

xDOC: A System for XML Based Document Annotation and Searching

Michael K. Baldwin

Department of Computer Science
Tennessee Technological University
Cookeville, TN



Background

- “ Aside from reading annotation is the most common activity involving documents
- “ Annotations are added to the most significant parts of a document
- “ Annotations provide additional content describing the content of the document



Background

- “ Annotations are usually in the form of
 - . Handwritten comments
 - . Highlighting
 - . Underlining [3]

- “ Readers use annotations as a guide for locating useful information[4]

Motivation

- “ Performing this kind of annotation electronically can distract a reader from the document
- “ Existing annotation tools require the reader
 - . Look away from the document content
 - . Manipulate the annotation tool interface



Motivation

- “ Restrict the annotations by only adding predefined descriptive annotations
 - . Abstract
 - . Definition
- “ These annotations could be an important addition when stored in a digital library

Introduction

- “ A user could specify a search that locates a keyword only within a specific type of annotation
- “ Search results can be obtained more quickly



Goals

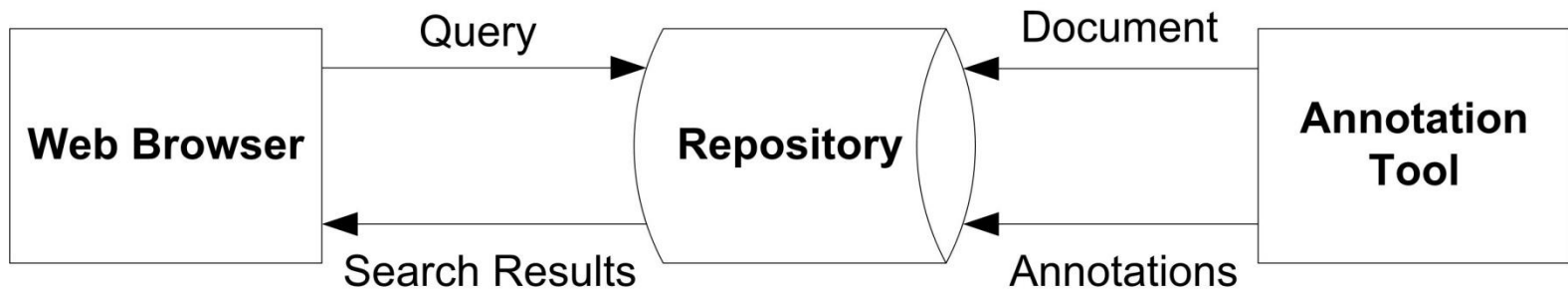
- “ Develop a prototype annotation tool
 - . Annotators can associate metadata with selected areas of the document
- “ Develop a document repository
 - . Search based on user submitted annotations

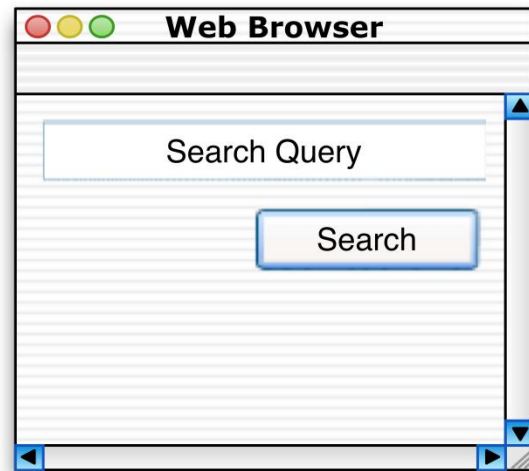
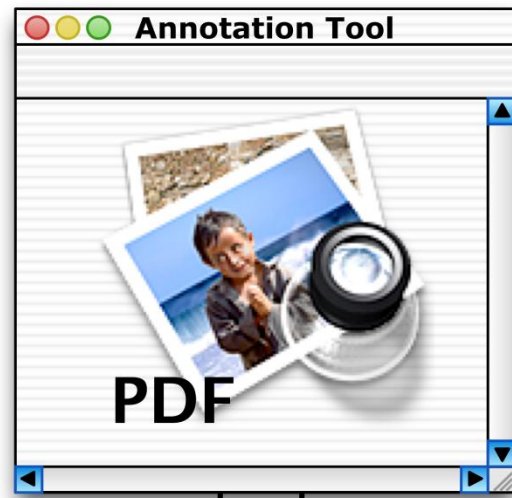


System Architecture

The project consists of two components:

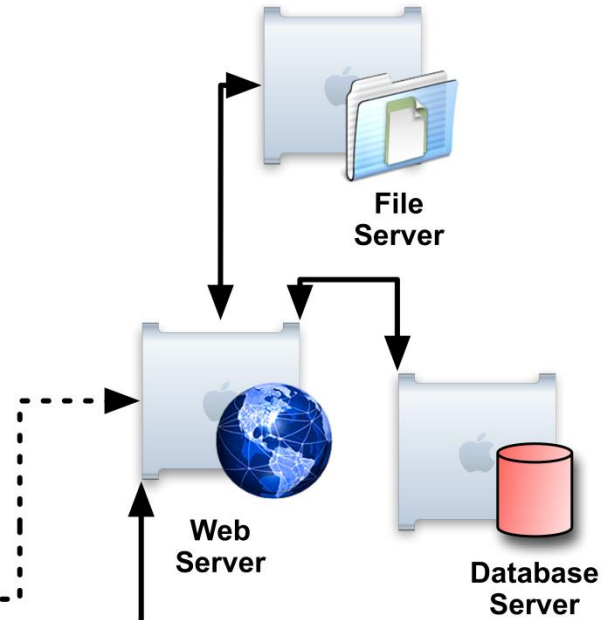
- “ Annotation Tool
- “ Document Repository





Web Interface

Document Repository



Annotation Tool

- “ Load & display a PDF document
- “ Add annotations to a document
- “ Export annotations to the repository

Based on the existing Mac OS X application



Skim

Annotation Tool Architecture

- “ The Skim executable itself was not modified
- “ Skim provides complete support for scripting via AppleScript
- “ Skim also provides the ability to create custom export templates for annotations

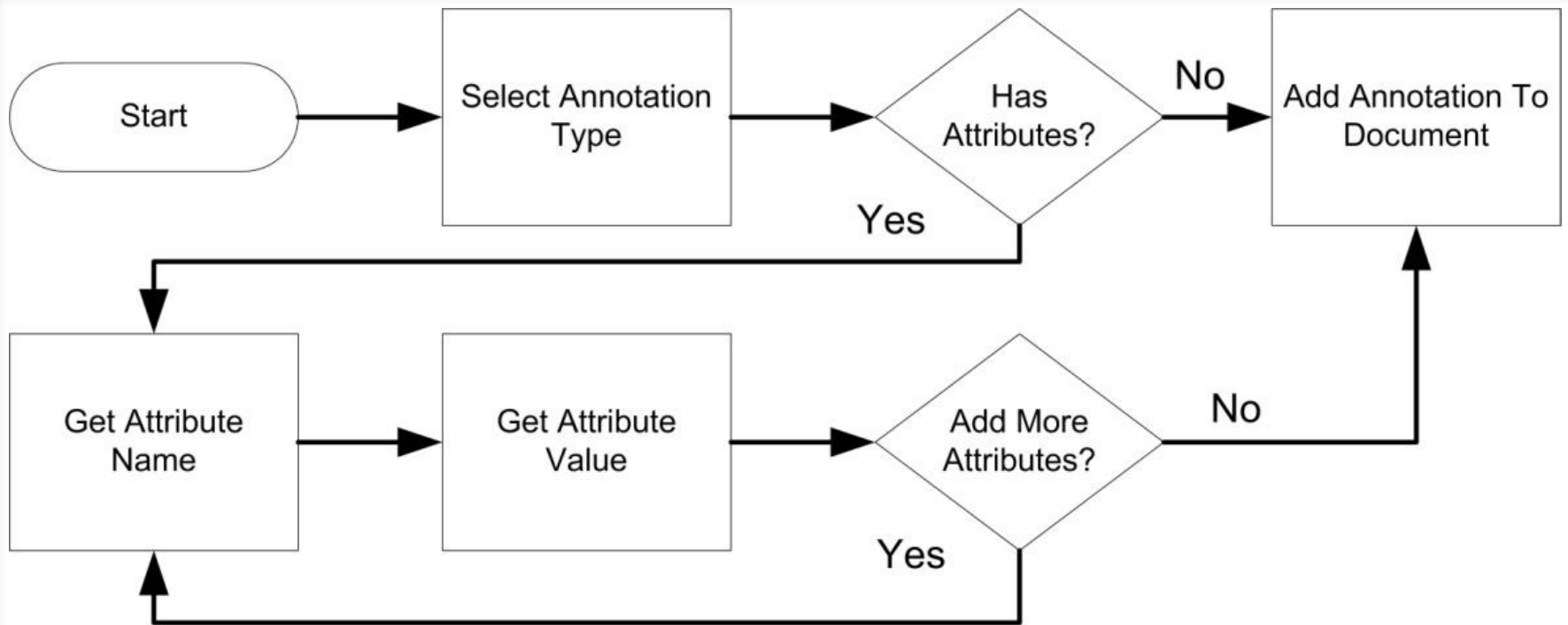


Annotation Tool Architecture

- “ Custom XML export template
- “ AppleScript for adding annotations
 - . Adds an annotation and graphical box to selected area of text
 - . Allows annotator to select an annotation type
 - . Add attributes if that type allows



Add Annotation Script



AnnotationTool

The screenshot shows a PDF viewer window titled "6700.pdf (5 pages)". The document content is as follows:

A Study of Requirement Defects in Undergrad SRS Documents

Michael Baldwin
Department of Computer Science
Tennessee Technological University

Abstract—The Software Requirements Specification document is an important part of the software development process. This document must accurately describe the project requirements, in order for the project to be a success. This paper describes a study designed to determine the most frequently occurring defects in software requirements. The results show which categories are the cause of the most defects, based on the documents studied. This defect information can be used to better direct requirement document inspections.

Index Terms—Software engineering, requirement defect, software requirement specification, defect study.

I. INTRODUCTION

The *Software Requirements Specification (SRS)* document is a crucial part of any software development project. The requirements are the basis for estimation of project cost and schedules, and for the design phase. [1] Therefore it is necessary to ensure that this document accurately describes the project requirements. It has been found that complete, consistent, unambiguous, and correct documentation during the entire software development process improves the overall quality of a system. [2] Other studies have shown that poorly specified requirements are one of the major causes of unavoidable rework, or fixing problems that could have been caught or prevented earlier in the process. [3]

A defect is, within the context of this study, any deficiency in the requirements which can affect the software development process. [2] The goal of this study is to determine what types of defects occur most frequently in student SRS documents. It is important to understand how frequently the different types of defects occur so that overall requirements quality can be improved, by focusing more heavily on preventing these most common defects.

II. HOW THE STUDY WAS CONDUCTED

The study described in this paper was conducted using six SRS documents from an undergraduate software engineering course. The students in this course were required to use the IEEE Recommended Practice for Software Requirements Specifications [4] as the standard for their documents. Five out of the six teams had three members, and the sixth was an individual working alone.

A checklist (described in section III) was used to locate defects in the SRS documents. The documents were read

checklist. Any defects found according to the checklist were recorded for future analysis.

Each occurrence of a defect based on one of the items in the checklist was recorded using a notation by category and the rule number. Throughout the paper a similar notation will be used as shorthand. During the reading phase of the study, the notation used to include the section number of the checklist first in the components previously mentioned, for example III-A-1. Correct would be denoted as III-A-1.

III. THE CHECKLIST

The checklist used in this study was designed to aid in the defect detection process, and allow for further classification. Checklist questions are designed to be easier for the reviewer to locate defects in the document.

This checklist is a subset of the checklist described in [6], and describes six broad categories of defects: unambiguous, complete, consistent, verifiable, and correct. These categories were selected because they are all applicable to student projects. The individual rules within each category were also selected due to their being applicable to this study. Ranking for importance and stability, and ability were excluded since they are not as important to define, for a student project. The exact check items and how to perform the analysis of the SRS documents, is covered in the following sections.

A. Correct

- 1) Are all of the requirements necessary to meet the project objectives?
- 2) Are the functional requirements logically organized and highly cohesive?
- 3) Are there requirements that seem illogical?
- 4) Are the requirements free of syntactical and grammatical errors?
- 5) Are all requirements listed in appropriate sections of the SRS?
- 6) For specific functions, are all specified inputs, outputs, necessary and if so, are they described completely?
- 7) Are contents, formats and constraints of outputs described?
- 8) If there are technical terms used in the requirements, are they the usage appropriate and justified?
- 9) Are there requirements that should really be in another section?

- Open Scripts Folder
- Open AppleScript Utility
- Skim Scripts
 - AddAnnotation
 - AttribAddAnnotation
 - Address Book Scripts
 - Basics
 - ColorSync
 - Finder Scripts
 - Folder Actions
 - Font Book
 - FontSync Scripts
 - Info Scripts
 - Internet Services
 - Mail Scripts
 - Navigation Scripts
 - Printing Scripts
 - Script Editor Scripts
 - Sherlock Scripts
 - UI Element Scripts
 - URLs
- Download Sample
- Edit Sample
- iPhoto Upload Sample
- Move Sample
- Open URL Sample
- Screenshot Sample
- SVN Add
- SVN Checkout
- SVN Commit
- SVN Copy
- SVN Delete
- SVN Move
- SVN Revert
- SVN Status
- SVN Update
- Upload Sample

Add Annotation

Annotation type:

- abstract
- definition
- author
- glossary**

Cancel Add

Document Repository

Custom web based application:



xDoc

“ Built using:

- . PHP
- . xHTML
- . CSS
- . XSLT

” Requires:

- . Apache Web Server
- . PHP5
- . MySQL 5.1

Document Repository

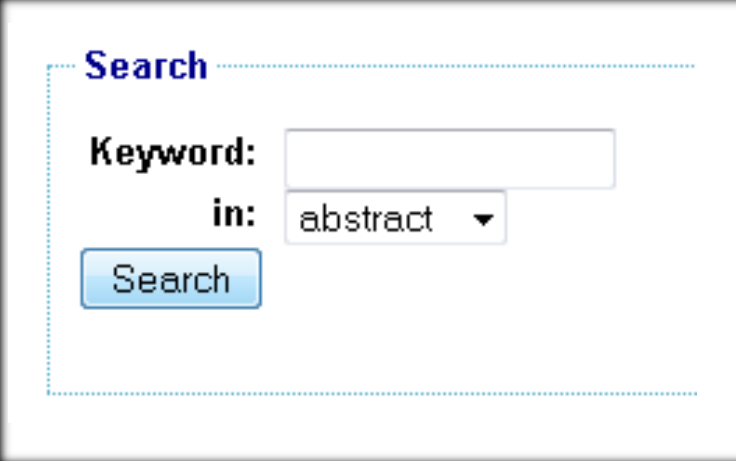
- “ Search for documents in multiple ways
- “ Retrieve documents
- “ View document details
- “ View stored annotations



Search Methods

” Standard Search

- Specify a keyword and select the annotation to search within



The image shows a search interface with the following elements:

- Search**: A blue header for the search section.
- Keyword:** A text input field for entering the search term.
- in:** A dropdown menu currently showing "abstract".
- Search**: A blue button to execute the search.

Search Methods

“ Advanced Search

- Specify a series of conditions consisting of a keyword and annotation type

Advanced Search

Match of the following conditions:

<input type="text" value="abstract"/>	<input type="text" value="contains exactly"/>	<input type="text"/>
<input type="text" value="definition"/>	<input type="text" value="does not contain"/>	<input type="text"/> <input type="button" value="-"/>

Search Methods

” XPath Search

- Specify a keyword and a custom XPath that returns the annotations to search within



A screenshot of a search interface. It features two input fields: the top one is labeled "XPath:" and the bottom one is labeled "Keyword:". Below the "Keyword:" field is a button labeled "Search".



Search Options

[Standard Search](#)

[Advanced Search](#)

[XPath Search](#)

Search Results:

[A Checklist for Requirement Defects Discovery in an Academic Environment](#)

Michael Baldwin

"For software projects, the software requirement specification (SRS) serves as the official statement of user need and what the systems developers are expected to implement. Development of the SRS is a critical task as it becomes the basis for all future development. Utilizing a checklist while inspecting documents, helps in locating defects, i.e. any deficiency with a potential of negative affect on the development process, within the documents. [1] Many studies have shown that these checkli..."

[Details >>](#)

[A Study of Requirement Defects in Undergraduate SRS Documents](#)

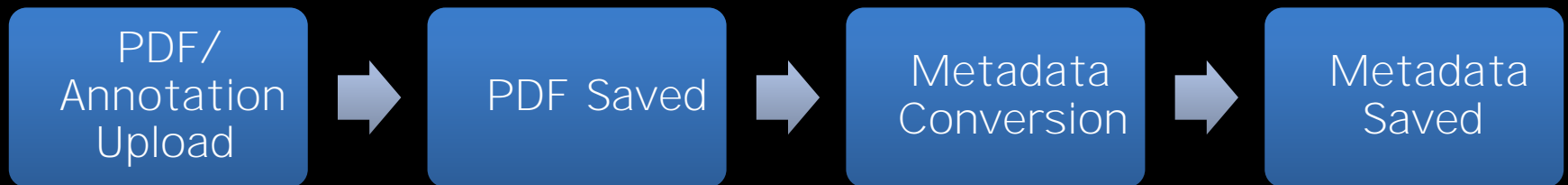
Michael Baldwin

"Abstractâ€”The Software Requirements Specification document is an important part of the software development process. This document must accurately describe the project requirements, in order for the project to be a success. This paper describes a study designed to determine the most frequently occurring defects in software requirements. The results show which categories are the cause of the most defects, based on the documents studied. This defect information can be used to better direct..."

[Details >>](#)

Document Uploads

- “ Document and annotations are uploaded
- “ PDF saved to file server
- “ Annotations are converted to internal form
- “ Metadata stored in database



Metadata Conversion

“ Metadata Converter

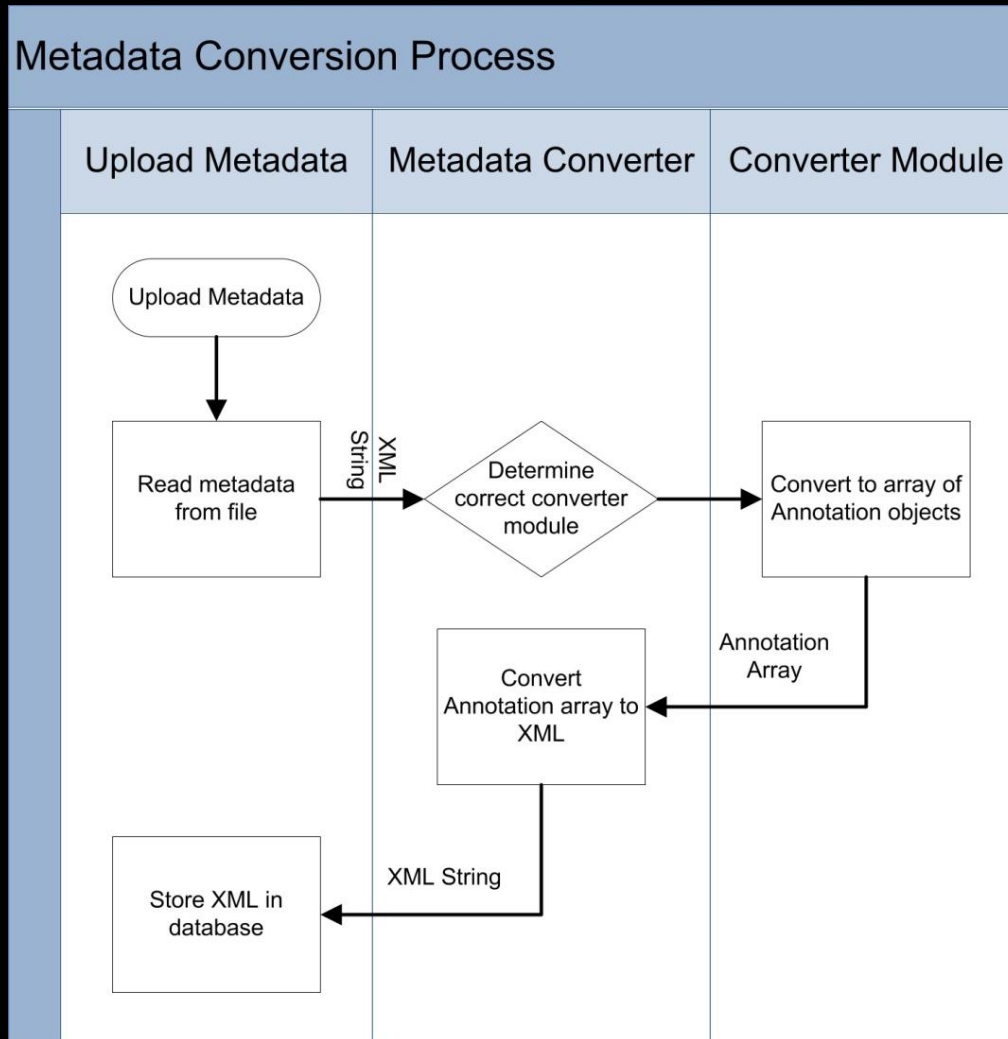
- . Selects the appropriate metadata converter for the input XML then passes them to the mod

“ Metadata Converter Modules

- . Take the raw XML and transform it into a PHP array that is then converted back to the correct XML format by the Metadata Converter



Metadata Conversion



Future Work

- “ Develop a custom cross-platform annotation tool
- “ Perform a study to determine the amount of improvement this method gives to search results



References

1. A. J. Bernheim-Bruhl, David Barger, Anoop Gupta, and J. J. Cadiz. Robust annotation positioning in digital documents. In CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems, pages 281-286. New York, NY, USA, 2001. ACM Press.
2. Katashi Nagao. Digital Content Annotation and Coding. Artech House Inc., 2003.
3. JJ Cadiz, A. Gupta, and Grudin. Using Web annotations for asynchronous collaboration around documents. Proceedings of the 2000 ACM conference on Computer supported cooperative work, pages 309-318, 2000.
4. Kenton O'Hara and Abigail Sellen. A comparison of reading paper and on-line documents. In CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems, pages 335-342, New York, NY, USA, 1997. ACM Press.
5. Catherine C. Marshall. Annotation: from paper books to the digital library. In DL '97: Proceedings of the second ACM international conference on Digital libraries, pages 131-140, New York, NY, USA, 1997. ACM.

