Research Paper
Impact Factor: 3.567
Peer Reviewed Journal

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

A STUDY OF SUPPORTING DOCUMENT ANNOTATION BASED ON CONTENT AND QUERYING VALUE SYSTEM

Pasupuleti Jhansi* Md. Amanatulla**
**Student of M.Tech., Dept. of CSE, Nimra Institute of Science & Technology, Ibrahimpatnam, Vijayawada.*
***Asst. Professor of CSE, Nimra Institute of Science & Technology, Ibrahimpatnam, Vijayawada.*

Abstract
A large number of organizations today generate and share textual descriptions of their products, services, and actions. Such collections of textual data contain significant amount of structured information, which remains buried in the unstructured text. While information extraction algorithms facilitate the extraction of structured relations, they are often expensive and inaccurate, especially when operating on top of text that does not contain any instances of the targeted structured information. The authors present a novel alternative approach that facilitates the generation of the structured metadata by identifying documents that are likely to contain information of interest and this information is going to be subsequently useful for querying the database.

Keywords — Document, Annotation, Adaptive forms, Collaborative platforms.

Introduction

There are many application domains where users create and share information; for instance, news blogs, scientific networks, social networking groups, or disaster management networks. Current information sharing tools, like content management software (e.g., Microsoft Share-Point), allow users to share documents and annotate (tag) them in an ad hoc way. Similarly, Google Base allows users to define attributes for their objects or choose from predefined templates. This annotation process can facilitate subsequent information discovery. Many annotation sys-tems allow only "untaped" keyword annotation: for instance, a user may annotate a weather report using a tag such as "Storm Category 3." Annotation strategies that use attribute-value pairs are generally more expressive, as they can contain more information than untyped approaches. In such settings,the above information can be entered as (Storm Category, 3). A recent line of work toward using more expressive queries that leverage such annotations, is the "pay-as-you-go" querying strategy in Data spaces In Data spaces, users provide data integration hints at query time. The assumption in such systems is that the data sources already contain structured information and the problem is to match the query attributes with the source attributes. Many systems, though, do not even have the basic "attribute-value" annotation that would make a "pay-as-you-go" querying feasible. Annotations that use "attribute-value" pairs require users to be more principled in their annotation efforts. Users should know the underlying schema and field types to use; they should also know when to use each of these fields. With schemas that often have tens or even hundreds of available fields to fill, this task become complicated and cumbersome. This results in data entry users ignoring such annotation capabilities. Even if the system allows users to arbitrarily annotate the data with such attribute-value pairs, the users are often unwilling to perform this task: The task not only requires considerable effort but it also has unclear usefulness for subsequent searches in the future: who is going to use an arbitrary, undefined in a common schema, attribute type for future searches? But even when using a predetermined schema, when there are tens of potential fields that can be used, which of these fields are going to be useful for searching the database in the future? Such difficulties results in very basic annotations, if any at all, those are often limited to simple keywords. Such simple annotations make the analysis and querying of the data cumbersome. Users are often limited to plain keyword searches, or have access to very basic annotation fields, such as "creation date" and owner of document."

How Data Mining Works?

While large-scale information technology has been evolving separate transaction and analytical systems, data mining provides the link between the two. Data mining software analyzes relationships and patterns in stored transaction data based on open-ended user queries. Several types of analytical software are available: statistical, machine learning, and neural networks. **Generally, any of four types of relationships are sought:**

- **Classes**: Stored data is used to locate data in predetermined groups. For example, a restaurant chain could mine customer purchase data to determine when customers visit and what they typically order. This information could be used to increase traffic by having daily specials.
- **Clusters**: Data items are grouped according to logical relationships or consumer preferences. For example, data can be mined to identify market segments or consumer affinities.
- **Associations**: Data can be mined to identify associations. The beer-diaper example is an example of associative mining.

- **Sequential patterns**: Data is mined to anticipate behavior patterns and trends. For example, an outdoor equipment retailer could predict the likelihood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes.

Benefits of Data Mining

1. It's one of the most effective services that are available today. With the help of data mining, one can discover precious information about the customers and their behavior for a specific set of products and evaluate and analyze, store, mine and load data related to them
2. An analytical CRM model and strategic business related decisions can be made with the help of data mining as it helps in providing a complete synopsis of customers
3. An endless number of organizations have installed data mining projects and it has helped them see their own companies make an unprecedented improvement in their marketing strategies (Campaigns)
4. Data mining is generally used by organizations with a solid customer focus. For its flexible nature as far as applicability is concerned is being used vehemently in applications to foresee crucial data including industry analysis and consumer buying behaviors
5. Fast paced and prompt access to data along with economic processing techniques have made data mining one of the most suitable services that a company seek

Literature Survey

According to S. R. Jeffery, M. J. Franklin, and A. Y. Halevy, A primary challenge to large-scale data integration is creating semantic equivalences between elements from different data sources that correspond to the same real-world entity or concept. Dataspaces propose a pay-as-you-go approach: automated mechanisms such as schema matching and reference reconciliation provide initial correspondences, termed candidate matches, and then user feedback is used to incrementally confirm these matches. The key to this approach is to determine in what order to solicit user feedback for confirming candidate matches. According to K. Saleem, S. Luis, Y. Deng, S.-C. Chen, V. Hristidis, and T. Li Crisis Management and Disaster Recovery have gained immense importance in the wake of recent man and nature inflicted calamities such as the terrorist attacks of September 11th 2001 and hurricanes/earthquakes i.e. Katrina (2005), Wilma (2005) and Indian Ocean Tsunami (2004). Most of the recent work has been conducted for crisis management under terrorist attacks and emergency management services under natural disasters with private business continuity and disaster recovery a secondary concern. In this paper, we propose a model for pre-disaster preparation and post-disaster business continuity/rapid recovery. The model is utilized to design and develop a web based prototype of our Business Continuity Information Network (BCIN) system facilitating collaboration among local, state, federal agencies and the business community for rapid disaster recovery. We present our model and prototype with Hurricane Wilma as the case study. As per the study of R. Fagin, A. Lotem, and M. Naor - Assume that each object in a database has m grades, or scores, one for each of m attributes. For example, an object can have a color grade, that tells how red it is, and a shape grade, that tells how round it is. For each attribute, there is a sorted list, which lists each object and its grade under that attribute, sorted by grade (highest grade first). There is some monotone aggregation function, or combining rule, such as min or average, that combines the individual grades to obtain an overall grade. K. C.-C. Chang and S.-w have addresses the problem of evaluating ranked top- queries with expensive predicates.

Objectives of the Study

- To present a novel alternative approach that facilitates the generation of the structured metadata by identifying documents that are likely to contain information of interest and this information is going to be subsequently useful for querying the database.
- To propose logical representation of our access control model that allows us to leverage the features of existing logic solvers to perform various analysis tasks on our model.
- To Adopt/further develop a model for formal, high-level system specification and verification.
- To demonstrate the efficacy of the developed model by applying it to a suitable part of the consortium demonstrator, the network terminal for broadband access.
- To develop a systematic method to refine the specification into synthesizable code and a prototype tool which supports the refinement process and links it to synthesis and compilation tools.

System Analysis

Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying

the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output.

- Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively.
- Select methods for presenting information.
- Create document, report, or other formats that contain information produced by the system.

Existing System

Many annotation systems allow only "untyped" keyword annotation: for instance, a user may annotate a weather report using a tag such as "Storm Category 3". Annotation strategies that use attribute-value pairs are generally more expressive, as they can contain more information than untyped approaches. In such settings, the above information can be entered as (StormCategory,3). The assumption in such systems is that the data sources already contain structured information and the problem is to match the query attributes with the source attributes. Many systems, though, do not even have the basic "attribute-value" annotation that would make a "pay-as-you go" querying feasible. Annotations that use "attribute-value" pairs require users to be more principled in their annotation efforts. Users should know the underlying schema and field types to use; they should also know when to use each of these fields. With schemas that often have tens or even hundreds of available fields to fill, this task become complicated and cumbersome. This results in data entry users ignoring such annotation capabilities.

UML Diagrams

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

System Components

The diagram shows a general view of how desktop and workstation computers are organized. Different systems have different details, but in general all computers consist of components (processor, memory, controllers, video) connected together with a *bus*. Physically, a bus consists of many parallel wires, usually printed (in copper) on the main circuit board of the computer. Data signals, clock signals, and control signals are sent on the bus back and forth between components. A particular type of bus follows a carefully written standard that describes the signals that are carried on the wires and what the signals mean. The PCI standard (for example) describes the PCI bus used on most current PCs.

- Keyboard
- Mouse
- Joystick
- Scanner
- Scanning Device
- Digitizing Tablet

- Touch-Sensitive Screen
- Microphone

Before a starting for actual coding, it is highly important to understand what we are going to create and what it should look like? The requirement specifications from first phase are studied in this phase and system design is prepared.

Data Flow Diagrams

Data Flow diagrams (DFD):

In the DFD, there are four symbols

1. A square defines a source(originator) or destination of system data
2. An arrow identifies data flow. It is the pipeline through which the information flows
3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4 .An open rectangle is a data store, data at rest or a temporary repository of data Process that transforms data flow.

Constructing a DFD

Several rules of thumb are used in drawing DFD'S:

1. Process should be named and numbered for an easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower level details, they are numbered.
4. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each work capitalized.

Types of Data Flow Diagrams

1. Current Physical
2. Current Logical
3. New Logical
4. New Physical

Current Physical

In Current Physical DFD process label include the name of people or their positions or the names of computer systems that might provide some of the overall system-processing label includes an identification of the technology used to process the data.

Current Logical

The physical aspects at the system are removed as much as possible so that the current system is reduced to its essence to the data and the processors that transforms them regardless of actual physical form.

New Logical

This is exactly like a current logical model if the user were completely happy with the user were completely happy with the functionality of the current system but had problems with how it was implemented typically through the new logical model will differ from current logical model while having additional functions, absolute function removal and inefficient flows recognized.

New Physical: The new physical represents only the physical implementation of the new system.

Data Flow Diagram

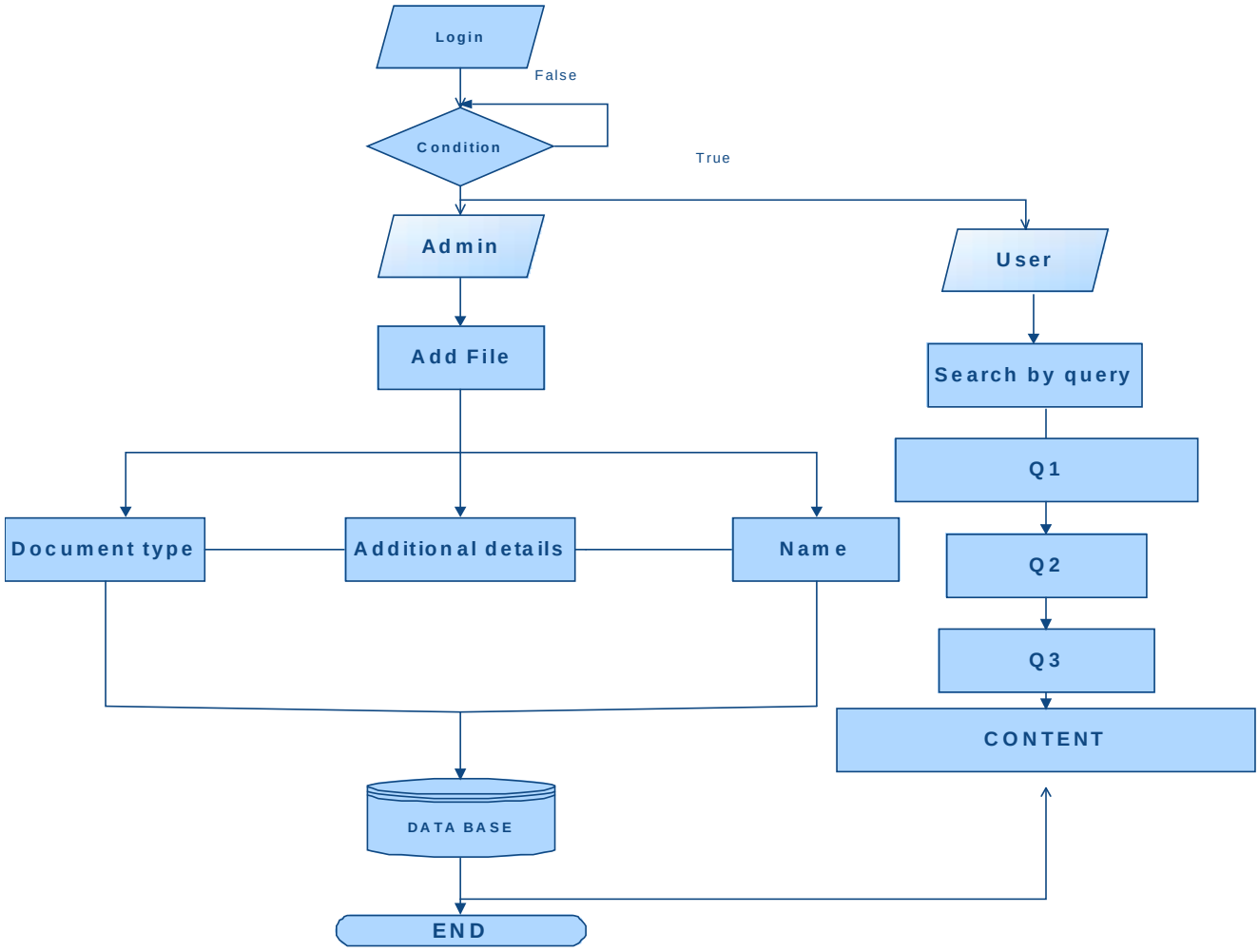

Fig 1.1 Data flow diagram

USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

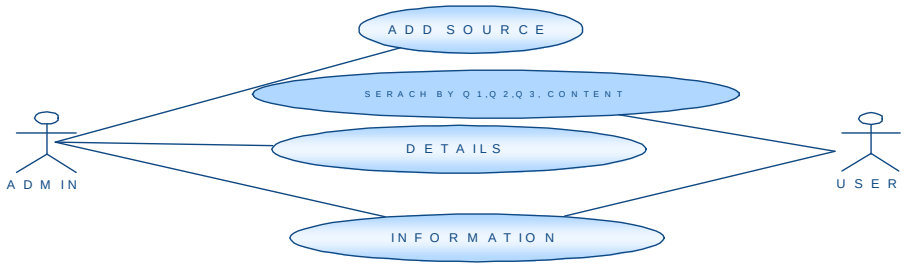

Fig 1.2 Use case diagram

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

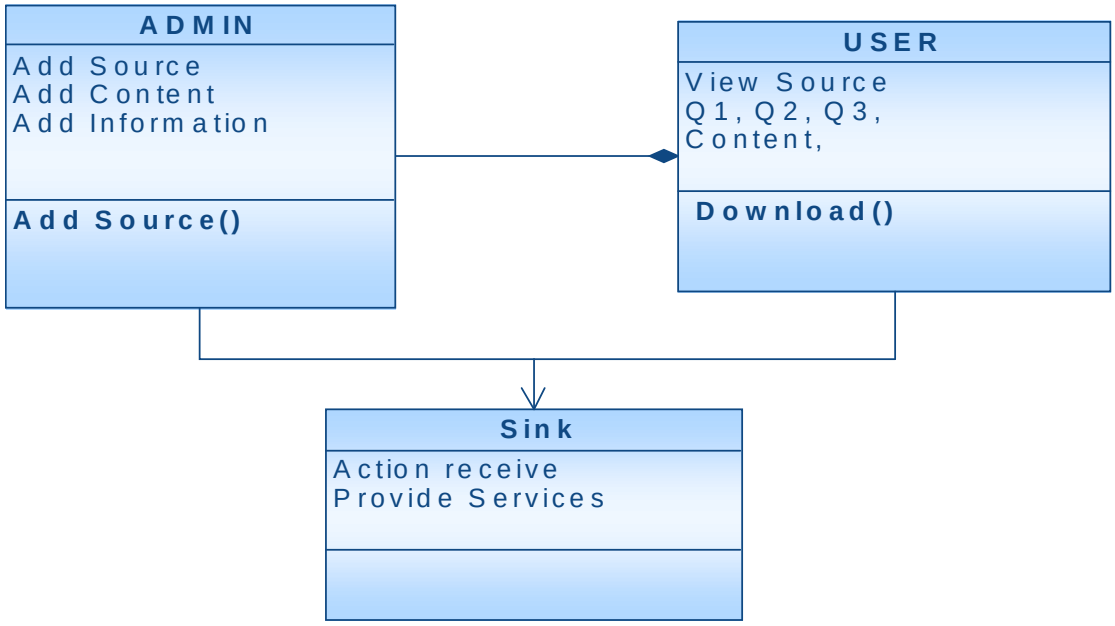

Fig 1.3 Class diagram

Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

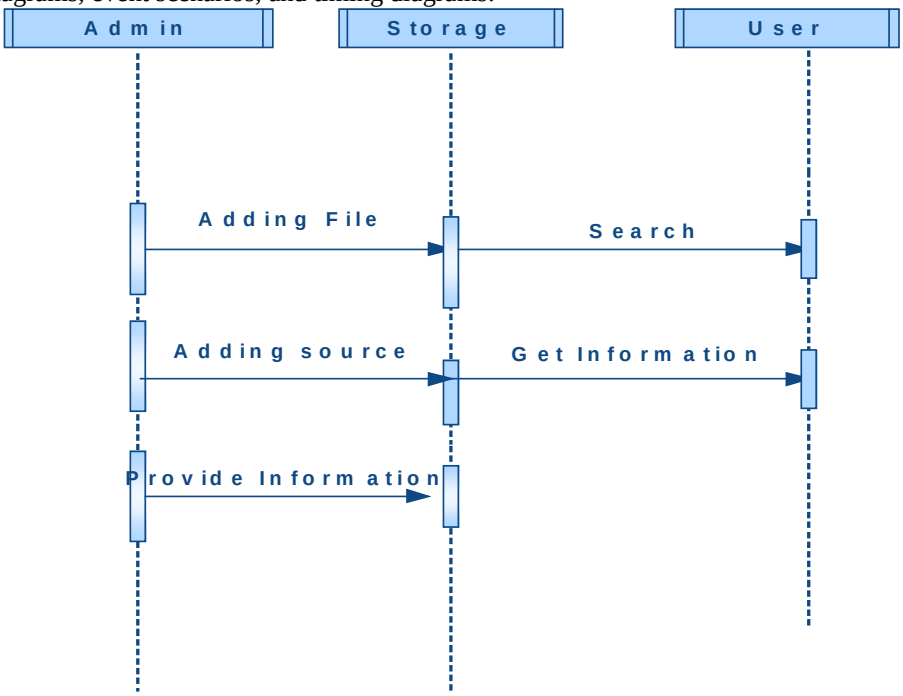

Fig 1.4 Sequence diagram

Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

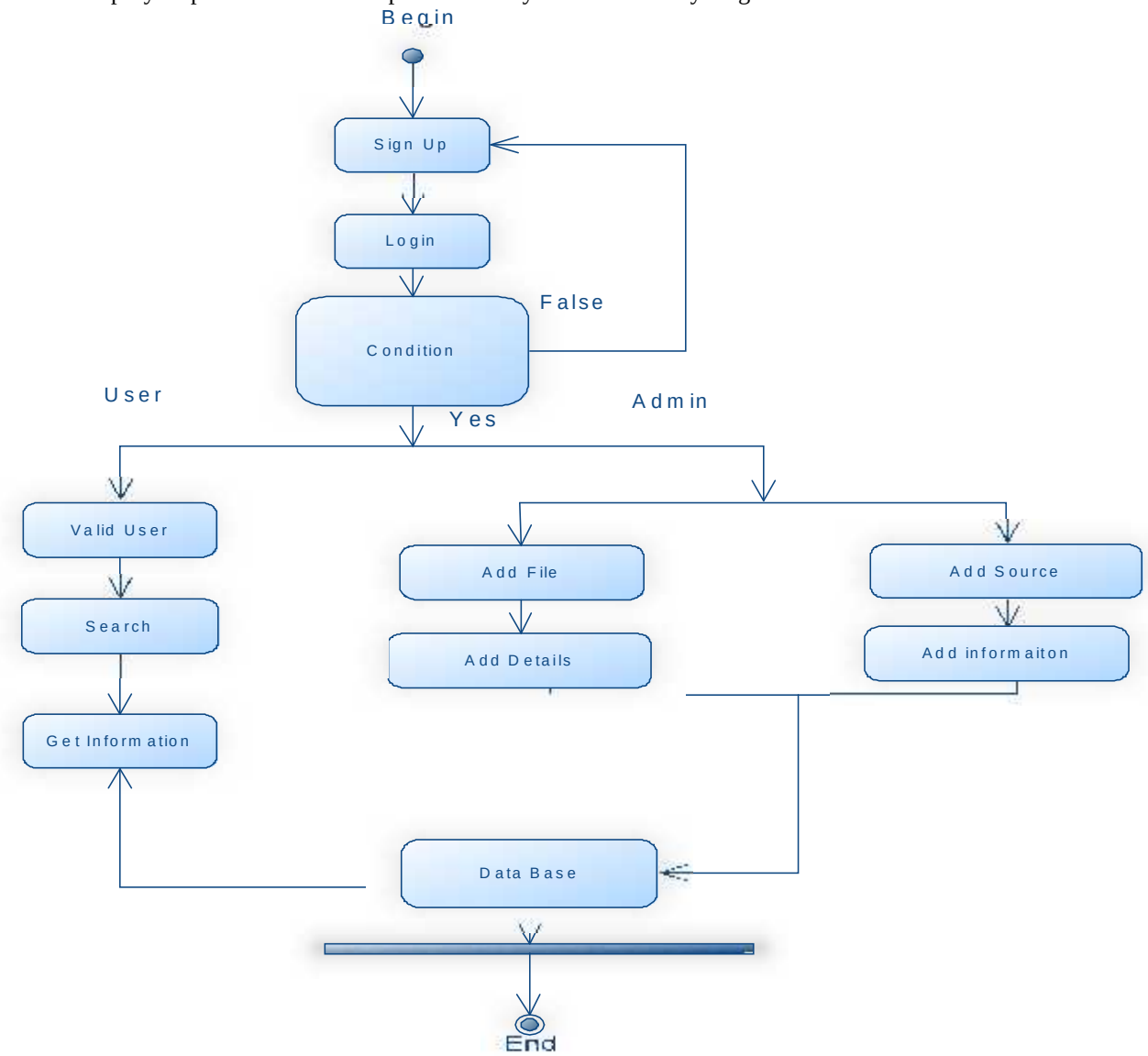

Fig 1.5 Activity diagrams

Conclusion

The authors proposed adaptive techniques to suggest relevant attributes to annotate a document, while trying to satisfy the user querying needs. They present two ways to combine these two pieces of evidence, content value and querying value: a model that considers both components conditionally independent and a linear weighted model. Experiments shows that using our techniques, we can suggest attributes that improve the visibility of the documents with respect to the query workload by up to 50%.

References

1. Head First Java 2nd Edition.
2. http://java.sun.com/javase/technologies/desktop/.
3. http://www.roseindia.net/jdbc/jdbc-access/CreateTable.shtml.
4. http://www.jdbc-tutorial.com/.