENT ENVIRONMENT

The PATDB software system is developed on Microsoft® Windows® XP platform using Microsoft® .NET Framework 1.1 C# programming language. The system is best viewed on a screen resolution of 1280×800 ppi (pixel per inch).

# CHAPTER 7

# CONCLUSIONS AND FUTURE WORK

## 7.1 INTRODUCTION

In this chapter, our conclusions are stated in Section 2, contributions to the literature are mentioned in Section 3 and suggestions for future work are indicated in Section 4.

## 7.2 CONCLUSIONS

Through the course of this work, the most popular Arabic printed (and handwritten) text databases available in the literature are survived. The survey indicates that none of the available Arabic printed text databases addresses the lack of a benchmark printed Arabic database that is public, large-scale and comprehensive.

The other side of the survey yields in a set of attributes. These attributes represents the most desirable attributes, by researchers working in text recognition area, in any future database aiming to serve the area of (printed) Arabic text recognition research. Based on the attributes coming from the survey and a set of self-generated attributes, a public large-scale comprehensive database of printed Arabic text is developed. The development is done through a systematic process that ensures the enforcement of the set-up (survived and self-generated) attributes.

The developed printed Arabic database includes different types of documents coming from different sources of real life written communications. The document pages are scanned with different color formats and different resolution levels. Some of the document pages, that are included in the database, simulate the real life faxing process. These document pages were faxed from one fax to another before the scanning process takes place.

In addition to the developed printed Arabic database, this work includes a software system. This software system mainly targets the users who would like to manipulate and add more documents to the database. The software system makes the job of those users easier with an easy-to-use graphical user interface (GUI) forms. In addition to these manipulation functions, the software system provides a set of basic functions related to the automatic text recognition (ATR) process. The ATR-related functions may help researchers in their work.

## 7.3 CONTRIBUTIONS

This work will contribute to the Automatic Arabic Text Recognition (AATR) field by constructing a database of printed Arabic text that is comprehensive, standard and public. The database will be made freely available to interested researchers in AATR field. The database is expected to address one of the limitations of AATR researches, i.e., the lack of a public large-scale comprehensive printed Arabic text database (PATDB) that is freely available to AATR researchers. It is expected, with time, that the database will be used as a benchmark database to compare the research results and the different techniques and researches of researchers.

## 7.4 FUTURE WORK

A future work for both the developed printed Arabic database and the developed software system can be pointed out. A deep analysis of the most popular databases of other than Arabic language can conducted. Through this analysis, the structure of the analyzed databases can be known and hence the most popular structure can be implemented on our developed database. This action makes our database combatable with the systems that are currently using databases following the same structure enforced on our database. In addition to this analysis, the database can be enhanced by adding more documents to it.

The developed software system can be enhanced by a number of actions. First, more functions related to the ATR process can be provided in addition to enhancing the existing ones. The bounding box detector function can be enhanced to handle page images having more than one column. In addition, the average smoothing function can be enhanced to hand colored page images not only the binary and grayscale ones. Second, the provided functions can be modified to incorporate batch processing.

# APPENDIX A

# PATDB METADATA

## A.1 INTRODUCTION

When training or testing an automatic text recognition (ATR) technique, sample page images are needed. The number of these sample page images varies depending on the used technique. These sample page images are not enough to support the training or testing process. Files fully describing these page images are definitely required. This appendix presents the different files that will be included with each page image available in the printed Arabic text database (PATDB).

The database will include metadata about each page (visual conditions, type, etc.) and about each zone in the page (position, type, content, etc.) as summarized in Figure 8 in addition to the scanned document pages. The page-related and zone-related metadata information, adopted from [Phil93b] , is addressed in Section 2 and Section 3, respectively. The structure according to which page-related and zone-related metadata information is stored is discussed in Section 4.

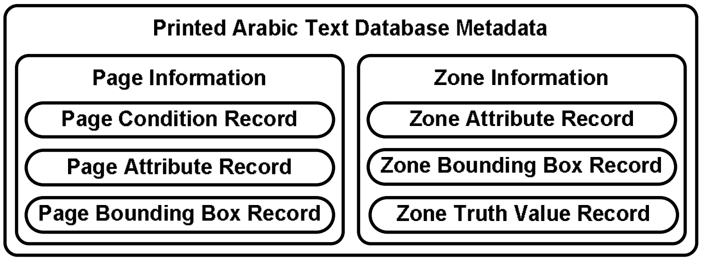


Figure 8: PATDB metadata records

## A.2 PAGE-RELATED METADATA INFORMATION

The information related to any page that is stored in the database is saved in three different records. These records are the page condition record, the page attribute record and the page bounding box record. The attributes of each of these records are addressed in the following sub-sections.

The reasons behind having three different records, not only one, to store the page-related metadata information can be summarized in three points: specialization, simplicity and usability. Specialization means that each record contains the set of attributes that reflect its name. Hence, the record will be simple to read and understand. For example, the page condition record has only the attributes that describe the visual conditions of the given page. Usability, here, means that a record of a specific page image can be duplicated to a similar page image with minor or no changes. For example, if a document page is scanned twice with different resolution each time. Then, exact copy of the page condition record and page attribute record of the first scan can be duplicated to the second scan with few simple modifications. These simple modifications will be clear after coming across the page-related metadata information in details in the next section.

### A.2.1 PAGE CONDITION RECORD

The page condition record includes attributes that describe the visual conditions or qualities of a given document page as shown in Table 6. The record can indicate whether a manual degradation model had been applied on the page or not. For example, if the page is copied 5 times (5th generation), the '***n-the copy***' attribute is set to '5'. Similarly, if the page has salt/pepper noise, the '***visible salt/pepper noises***' attribute is set to 'yes'.

TABLE 6: ATTRIBUTES OF PAGE CONDITION RECORD

| No. | Attribute | Possible Values |
|---|---|---|
| 01 | Document ID | |

| | | |
|---|---|---|
| 02 | n-th copy | 0, 1, 2, … |
| 03 | n-th fax | 0, 1, 2, … |
| 04 | Resolution level | 200 (fax), 300, 600, … |
| 05 | Scanning type | black & white, grayscale, color |
| 06 | Color depth (in bits) | 1, 2, 4, 8, … |
| 07 | Degradation type | original, page aging, … |
| 08 | Visible salt / pepper noises | yes, no |
| 09 | Visible vertical streaks | yes, no |
| 10 | Visible horizontal streaks | yes, no |
| 11 | Extraneous symbols on the top | yes, no |
| 12 | Extraneous symbols on the bottom | yes, no |
| 13 | Extraneous symbols on the left | yes, no |
| 14 | Extraneous symbols on the right | yes, no |
| 15 | Page skewed on the left | yes, no |
| 16 | Page skewed on the right | yes, no |
| 17 | Page smeared on the left | yes, no |
| 18 | Page smeared on the right | yes, no |
| 19 | Visible page rotation | yes, no |
| 20 | Page rotation angle (in degree) | |
| 21 | Page rotation angle standard deviation | |

## A.2.2 PAGE ATTRIBUTE RECORD

The page attribute record contains a set of attributes that describe the various properties/attributes of a given document page. Table 7 presents the various attributes used in the page attribute record.

The value of the '***document language***' attribute is 'Arabic' for this database. This attribute is provided for upward compatibility with future databases that may be produced in other languages later.

The publication information from which the document page came is held by the '***publication information***' attribute. The information may include the name, volume number, issue number, publishing date and the corresponding page number of the document page of the publication.

The '***multiple pages from the same article***' attribute indicates whether multiple pages from the same article are included in the database. If multiple pages within the same article exist, they can be retrieved by reference to the publication name, volume and issue number of the page.

If the page contains symbols other than the standard ASCII symbols, the '***special symbol present in text zone***' attribute is set to 'yes'.

The '***max number of text columns***' includes a numeric value that corresponds to the number of equal-width text columns within the live matter area of the document page.

The character orientation within the text line when the page is oriented to up-right position is indicated by the '***character orientation***' attribute. The character orientation may be the same as the page orientation ('up-right'), rotated to the right ('rotated-right'), or rotated to the left ('rotated-left'). Similarly, the '***text reading direction***' gives the text reading direction within a text line when a page is oriented to up-right position.

TABLE 7: ATTRIBUTES OF PAGE ATTRIBUTE RECORD

| No. | Attribute | Possible Values |
|---|---|---|
| 01 | Document ID | |
| 02 | Document language | Arabic, Farsi, Dari, Azeri, Urdu, Uygur, Tajik, Pashto, Kurdish, English |
| 03 | Document script | Arabic, Latin |
| 04 | Document type | newspaper, book, report, magazine |
| 05 | Publication information | |
| 06 | Multiple pages from the same article | yes, no |
| 07 | Text zone present | yes, no |
| 08 | Special symbol present in text zone | yes, no |
| 09 | Displayed math zone present | yes, no |
| 10 | Table zone present | yes, no |
| 11 | Half-tone zone present | yes, no |
| 12 | Drawing zone present | yes, no |
| 13 | Page header present | yes, no |
| 14 | Page footer present | yes, no |
| 15 | Max number of text columns | |
| 16 | Page column layout | regular, combined-columns |
| 17 | Character orientation | up-left, up-right, rotated-right, rotated-left |

| | | |
|---|---|---|
| 18 | Text reading direction | left-right, right-left, top-down, bottom-up |
| 19 | Dominant font type | Traditional Arabic, Arial, Simplified Arabic, Arabic Transparent, Times New Roman, Andalus, Courier New, Microsoft Sans Serif, Tahoma |
| 20 | Dominant character spacing | proportional, fixed |
| 21 | Dominant font size (pts) | << 9, 9-12, 13-18, 19-24, 25-36, >> 36 |
| 22 | Dominant font style | plain, bold, italic, underline, other |

### A.2.3 PAGE BOUNDING BOX RECORD

The boundary of the header area, body/live matter area and footer area of a given document page is defined in the page bounding box record. For each area, the coordinates of the upper-left and lower-right corners are included in the record. Table 8 shows these six coordinates of the three areas of a given document page.

TABLE 8: ATTRIBUTES OF PAGE BOUNDING BOX RECORD

| No. | Attribute | Possible Values |
|---|---|---|
| 01 | Document ID | |
| 02 | Header area upper-left corner coordinates | (X,Y) |
| 03 | Header area lower-right corner coordinates | (X,Y) |

| | | |
|---|---|---|
| 04 | Live matter area upper-left corner coordinates | (X,Y) |
| 05 | Live matter area lower-right corner coordinates | (X,Y) |
| 06 | Footer area upper-left corner coordinates | (X,Y) |
| 07 | Footer area lower-right corner coordinates | (X,Y) |

## A.3. ZONE-RELATED METADATA INFORMATION

Each identified zone in any given document page is described using three records: a zone bounding box record, a zone attribute record, a zone truth-value record. The reasons behind having three records, not only one, for each identified zone are the same reasons behind having multiple records for each page image. The previous section explains these reasons in details. The attributes related to each of the zone-related records are presented in the following sub-sections.

### A.3.1 ZONE BOUNDING BOX RECORD

The zone bounding box record of a given zone in a given document page holds 2 main properties about the zone: its identification number and its boundary coordinates within the document page. The zone bounding box attributes are shown in Table 9.

TABLE 9: ATTRIBUTES OF ZONE BOUNDING BOX RECORD

| No. | Attribute | Possible Values |
|---|---|---|
| 01 | Document ID | |
| 02 | Zone ID | |

| | | |
|---|---|---|
| 03 | Zone upper-left corner coordinates | (X,Y) |
| 04 | Zone lower-right corner coordinates | (X,Y) |

## A.3.2 ZONE ATTRIBUTE RECORD

The zone attribute record describes the common characteristics of an identified zone in a given document page. Table 10 presents these set of attributes.

The dominant font type, character spacing, font size and font style within the zone are defined by the '***dominant font type***', '***dominant character spacing***', '***dominant font size***' and '***dominant font style***' attributes, respectively.

The '***zone's column number***' attribute describes the zone's column location. A zone may be in the header area, footer area and column 1 of 1, 1 of 2, etc.

The zones of each document page can be grouped into several logical units. Within each logical unit, the reading order is sequential. This logical unit is called a semantic thread. So, the '***next zone ID within the same thread***' attribute is used to indicate the reading order among the zones that constitute a semantic thread. 'nil' is used to indicate the end of the semantic thread.

TABLE 10: ATTRIBUTES OF ZONE ATTRIBUTE RECORD

| No. | Attribute | Possible Values |
|---|---|---|
| 01 | Document ID | |
| 02 | Zone ID | |
| 03 | Zone content | text, text with special symbols, displayed math, table, half-tone, drawing, form, ruling, bounding box, logo, map, advertisement, announcement, handwriting, others |

| | | |
|---|---|---|
| 04 | Text zone label | text body, list item, drop cap, caption, abstract body, abstract heading, section heading, synopsis, highlight, pseudo-codes, reference heading, reference list item, footnote, author biography, page header, page footer, page number, article title, author, affiliation, diploma information, society membership information, article submission information, abstract heading, abstract body, footnote heading, keyword heading, keyword body, other |
| 05 | Text alignment within the zone | left aligned, center aligned, right aligned, justified, justified hanging, left hanging |
| 06 | Dominant font type | Traditional Arabic, Arial, Simplified Arabic, Arabic Transparent, Times New Roman, Andalus, Courier New, Microsoft Sans Serif, Tahoma |
| 07 | Dominant character spacing | proportional, fixed |
| 08 | Dominant font size (pts) | << 9, 9-12, 13-18, 19-24, 25-36, >> 36 |
| 09 | Dominant font style | plain, bold, italic, underline, other |
| 10 | Character orientation | up-left, up-right, rotated-right, rotated-left |
| 11 | Text reading direction | left-right, right-left, top-down, bottom-up |
| 12 | Zone's column number | |
| 13 | Next zone ID within the same thread | |

### A.3.3 ZONE TRUTH-VALUE RECORD

The zone truth-value record is only available in the case of text zones. It holds the ground-truth value of the given text zone.

## A.4 RECORDS' STRUCTURE

The page-related and zone-related metadata information is normally represented by unstructured files. Although this representation is easier for the developer to generate, it is very difficult for users (and even for the developer after a period) to understand. One alternative solution is to use a fully structured representation, such as XML (eXtensible Markup Language), to represent the metadata information. However, this fully structured representation may be complex and hence difficult to understand by naïve users. What actually needed is a simple and easy to understand representation and at the same time structured or semi-structured. One of the simple, easy to understand and structured representation is JavaScript Object Notation (JSON).

Although we can represent the page-related and zone-related metadata information using an unstructured representation, as most databases do, we choose JSON a standard data representation for the PATDB for two reasons. First, JSON representation makes the files of the metadata information more readable since each attribute has its name and value stored in the file (not only the attribute value). Second, the metadata information files can be manipulated automatically by machines. Appendix B gives summary of JSON and compares it with XML via an example.

JSON will be used as a data representation for all the page-related and zone-related records' files except the zone truth-value record. The zone ground-truth value is stored directly, without any data representation, to the respective zone truth-value record's file.

JSON will not be used as is when representing the records' files; a simple rule must be followed. This rule states that no more that a single JSON object can exists in a record's file. The rule also states that each name/value pair must exist on a separate line. The object curly brackets must also be on separate lines. Figure 9 shows the content of the page bounding box record's file of a sample page image using JSON representation.

```
{
        "Document ID": "REP0001PG001_C00F00_R0300GS008",
        "Header area upper-left corner coordinates": [272,67],
        "Header area lower-right corner coordinates": [1297,100],
        "Live matter area upper-left corner coordinates": [213,167],
        "Live matter area lower-right corner coordinates": [1363,2079],
        "Footer area upper-left corner coordinates": [208,2170],
        "Footer area lower-right corner coordinates": [1362,2200]
}
```

Figure 9: A sample page bounding box records' file

# APPENDIX B

# JAVASCRIPT OBJECT NOTATION (JSON) IN GLANCE

## B.1 INTRODUCTION

JSON (JavaScript Object Notation) is a lightweight, text-based, language-independent data interchange format. It was derived from the European Computer Manufacturers Association (ECMA) Script Programming Language Standard. JSON defines a small set of formatting rules for the portable representation of structured data. It was submitted to the Internet Engineering Task Force (IETF) by JSON.org in July 2006 under RFC (Request for Comments) 4627.

JSON can represent 6 types: 4 primitive types (string, number, boolean and null) and 2 structured types (object and array). The object type is enclosed between 2 curly brackets {…} and consists of an unordered collection of zero or more name/value pairs (members) separated by comma (,). The name is of string type while the value is one of the six types mentioned above. The name is separated from the value by a colon (:). The array type is enclosed between 2 square brackets [...] and consists of an ordered sequence of two or more values separated by comma.

## B.2 JSON VS XML: AN EXAMPLE

Suppose we have two document pages with the following information:

| | Document Page 1 | Document Page 2 |
|---|---|---|
| Title | Arabic ATR Field | English ATR Field |
| Author | Dr. X | ***unknown*** |
| Year | 2008 | 1990 |
| Is Printed | No | Yes |

The information of these two document pages can be represented based on JSON using one JSON array that contains two JSON objects. Each JSON object contains four name/values pairs (members) as follows:

```
[
        {
                “Title”: “Arabic ATR Field”,
                “Author”: “Dr. X”,
                “Year”: 2008,
                “Is Printed”: false
        },
        {
                “Title”: “English ATR Field”,
                “Author”: null,
                “Year”: 1990,
                “Is Printed”: true
        }
]
```

On the other hand, the two document pages information can be represented using XML after probably defining an appropriate DTD (Data Type

Definition) or XSD (XML Schema Definition) file in order for the XML file to be checked against it. The XML file may look as follows:

```
<pages>
        <page>
                <title>Arabic ATR Field</title>
                <author>Dr. X</author>
                <year>2008</year>
                <isPrinted>no</isPrinted>
        </page>
        <page>
                <title>English ATR Field</title>
                <author></author>
                <year>1990</year>
                <isPrinted>yes</isPrinted>
        </page>
</pages>
```

From the above example, we notice that JSON is more natural than XML. JSON is simpler and cleaner than XML. There are many freely available converters from JSON and XML and vice versa [JSON-b].

# APPENDIX C

# PATDB SOFTWARE SYSTEM USER MANUAL

## C.1 INTRODUCTION

This appendix guides the user of the printed Arabic text database (PATDB) software system through the installation steps of the PATDB software system (Section 2). In addition, this appendix takes the user in a guided tour (Section 3). Through this guided tour; the user is familiarized with the functions of the PATDB software system.

## C.2 INSTALLATION

This section discusses the system requirements and installation of the PATDB software system.

### C.2.1 SYSTEM REQUIREMENTS

The PATDB software system is not tested on a variety of machines in order to determine the minimum requirements of systems configuration. However, it was experiencing slow image rendering on a machine with 512 MB (megabyte) RAM. This obstacle totally disappeared when an extra 2 GB (gigabyte) of RAM was installed.

Following is the latest system configuration on which PATDB software system is developed:

- Intel® Core™ Duo T2500 @ 2.00 GHz processor
- 2.50 GB RAM

- Microsoft® Windows® XP operating system
- Microsoft® .NET Framework 1.1
- 1280×800 ppi screen resolution

To be able to run PATDB software system, the Microsoft® .NET Framework 1.1 (or higher) must be installed on a machine having operating system compatible with the installed version of the Microsoft® .NET Framework.

### C.2.2 INSTALLING PATDB SOFTWARE SYSTEM

To install the PATDB software system, carry out the following steps in sequence:

1. Double-click the file entailed 'PATDB.exe'
2. Select 'Destination folder'. Let us say 'Desktop'.
3. Click 'Install'. A directory named 'PATDB' in the destination folder will be created. In our case, the 'PATDB' directory will be created in the 'Desktop'. This directory will contain two directories, 'PATDB Software System' and 'Sample DB'. The 'PATDB Software System' directory, as its name indicates, contains the software system through which any database following the PATDB specifications can be manipulated. The 'Sample DB' directory includes a sample document pages that will be used throughout this user manual.

## C.3 GUIDED TOUR

PATDB software system has two main sets of functions: database -related functions and automatic text recognition (ATR)-related functions. All the major functions of these two sets are discussed in the following sub-sections.

## C.3.1 STARTING PATDB SOFTWARE SYSTEM UP

To start PATDB software system, simply double-clicks the system icon (Figure 10) available in the ‘installation folder\PATDB Software System’. The application window is then displayed (Figure 11).

Figure 10: PATDB software system icon

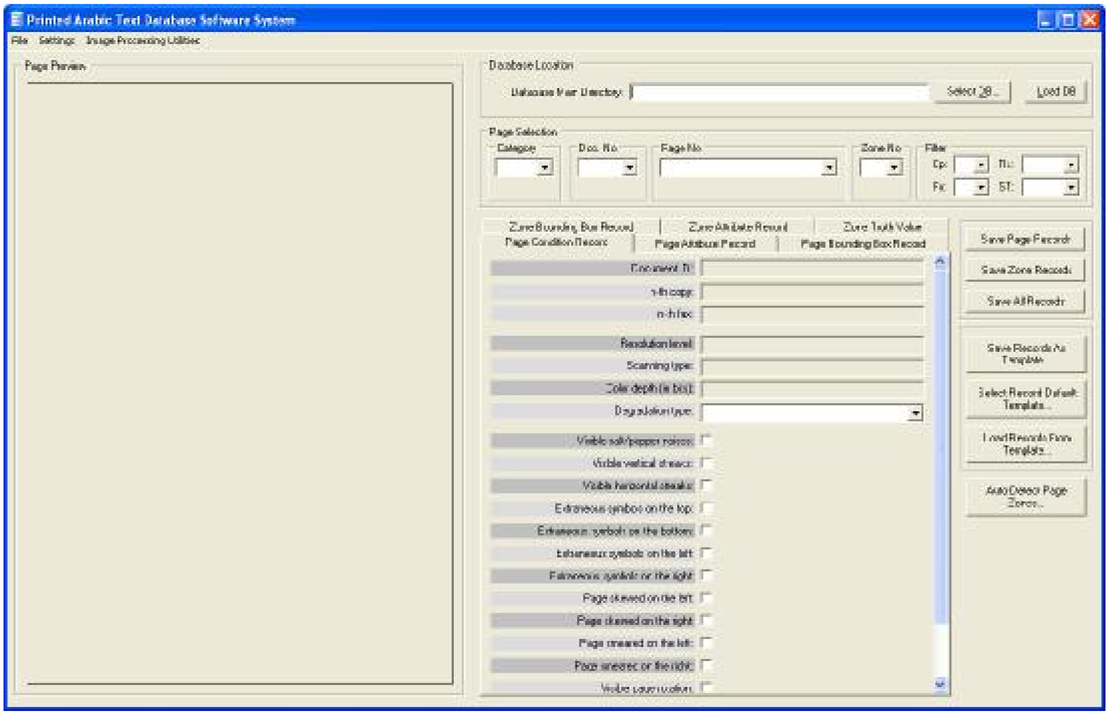


Figure 11: PATDB software system application window

## C.3.2 DISCOVERING PATDB SOFTWARE SYSTEM MAIN INTERFACE

The application window (Figure 12) of PATDB software system contains the following main areas:

- Command Menus area (Area 1)
  Most of the jobs carried out by the system can be launched through the command menus area.

- Database Location area (Area 2)
  This area enables you to select the main directory of the database that you need to be load and manipulate using the PATDB software system.

- Selected Page area (Area 3)
  Through this area, you will be able to traverse through the available categories, documents, pages and zones that are available in the selected database.

- Image Preview area (Area 4)
  The image preview area is the location where the image of the page you select is displayed.

- Page-Related and Zone-Related Records Display area (Area 5)
  This area displays the data related to any page or zone you select.

- Quick-Access Buttons area (Area 6)
  The quick access buttons provides you will the frequently used commands that are concerned with the page or zone you select.

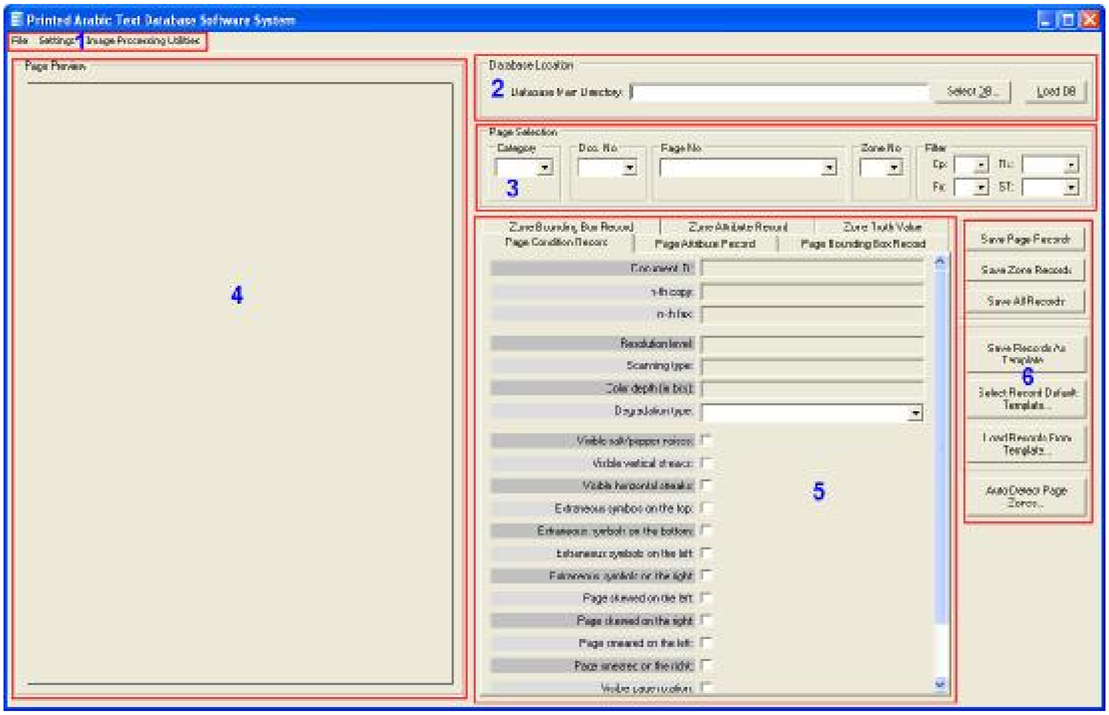
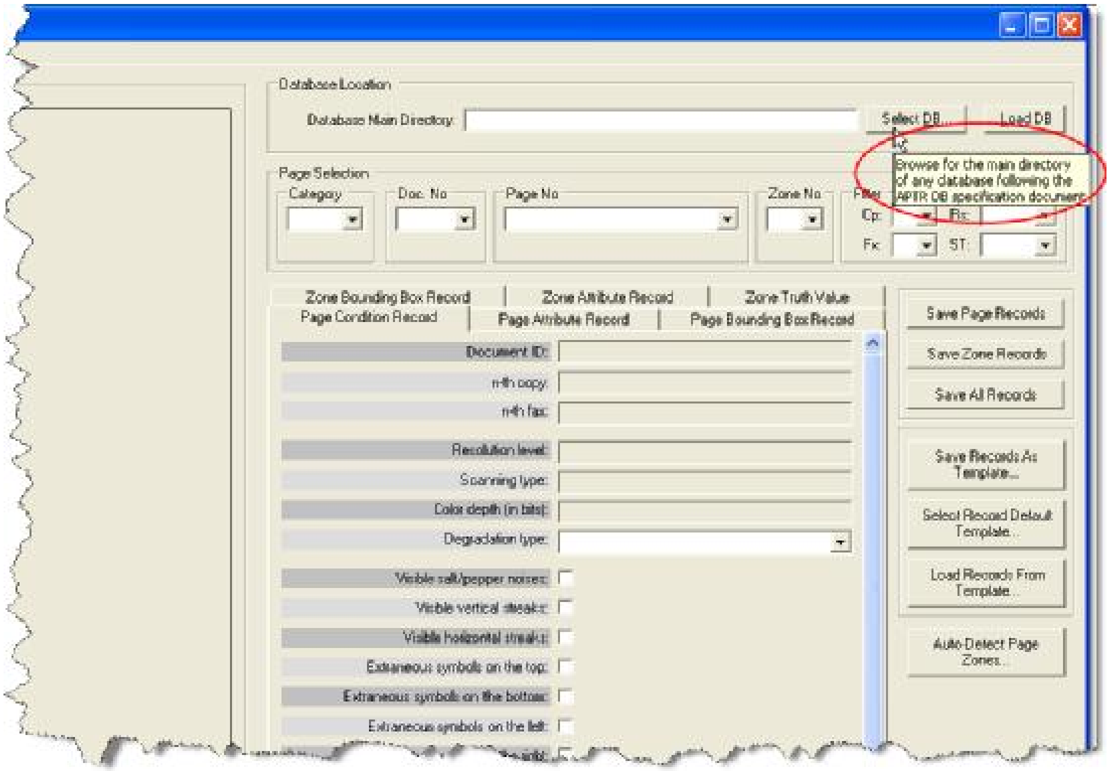


Figure 12: Application window with main areas illustration

The function of some of the buttons in the above areas can be learned by holding your pointer over it for a while. A tooltip will tell you what the button does (see Figure 13).



Browse for the main directory of any database following the APTR DB specification document

Figure 13: Application window with tooltip illustration

## C.3.3 GETTING STARTED WITH A FIRST TUTORIAL

The best way to get familiar with the functions of PATDB software system is undoubtedly by using it. A number of sample page images are provided with the software; they allow you to get started even when you do not have images to start with. Let us turn to these now one by one.

### C.3.3.1 DATABASE-RELATED FUNCTIONS

#### C.3.3.1.1 Loading a Database

To manipulate a database following PATDB specification (refer to Chapter 5 for details), you need to:

1. Show ‘Select DB Main Directory’ dialog by carrying out one of the following methods:

    a. Clicking ‘Select DB…’ button in the ‘Database Location’ area

    b. Pressing ‘Alt-D’

    c. Selecting ‘File’ ª ‘Select Database Main Directory…’

    d. Pressing ‘Alt-F-D’.

2. From the dialog, browse to the installation directory and select ‘Sample DB’, then click ‘OK’ to confirm the selection.

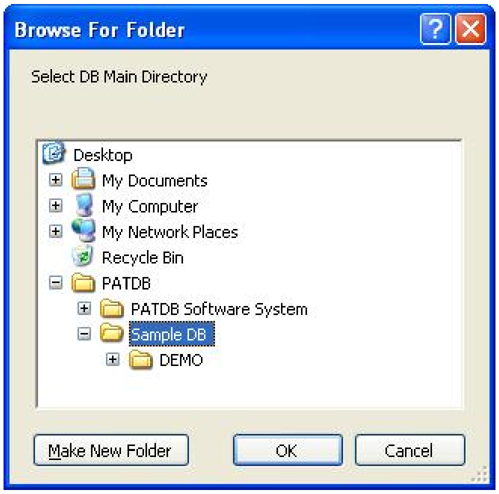


Figure 14: Database main directory selector dialog box

3. Load the selected database to PATDB software system by one of the following methods:

    a. Clicking ‘Load DB’ button

    b. Pressing ‘Alt-L’

    c. Selecting ‘File’ ª ‘Load Selected Database’

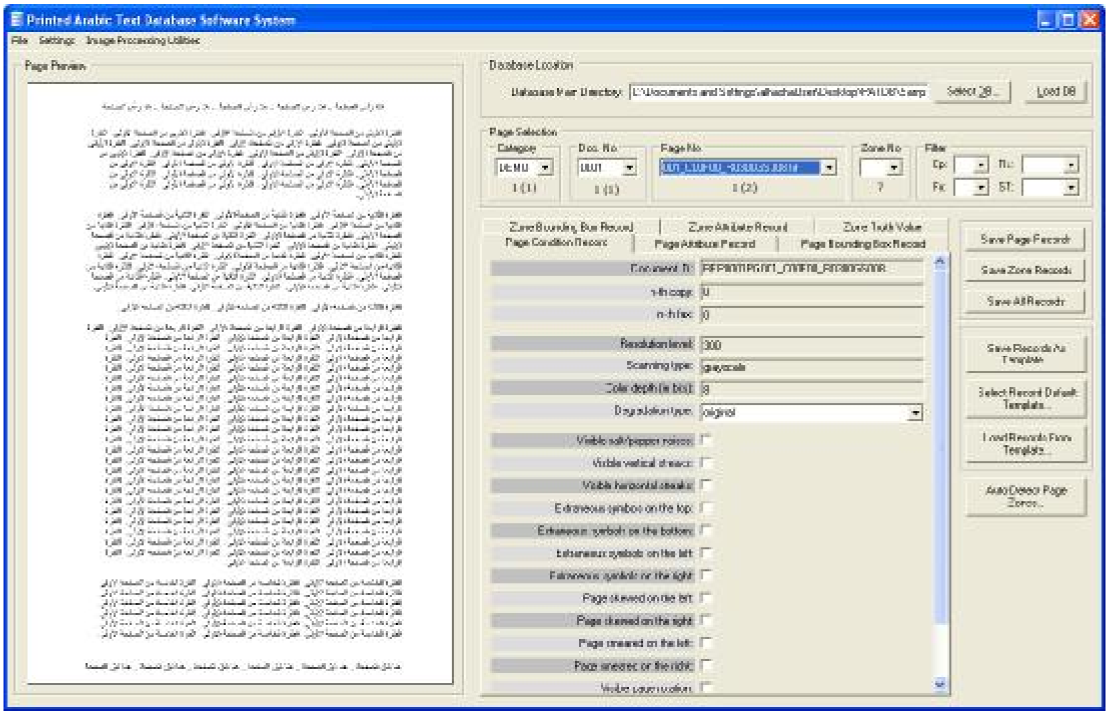
d. Pressing ‘Alt-F-L’

Now, you can manipulate the selected database using the PATDB software system.

#### C.3.3.1.2 Presenting a Page Image and its Associated Records

To display the image of a page and its associated records, you need first to select the category to which it belongs and the document in which it appears. You can do this through the dropdown menus (combo boxes) in the ‘Selected Page’ area.

Let us load a demo page. Select ‘DEMO’ from the category dropdown menu, ‘0001’ from the document number dropdown menu and ‘001_C00F00_R0300GS008.tif’ from the page number dropdown menu. Directly after you select the page number, the system will display the image of the page you select in the preview area and populate the 3 page-related tabs (‘Page Condition Record’ tab, ‘Page Attribute Record’ tab and ‘Page Bounding Box Record’ tab) in the ‘Page-Related and Zone-Related Records Display’ area (Figure 15).



Database Main Directory: C:\Documents and Settings\alhashai\User\Desktop\PATDB\Samp

Category: DEMO 1 (1)

Doc. No: 0001 1 (1)

Page No: 001_C00F00_R0300G008.tif 1 (2)

Zone No: ?

Document ID: RFP0001PG001_C00F00_R0300G008

n-th copy: 0

n-th fax: 0

Resolution level: 300

Scanning type: grayscale

Color depth (in bits): 8

Degradation type: original

Visible salt/pepper noises: □

Visible vertical streaks: □

Visible horizontal streaks: □

Extraneous symbols on the top: □

Extraneous symbols on the bottom: □

Extraneous symbols on the left: □

Extraneous symbols on the right: □

Page skewed on the left: □

Page skewed on the right: □

Page smeared on the left: □

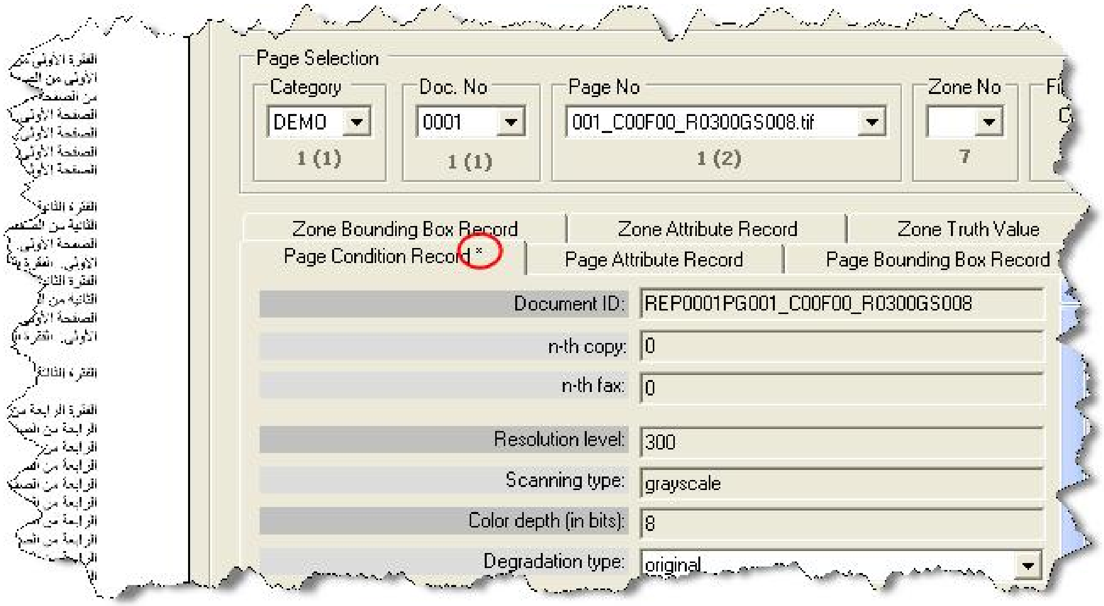
Page smeared on the right: □

Visible page rotation: □

Figure 15: Application window after selecting a sample page image

C.3.3.1.3 Modifying and Saving Page Associated Records

To modify any record associated with the page you select, go to the record corresponding tab in the 'Page-Related and Zone-Related Records Display' area and change the necessary values from there. When you change a value of a record, an asterisk (*) after the corresponding record tab name will be shown. This asterisk indicates that the record is modified but not saved (Figure 16).



Category: DEMO 1 (1)
Doc. No: 0001 1 (1)
Page No: 001_C00F00_R0300GS008.tif 1 (2)
Zone No: 7

Document ID: REP0001PG001_C00F00_R0300GS008
n-th copy: 0
n-th fax: 0
Resolution level: 300
Scanning type: grayscale
Color depth (in bits): 8
Degradation type: original

Figure 16: Application window after modifying a page-related record

To save any modified page-related record, you can carry out one of the following methods:

a. Clicking ‘Save Page Records’ button

b. Clicking ‘Save All Records’ button

c. Selecting ‘File’ ª ‘Save Page Records’

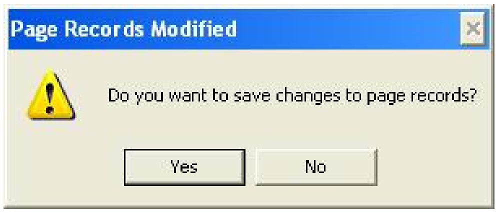
d. Pressing ‘Alt-F-P’

e. Selecting ‘File’ ª ‘Save All Records’

f. Pressing ‘Alt-F-A’

g. Pressing ‘Ctrl-S’

When you try to do any change that will remove the focus from the selected page without saving the modifications done on its records, then a warning dialog will show up to avoid losing your changes.



Page Records Modified

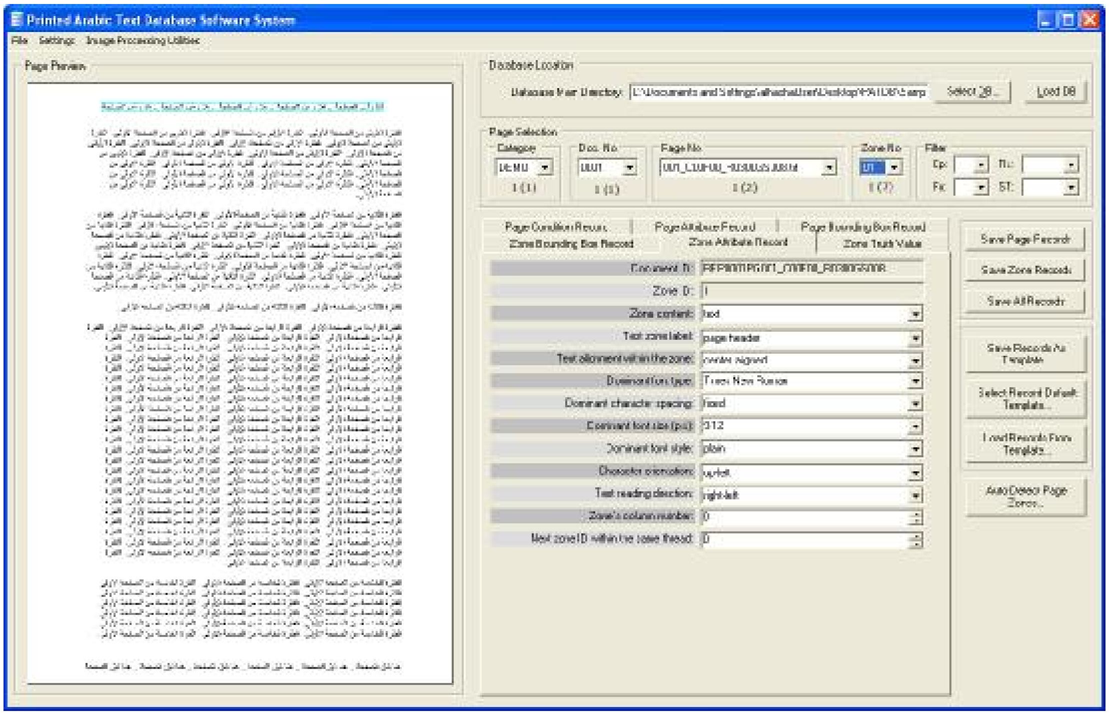
Do you want to save changes to page records?

Figure 17: A warning dialog when a page record is modified but not saved

### C.3.3.1.6 Presenting a Page Zone and its Associated Records

To display the records associated with an identified zone on a page you select, simply select the zone from the zone number dropdown box in the 'Selected Page' area. After you select the zone, they system will populate the three zone-related ('Zone Bounding Box Record' tab, 'Zone Attribute Record' tab and 'Zone Truth Value' tab). The focus will go to the 'Zone Attribute Record' tab if none of the zone-related tabs in already selected.

Let us try to load the records associated with the first zone of the previously selected page image. To do so, select '01' from the zone number dropdown box (Figure 18).



Database Main Directory: C:\Documents and Settings\alhashal\user\Desktop\PATDB\Samp

Category: DEMO 1 (1)

Doc. No: 0001 1 (1)

Page No: [illegible] 1 (2)

Zone No: 01 1 (7)

Cp: Fx: Ts: ST:

Document ID: [illegible]

Zone ID: 1

Zone content: text

Text zone label: page header

Text allignment within the zone: center aligned

Dominant font type: Times New Roman

Dominant character spacing: fixed

Dominant font size (pts): 312

Dominant font style: plain

Character orientation: up-left

Text reading direction: right-left

Zone's column number: 0

Next zone ID within the same thread: 0

Figure 18: Application window after selecting a zone

## C.3.3.1.7 Modifying and Saving Page Zone Associated Records

After you select a zone, you can modify any value related to any record associated with it. Similar to the page-related records, when you modify any record related to the zone you select, an asterisk (*) will be shown after the corresponding record tab name to indicate that the record is modified but not saved.

To save any modified zone-related record, you can carry out one of the following methods:

a. Clicking ‘Save Zone Records’ button

b. Clicking ‘Save All Records’ button

c. Selecting ‘File’ ª ‘Save Zone Records’

d. Pressing ‘Alt-F-Z’

e. Selecting ‘File’ ª ‘Save All Records’

f. Pressing ‘Alt-F-A’

g. Pressing ‘Ctrl-S’

When you try to do any change that will remove the focus from the zone that has modifications not yet saved, a warning dialog will show up to avoid losing your changes.

### C.3.3.1.10 Saving Page and/or Page Zone Associated Records as Template

Sometimes, you need to copy a similar record data of a page and/or a zone to another page or zone. You can do so by first saving the record(s) data of the intended page and/or zone as template then loading this template to the target page and/or zone.

To save the ‘Page Condition Record’ and ‘Page Attribute Record’ data of the page ‘DEMO0001P001_C00F00_R0300GS008.tif’ as template, follow the steps below:

1. Load ‘DEMO0001P001_C00F00_R0300GS008.tif’ page. Refer to ‘Presenting a Page Image and its Associated Records’ section for more information about how to load a page image.

2. Show ‘Save Records As Template’ dialog by carrying one of the following methods:

   a. Clicking ‘Save Records As Template…’ button

   b. Selecting ‘File’ ª ‘Save Records As Template…’

   c. Pressing ‘Alt-F-T’

   d. Pressing ‘Ctrl-Shift-S’

3. Check both ‘Page Condition Record’ and ‘Page Attribute Record’ checkboxes.

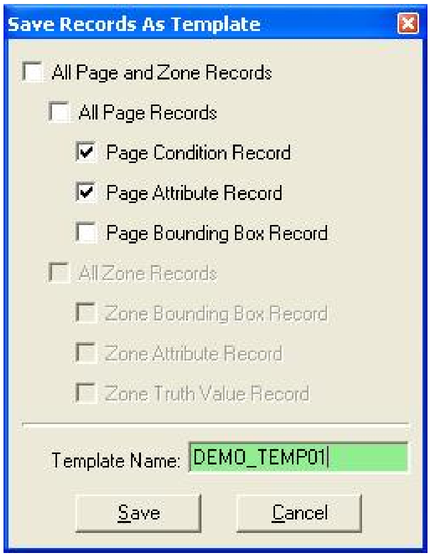
4. Type a name, say ‘DEMO_TEMP01’, in the template name textbox (Figure 20).

---

***Hint:***

***The background color of the template name textbox will change according to the template name availability. If the template name is not used, the textbox background color will be green. Otherwise, it will be red.***

---



Save Records As Template

☐ All Page and Zone Records

☐ All Page Records

☑ Page Condition Record

☑ Page Attribute Record

☐ Page Bounding Box Record

☐ All Zone Records

☐ Zone Bounding Box Record

☐ Zone Attribute Record

☐ Zone Truth Value Record

Template Name: DEMO_TEMP01

Figure 19: Save records as template dialog

5. Save the template by carrying out one of the following methods:

    a. Clicking ‘Save’ button

    b. Pressing ‘Alt-S’

6. A confirmation message showing the number of the saved records pops up, click 'OK' to confirm.

C.3.3.1.11 Setting Page and/or Page Zone Associated Records Default Templates

After you save more than one page-related and zone-related records as templates, you may need to assign a default template for each page-related and zone-related record. You will benefit from this assignment when you load a page or zone template into a page or zone record, respectively.

Let us set the default template of the 'Page Condition Record' to the 'Page Condition Record' template created in the above point. Carry out the following steps to accomplish this setting:

1. Show 'Select Record Default Template' dialog by carrying out one of the following methods:

   a. Clicking 'Select Record Default Template…' button

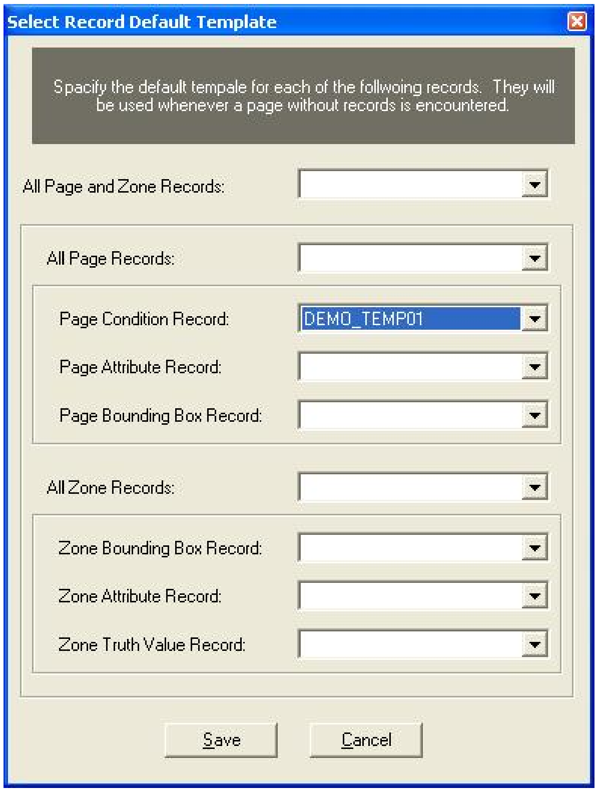
   b. Selecting 'Settings' ª 'Select Record Default Template…'

   c. Pressing 'Alt-T-T'

2. From the 'Page Condition Record' dropdown menus, select 'DEMO_TEMP01' (Figure 21).

***Hint:***

***Each record dropdown menu will have only the template names that correspond to it. For example, the template created in the above point, 'DEMO_TEMP01', stores only the 'Page Condition Record' and 'Page Attribute Record' as template. This means that only the 'Page Condition Record' and 'Page Attribute Record' dropdown menus will have the template name 'DEMO_TEMP01' in them.***



Select Record Default Template

Spacify the default tempale for each of the follwoing records. They will be used whenever a page without records is encountered.

All Page and Zone Records:

All Page Records:

Page Condition Record: DEMO_TEMP01

Page Attribute Record:

Page Bounding Box Record:

All Zone Records:

Zone Bounding Box Record:

Zone Attribute Record:

Zone Truth Value Record:

Figure 20: Select record default template dialog

3. Confirm the changes by carrying out one of the following:
    a. Clicking ‘Save’ button
    b. Pressing ‘Alt-S’

### C.3.3.1.12 Loading Page and / or Page Zone Associated Records Templates

When you add a new page into the database, definitely you need to populate its page-related records quickly. You can do so by loading any saved record (as template) to this page in three steps only.

In the ‘Sample DB’ provided with this system, the second page of the ‘DEMO0001’ document does not have a ‘Page Condition Record’ created for it. Let us load the ‘Page Condition Record’ template stored in the above point into it. Carry out the following steps in order to accomplish this job:

1. Load ‘DEMO0001P002_C00F00_R0300GS008.tif’ page. Refer to ‘Presenting a Page Image and its Associated Records’ section for more information about how to load a page image.

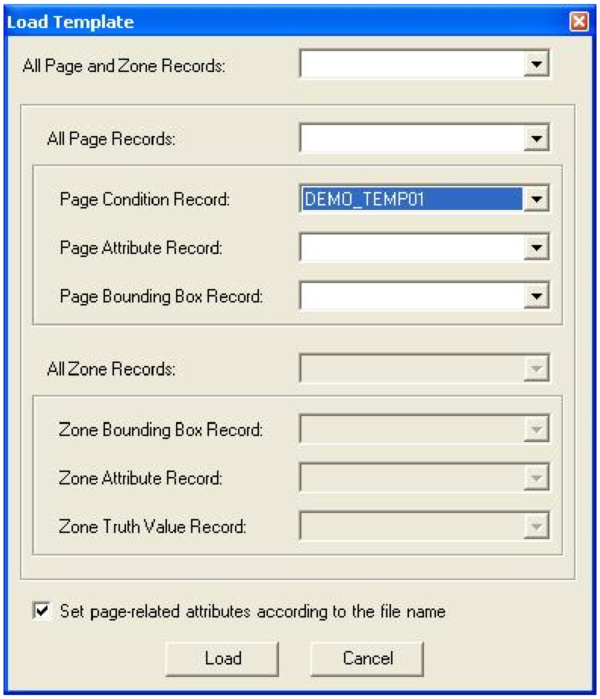
2. Show ‘Load Template’ dialog (Figure 22) by carrying out one of the following steps:

    a. Clicking ‘Load Records From Template…’ button

    b. Selecting ‘File’ ª ‘Load Records From Template…’

    c. Pressing ‘Ctrl-Shift-L’

3. If ‘DEMO_TEMP01’ is already set as a value for the ‘Pgae Condition Record’ dropdown menu, then keep it as is. Otherwise, change it to be ‘DEMO_TEMP01’



Load Template

All Page and Zone Records:

All Page Records:

Page Condition Record: DEMO_TEMP01

Page Attribute Record:

Page Bounding Box Record:

All Zone Records:

Zone Bounding Box Record:

Zone Attribute Record:

Zone Truth Value Record:

☑ Set page-related attributes according to the file name

Figure 21: Load template dialog

4. Load the template to the corresponding record by carrying out one of the following steps:

    a. Clicking ‘Load’ button

    b. Pressing ‘Alt-L’

5. A confirmation record showing the number of loaded records pops up, click ‘OK’ to confirm it.

6. Save the record. Refer to ‘Modifying and Saving Page Associated Records’ section for more information about how to save changes done on a page-associated record.

## C.3.3.2 ATR-RELATED FUNCTIONS

### C.3.3.2.1 Noise Reduction Using Statistical Based Smoothing Approach

To reduce a noise encountered on a page image, say 'DEMO0001P001_C00F00_R0300GS008.tif' page, using the statistical based smoothing approach; you need to carry out the following steps in sequence:

1. Load 'DEMO0001P001_C00F00_R0300GS008.tif'. Refer to 'Presenting a Page Image and its Associated Records' section for more information about how to load a page image.

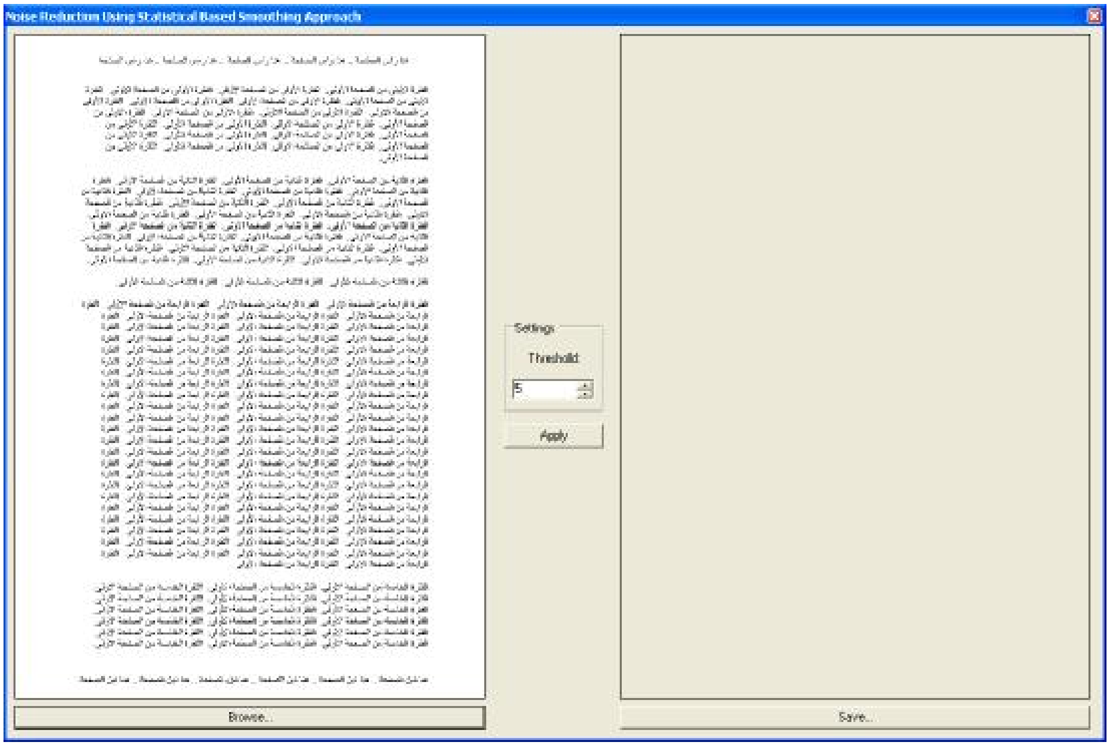
   ***Hint:***

   ***You can load images from outside the database by starting from step 2 below and then clicking 'Browse' button.***

2. Show 'Noise Reduction Using Statistical Based Smoothing Approach' window (Figure 23) by carrying out one of the following methods:

   a. Selecting 'Image Processing Utilities' ª 'Noise Reduction' ª 'Statistical Based Smoothing…'

   b. Pressing 'Alt-U-N-S'



Threshold: 5

Figure 22: Statistical based smoothing window

3. Set the ‘Threshold’ field to a value between 1 and 8. This value represents the minimum number of black pixels in the 3×3 image window, that surround the pixel in the middle of the image window, which will change the status of the middle pixel from white to black.

4. Click the ‘Apply’ button to run the algorithm on the presented page image on the left side.

5. The resulting page image is presented on the right side. You can save it by clicking the ‘Save’ button and providing a name for it. An automatically generated name based on the original page image name will be prompted. You can accept it or change it if you wish.

#### C.3.3.2.2 Noise Reduction Using Average Smoothing Approach

To reduce a noise encountered on a page image, assume ‘DEMO0001P001_C00F00_R0300GS008.tif’ page, using the average smoothing approach; you need to carry out the following steps in sequence:

1. Load ‘DEMO0001P001_C00F00_R0300GS008.tif’. Refer to ‘Presenting a Page Image and its Associated Records’ section for more information about how to load a page image.

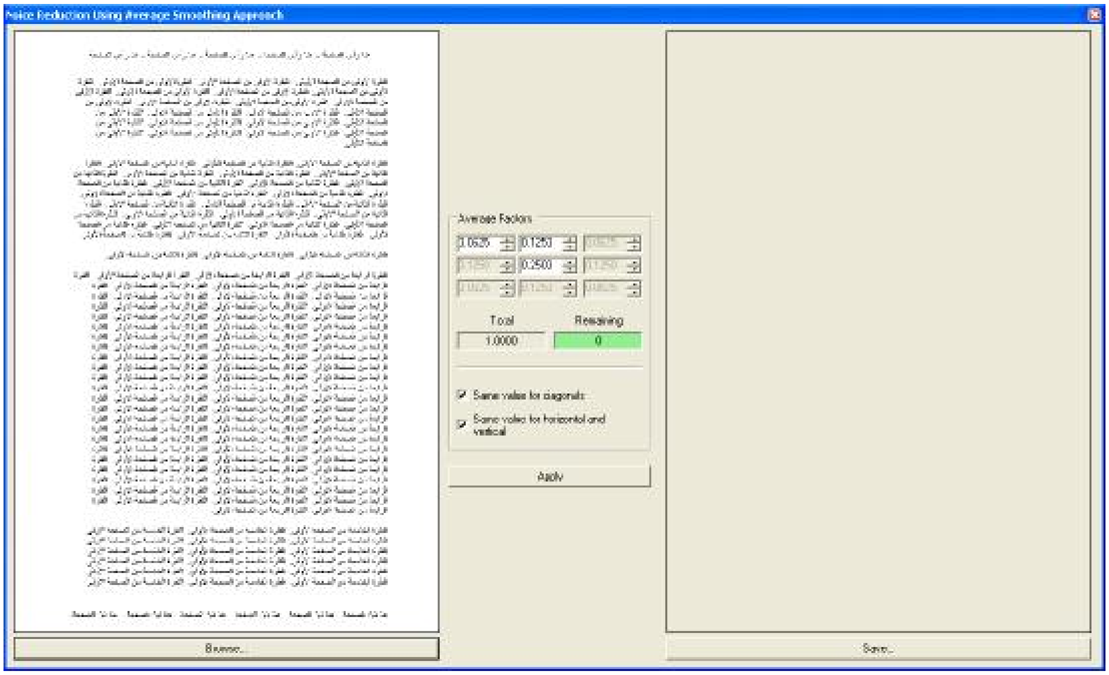
   ***Hint:***

   ***You can load images from outside the database by starting from step 2 below and then clicking ‘Browse’ button.***

2. Show ‘Noise Reduction Using Average Smoothing Approach’ window (Figure 24) by carrying out one of the following methods:

   a. Selecting ‘Image Processing Utilities’ ª ‘Noise Reduction’ ª ‘Average Smoothing…’

   b. Pressing ‘Alt-U-N-A’



Noise Reduction Using Average Smoothing Approach

Average Factors

| | | |
|---|---|---|
| 0.0625 | 0.1250 | 0.0625 |
| 0.1250 | 0.2500 | 0.1250 |
| 0.0625 | 0.1250 | 0.0625 |

| Total | Remaining |
|---|---|
| 1.0000 | 0 |

☑ Same value for diagonals

☑ Same value for horizontal and vertical

Figure 23: Average smoothing window

3. Set the ‘Average Factors’ fields to a total sum equals to 1. These values represent the scales by which each entry in the 3×3 image window will be multiplied. The sum of the multiplications will decide the color of the pixel in the middle of the image window.

***Hint:***

***The ‘Total’ field presents the summation of the average factors while the ‘Remaining’ field represents the value of the ‘Total’ field after subtracting 1 from it.***

***Hint:***

***You will notice that the background color of the 'Remaining' field is always red when its value is not equals to zero which indicates that the value is invalid.***

***Hint:***

***The two checkboxes in the 'Average Factors' box helps you to set balanced average factors, i.e., same values on the diagonals and/or same value on the vertical and horizontal lines.***

4. Click the 'Apply' button to run the algorithm on the presented page image on the left side

5. The resulting page image will be presented on the right side. You can save it by clicking the 'Save' button and providing a name for it. An automatically generated name based on the original page image name will be prompted. You can accept it or change it if you wish.

### C.3.3.2.3 Skew Correction Using a Basic Technique

To correct the skew encountered on a page image, assume 'DEMO0001P001_C00F00_R0300GS008.tif' page, using the basic skew correction approach; you need to carry out the following steps in sequence:

1. Load 'DEMO0001P001_C00F00_R0300GS008.tif'. Refer to 'Presenting a Page Image and its Associated Records' section for more information about how to load a page image.

***Hint:***

***You can load images from outside the database by starting from step 2 below and then clicking 'Browse' button.***

2. Show ‘Correcting Page Skew Using Basic Approach’ window (Figure 25) by carrying out one of the following methods:

    a. Selecting ‘Image Processing Utilities’ ª ‘Skew Correction’ ª ‘Basic…’

    b. Pressing ‘Alt-U-S-B’

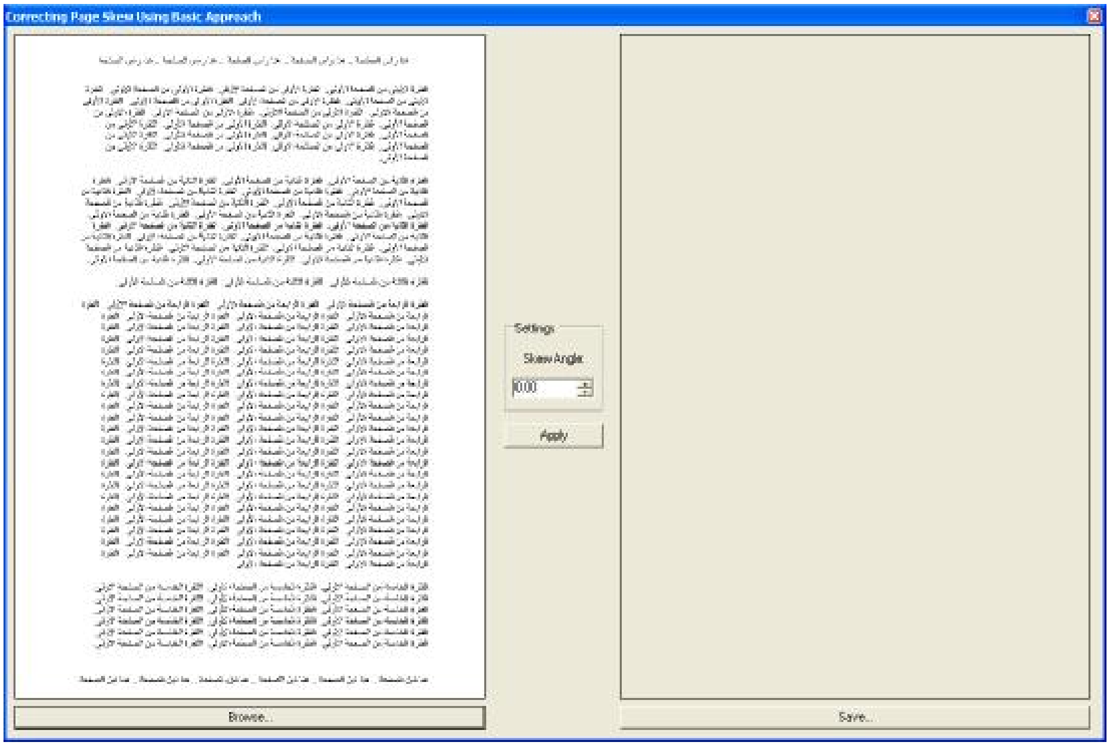


Figure 24: Basic skew correction window

# APPENDIX D

# PATDB SOFTWARE SYSTEM DESIGN

## D.1 INTRODUCTION

When a developer builds a system for others to extend, he normally utilize some of the software design techniques to make other developers’ job easier. This appendix provides developers with an abstract class diagram of the printed Arabic text database (PATDB) software system (Section 2) the list of tools used in development process of the PATDB software system (Section 3).

## D.2 CLASS DIAGRAM

Figure 25 shows an abstract class diagram of PATDB software system. It contains three main categories of classes: static classes, main classes, simple classes. The static classes’ main characteristic is that they provide a set of general-purpose services/functions to the rest of the classes in the system. The static classes, here, correspond to the utilities category in Chapter 6. The main classes carry out the main functions provided by the system. The main classes, here, correspond to the forms, modules and algorithms categories in Chapter 6. The simple classes, as their name indicates, are simple classes encapsulate a set of entries into a single object, which makes the manipulation job of the entries set easier. The simple classes, here, correspond to the linkers’ category in Chapter 6.

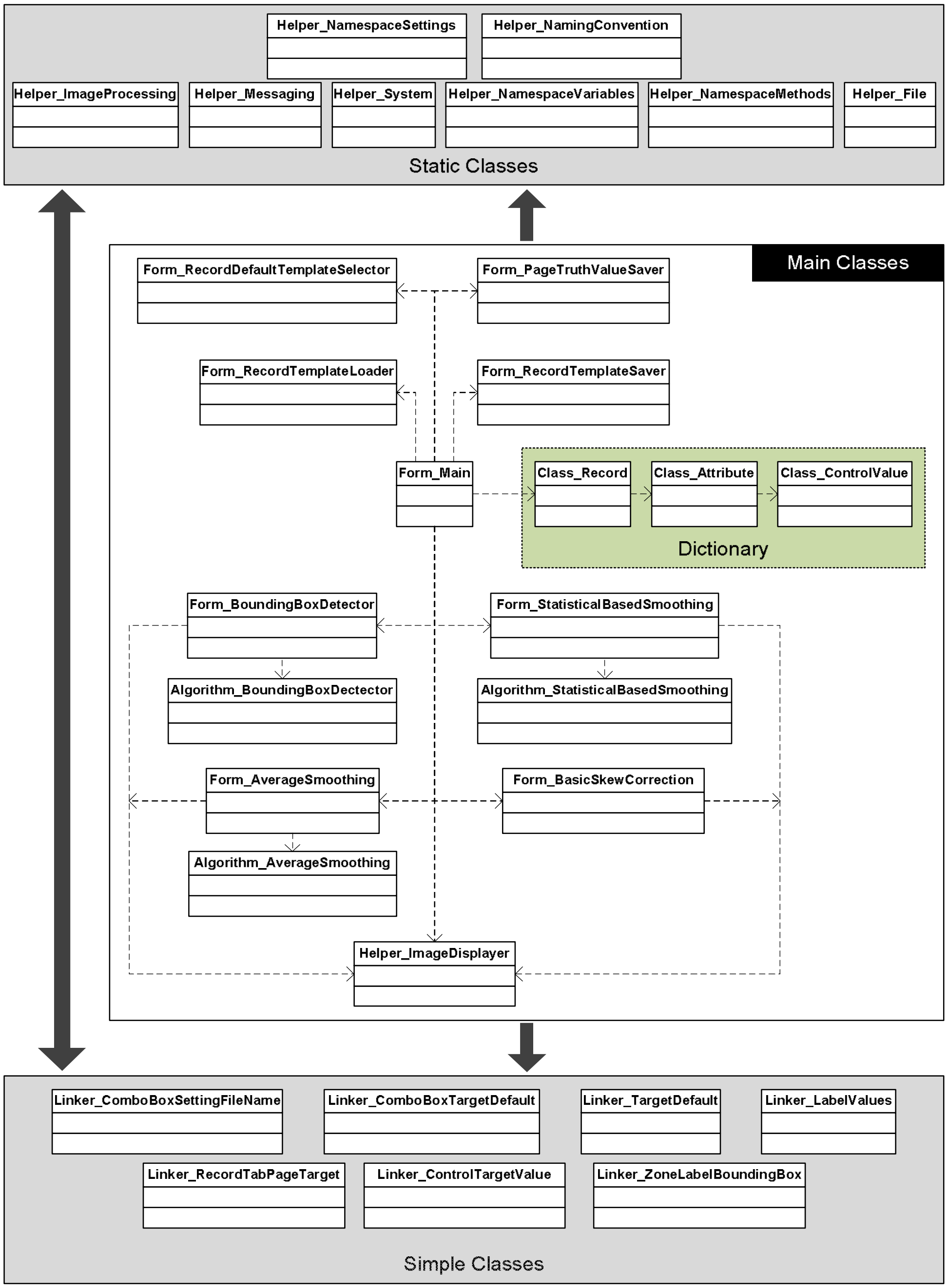


Figure 25: An abstract class diagram for PATDB software system

## D.3 USED TOOLS

Table 11 lists the tools that were used in preparing PATDB software system and its related elements such as this report.

TABLE 11: A LIST OF USED TOOLS IN DEVELOPMENT

| Tool | Main Intended Job |
|---|---|
| Microsoft® Office Word 2007 | Preparing documents |
| Microsoft® Office Visio 2007 | Preparing diagrams |
| Microsoft® Office Excel 2007 | Preparing charts |
| SharpDevelop 1.0.1 by icsharpcode.net (free) | Writing-up the PATDB software system code |
| Picas by Google.com (free) | Viewing page images |
| FileMenu Tools 5.2 by LopeSoft (free) | Renaming file names in batch |
| MathType 6.0a | Preparing formulas |

# REFERENCES

[JSON-a] Introducing JSON. http://www.json.org/ (02-Feb-2008)

[Srih07] S. N. Srihari and G. R. Ball, “An Assessment of Arabic Handwriting Recognition Technology,” Center of Excellence for Document Analysis and Recognition (CEDAR), 2007.

[Lori06] L. M. Lorigo and V. Govindaraju, “Off-line Arabic Handwriting Recognition: A Survey,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28 (5), May 2006, pp. 712-724.

[Alba95] B. Al-Badr and S. Mahmoud, “Survey and Bibliography of Arabic Optical Text Recognition,” Signal Processing 41, Jan. 1995, pp. 49-77.

[Bipp95] R. Bippus and V. Margner, “Data Structures and Tools for Document Database Generation: An Experimental System,” Proceedings of the Third International Conference on Document Analysis and Recognition, vol. 2, 14-16 Aug. 1995, pp. 711-714.

[Phil93a] I. T. Phillips, S. Chen and R. M. Haralick, “CD-ROM Document Database Standard,” Proceedings of the Second International Conference on Document Analysis and Recognition, 20-22 Oct. 1993, pp. 478-483.

[Khar99] N. Kharma, M. Ahmed and R. Ward, “A New Comprehensive Database of Handwritten Arabic Words, Numbers and Signatures used for OCR Testing,” Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering, Shaw Conference Center, Edmonton, Alberta, Canada, 9-12 May 1999, pp. 766-768.

[Märg01] V. Märgner and M. Pechwitz, ”Synthetic Data for Arabic OCR System Development,” ICDAR, Proceedings of the Sixth

International Conference on Document Analysis and Recognition, 2001, pp. 1159-1163.

[Hull94] J. J. Hull, "A Database for Handwritten Text Recognition Research," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16 (5), May 1994, pp. 550-554.

[Alma02] S. Alma'adeed, D. Elliman and C. A. Higgins, "A Data Base for Arabic Handwritten Text Recognition Research," Proc. Eighth Int'l Workshop Frontiers in Handwriting Recognition, 2002, pp. 485-489.

[Suen90] C. Y. Suen, presented at the int. Workshop frontiers handwriting recogn., Montreal, Canada, 2-3 Apr. 1990.

[Suen92] C. Y. Suen, "At the Frontiers of OCR," Proceedings of the IEEE, vol. 80 (7), July 1992, pp. 1093-1100.

[Mart99] U. Marti and H. Bunke, "A full English sentence database for off-line handwriting recognition," Proc. of the 5th Int. Conf. on Document Analysis and Recognition, ICDAR'99, Bangalore, 1999, pp. 705-708.

[Joha78] S. Johansson, G. N. Leech and H. Goodluck, "Manual of Information to Accompany the Lancaster-Oslo/Bergen Corpus of British English, for Use with Digital Computers," Department of English, University of Oslo, Norway, 1978.

[UNIB] http://iamwww.unibe.ch/~zimmerma/iamdb/iamdb.html (11-May-2008)

[Pech02] M. Pechwitz et al. "IFN/ENIT - Database of Handwritten Arabic Words," In Proc. Of CIFED 2002, Hammamet, Tunisia, 21-23 October 2002, pp. 129–136.

[Märg05] V. Märgner, M. Pechwitz and H. El Abed, "ICDAR 2005 Arabic Handwriting Recognition Competition," Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05), 2005.

[Aloh03] Y. Al-Ohali, M. Cheriet and C. Suen, “Databases for Recognition of Handwritten Arabic Cheques,” Pattern Recognition, vol. 36, 2003, pp. 111-121.

[ERIM] www.erim.org (22-Jan-2008)

[Davi97] R. Davidson and R. Hopely, “Arabic and Persian OCR Training and Test Data Sets,” Proc. of Symp. on Document Image Understanding Technology, 30 April-2 May 1997.

[Phil93b] I. T. Phillips et al., “The Implementation Methodology for CD-ROM English Document Database,” Proceedings of the Second International Conference on Document Analysis and Recognition, 20-22 Oct. 1993, pp. 484 - 487.

[Howe00] D. Howell, “Getting to grips with Graphic file format,” Computer Publishing, Issue 9, 2000.

[Ferr01] X. Ferré et al., “Usability Basics for Software Developers,” IEEE Software, vol. 18 (1), Jan./Feb. 2001, pp. 22-29.

[Krish98] S. Krishnamurthi and M. Felleisen, “Toward a Formal Theory of Extensible Software,” ACM SIGSOFT Software Engineering Notes, vol. 23 (6), Nov. 1998, pp. 88-98.

[Mahm94] S. A. Mahmoud, “Arabic Character Recognition Using Fourier Descriptors and Character Contour Encoding,” Pattern Recognition, vol. 27 (6), 1994, pp. 815-824.

[JSON-b] JSON: The Fat-Free Alternative to XML. http://www.json.org/xml.html (17-May-2008)

# VITA

Amin Al-Hashim was born on April 18, 1982 in Sanabis, a village of Qatif. He grew up and finished his precollege study in his home village, Sanabis. He earned his Bachelor of Science degree in Computer Science in June 2005 with second honor distinction from King Fahd University of Petroleum & Minerals (KFUPM). Al-Hashim completed his Master of Science degree in Computer Science in June 2009 from KFUPM.

Amin Al-Hashim is currently a graduate assistant at Information and Computer Science Department of KFUPM. He is planning to be part of the faculty members of his current department after getting abroad soon and earning a PHD degree in Computer Science.