INTRUSION DETECTION SYSTEM USING GENETIC ALGORITHM

A PROJECT REPORT

Submitted by

…………………………….

…………………………………

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

…………………………………

…………………………………

……….

BONAFIDE CERTIFICATE

Certified that this project report **"INTRUSION DETECTION SYSTEM USING GENETIC ALGORITHM"** is the bonafide work of **……..** who carried out the project work under my supervision.

ABSTRACT

Network security has played a major role in any network design in recent times. To provide Network security, various tools are available like firewalls and intrusion detection systems (IDS). In firewalls we can't detect *inside intruders* whereas *IDS* can detect them. Here an intrusion detection system based on *Genetic Algorithm* has been proposed. The proposed system performs its tasks at *Host level* and *Network level.* In Host level, the entire system is monitored based on the various host parameters. The user can also protect his files using *"Modified DES"* which varies from DES in the key generation part. In the network level, the system works by monitoring the packets transferred across the network. First, entire knowledge about the network is obtained. Then using the *"cross-over"* operator of Genetic Algorithm, the system creates initial population of all sorts of combination of packets. After initial population, the system obtains the intrusion condition, priorities of the parameters and intrusion level. Then the *rule sets* are framed from the initial population based on those entries whose fitness value exceeds the intrusion level. Then based on the *rule sets* the packets are monitored. In the proposed system, the user can protect his files even from the super user; the priorities of intrusion parameters and the intrusion level can be varied according to administrator's preference.

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

2.2 SYSTEM DESIGN

2.3 SYSTEM TESTING

3 CONCLUSIONS

APPENDICES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

With the advent of the World Wide Web, people nowadays not only have access to more worldwide news information than ever before, but they can also obtain it in a more timely manner. Online newspapers present breaking news on their websites in real time, and users can receive automatic notifications of them via RSS feeds.

RSS is a free way to promote a site without the need to advertise or create complicated content sharing partnerships, and an easy mechanism for the users to be informed of the latest news or web contents. However, the increasing volume, growth rate, ubiquity of access, and the unstructured nature of the contents challenge the limits of human processing capabilities. It is in such scenario where recommender systems can do their most, by scanning the space of choices, and predicting the potential usefulness of news for each particular user, without explicitly specifying needs or querying for items whose existence is unknown beforehand.

However, general common problems have not been fully solved yet, and further investigation is needed. For example, typical approaches are domain dependant. Their models are generated from information gathered within a specific domain, and cannot be easily extended and/or incorporated to other systems. Moreover, the need for further flexibility in the form of query-driven or group-oriented recommendations, and the consideration of contextual features during the recommendation processes are also unfulfilled requirements in most systems.

In this work, we present ONTOLOGY BASED WEB CRAWLER, a system that makes use of Semantic Web technologies to recommend news. The system supports different recommendation models for single and multiple users which address several recommender systems limitations. The exploitation of meta-information in the form of ontologies that describe items and user profiles in a general, portable way, along with the capability of inferring knowledge from the semantic relations defined in the ontologies, are the key aspects of the system.

Section 2 presents the architecture, functionalities and recommendation models of ONTOLOGY BASED WEB CRAWLER, referencing previous works that have more detailed explanations and evaluations, and section 3 emphasises the benefits of our proposal.

1.2 ONTOLOGY BASED WEB CRAWLER

ONTOLOGY BASED WEB CRAWLER combines textual features and collaborative information to make news suggestions. However, as opposite to previous systems, it uses a controlled and structured vocabulary to describe the news contents and user preferences [7]. For this purpose, it makes use of Semantic Web technologies. News items and user profiles are represented in terms of concepts appearing in domain ontologies, and semantic relations among those concepts are exploited to enrich the above representations, and enhance recommendations.

Figure 1 depicts how ontology-based item descriptions and user profiles are created in ONTOLOGY BASED WEB CRAWLER. Like other systems [1][10][13], news are automatic and periodically retrieved from several on-line news services via RSS feeds. The title, summary and category of the retrieved news are then annotated with concepts (classes and instances) of the system domain ontologies. Similarly to other approaches [1][2], a TF-IDF technique is applied to assign weights to the annotated concepts. A total of 17 ontologies have been used for the first version of the system. They are adaptations of the IPTC ontology[1], which contains concepts of multiple domains such as education, culture, politics, religion, science, technology, business, health, entertainment, sports, etc.

ONTOLOGY BASED WEB CRAWLER follows a client/server architecture, where users utilise a web interface to receive on-line news recommendations, and update their preferences. Thanks to the novel AJAX technology, a dynamic graphical interface allows the system to automatically store all the users' inputs, analyse their behaviour with the system update their preferences, and adjust the news recommendations in real time. As done in [8], explicit and implicit user preferences are taking into account, via manual preferences, tags and ratings, and via automatic learning from the users' actions.

Deriving benefit from the semantically annotated news items, the defined ontology-based user profiles, and the knowledge represented by the domain ontologies, a set of recommendation algorithms are executed. Specifically, ONTOLOGY BASED WEB CRAWLER offers personalised, context-aware [14], group-oriented [6], and multi-facet [4][5] recommendations.

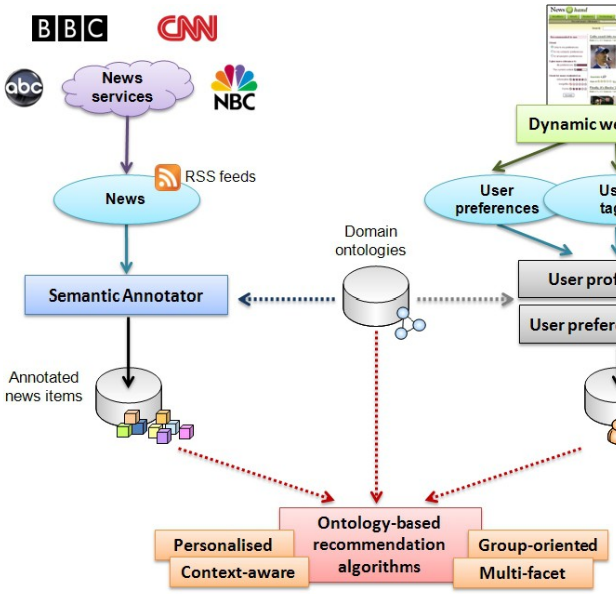
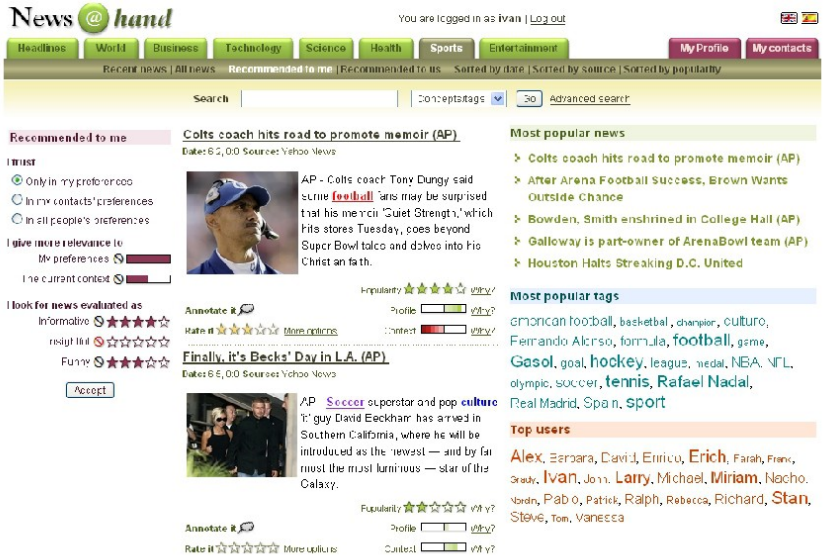


Figure 1.1. Item descriptions and user profiles acquisition in ONTOLOGY BASED WEB CRAWLER

Figure 2 shows a screenshot of a typical news page in ONTOLOGY BASED WEB CRAWLER. The news items are classified in 8 different sections: headlines, world, business, technology, science, health, sports and entertainment. When the user is not logged in the system, she can browse any of the previous sections, but the news items are listed without any personalisation criterion. She can only sort them by their publication date, source or level of popularity. On the other hand, when the user is logged in the system, recommendation and profile edition functionalities are enabled, and the user can browse the news according to her and others' semantic preferences in different ways. Like in other approaches [1][2][3], short and long term preferences are considered. As done in [9], click history is used to define the short term concepts, and similarly to [13], the resultant ranks can be adapted to the current context of interest.

In the middle of the screen, apart from its title, source, date, summary, images, and link to the full article, additional information is shown for each news item. Those terms appearing in the item that are associated to semantic annotations of the contents, the user profile and the current context are highlighted with different colours. A global rating is shown in a 5-star scale, and two coloured bars indicate the relevance of the news item for the user profile and the context. The user has the possibility of view and add comments, tags and ratings to the article, following the ideas presented in [11][12]. On the left side of the screen, the user can set the parameters she wants for single or group-oriented recommendations: the consideration of preferences of her profile, of her contacts, or of all the users, the degree of relevance than the concepts of the profile and the context should have, and multi-criteria conditions to be fulfilled by the user evaluations. Finally, on the right side of the screen, general social information such as the most popular news, the most popular tags and the top users is shown.



Recommended to me

I trust

● Only in my preferences

○ In my contacts' preferences

○ In all people's preferences

I give more relevance to

My preferences

The current context

I look for news evaluated as

Informative ★★★★☆

Insightful ☆☆☆☆☆

Funny ★★★☆☆

Colts coach hits road to promote memoir (AP)

Date: 6 2, 0:0 Source: Yahoo News

AP - Colts coach Tony Dungy said some football fans may be surprised that his memoir 'Quiet Strength,' which hits stores Tuesday, goes beyond Super Bowl tales and delves into his Christian faith.

Popularity ★★★★☆

Finally, it's Becks' Day in L.A. (AP)

Date: 6 6, 0:0 Source: Yahoo News

AP Soccer superstar and pop culture 'it' guy David Beckham has arrived in Southern California, where he will be introduced as the newest — and by far most the most luminous — star of the Galaxy.

Popularity ★★☆☆☆

Most popular news

- Colts coach hits road to promote memoir (AP)
- After Arena Football Success, Brown Wants Outside Chance
- Bowden, Smith enshrined in College Hall (AP)
- Galloway is part-owner of ArenaBowl team (AP)
- Houston Halts Streaking D.C. United

Most popular tags

american football, basketbal, champion, culture, Fernando Alonso, formula, football, game, Gasol, goal, hockey, league, medal, NBA, NFL, olympic, soccer, tennis, Rafael Nadal, Real Madrid, Spain, sport

Top users

Alex, Barbara, David, Enrico, Erich, Farah, Frank, Grady, Ivan, John, Larry, Michael, Miriam, Nacho, Nordin, Pablo, Patrick, Ralph, Rebecca, Richard, Stan, Steve, Tom, Vanessa

Figure1. 2. SCREENSHOT OF NEWS RECOMMENDATIONS IN ONTOLOGY BASED WEB CRAWLER

1.3 BENEFITS OF THE PROPOSAL

ONTOLOGY BASED WEB CRAWLER supports multiple recommendation functionalities, and addresses some of the limitations in current recommender systems:

- *Domain dependency*. The use of ontologies and Semantic Web standards to represent user profiles and news items makes it possible to easily incorporate new domains into the system, and export the obtained knowledge to other applications.
- *Restricted content analysis*. Our annotation mechanism allows the distribution and exploitation of metadata from different multimedia sources, such as texts, videos, or audios.
- *Content overspecialisation*, *cold-start*, *portfolio* and *sparsity*. The extension of user preferences and item features through ontology properties enable the detection of further co-occurrences of interests between users, and finds new interests, available for recommendations.
- *Gray sheep*. The proposed hybrid models compare user profiles at different semantic interest layers, enabling further possibilities to find relations between users.
- *Group-oriented recommendations*. The vector-based preference description facilitates the combination of multiple profiles to generate a shared profile for a group of users.
- *Context-aware recommendations*. Under the ontology-based knowledge representation, we define the notion of semantic runtime context, which we apply to provide recommendations according to the user's current interests.

1.4. PROJECT CATEGORY

Business Application (RDBMS) Expert Systems With Cloud Computing:

A relational database management system (RDBMS) has the following properties:

- Represents data in the form of tables,
- Doesn’t hard-code relationships between tables,
- Doesn’t require the user to understand its physical implementation,
- Provides information about its contents and structure in system tables,
- Can be manipulated through SQL commands,
- Supports the concept of NULL values

This project is basically comes into the category of RDBMS (Relational Data Base Management System), since we are using SQL Server as back-end engine and ASP.NET as front-end which is an object based language.

The main aim of this project is to do work on data (i.e. all daily transactions) so that we can manipulate or do various operations on the data. To do different operations on the data we need to store the data somewhere so that we can use/kept this data for future use. So, the best option which is available in the market is to store the data using RDBMS. The main purpose of using RDBMS is to store the data in normalize form, store procedures and create links b/w the same entities but which are in different tables. SQL Server is one of widely used RDBMS in the market that’s why used as backend in this project.

1.5. PROPOSED SYSTEM

There are three basic types of recommendation systems: content-based recommenders, which recommend news items based on their content, collaborative filtering recommenders, which recommend news items by means of user similarity, and hybrid recommenders, that combine the previous two approaches. In this paper we focus on content-based recommenders. We analyze two types of content-based recom-menders: traditional, which are term-based, and semantic, which are concept-based.

TF-IDF [32] is a well-known method for assigning an im-portance weight to a term in a document. Combined with the vector space model [33], TF-IDF can be used to recommend news items to a specific user. When employing user profiles that describe users' interest based on the previ-ously browsed items, these can be translated into vectors of TF-IDF weights. With a measure like cosine similarity, one can calculate how interesting a new item might be based on user profiles. For this, TF-IDF weights are computed on every term within a document. Since the last decade, methods have been developed to find key concepts in a text. A framework which implements this kind of methods is the news personalization framework called Hermes [10, 17, 34], which uses an ontology to store concepts and their relations.

This paper proposes a new method for recommending news items, i.e., CF-IDF (Concept Frequency - Inverse Doc-ument Frequency) weighting. This method is based on TF-IDF, but instead of using all the terms of a text, this method only looks at the key concepts found in this text. In order to test this new method, we implement it in Athena [21], which is an extension to the Hermes framework.

Athena is able to observe user browsing behavior and gen-erate recommendations based on this behavior. In order to recommend news items, first the user's browsing behavior is modeled. By recording a history of read news items, a profile of the user can be made. Based on this profile, it is possible to propose `new' (unread) news items that are possibly interesting to the user. Athena already implements several recommendation algorithms using various similarity measures: TF-IDF weights combined with cosine similarity, concept equivalence similarity, binary cosine similarity, Jac-card coefficient, concept neighborhood, and ranked semantic similarity [21]. However, in our current endeavors we solely focus on the existing TF-IDF recommender and the newly created CF-IDF recommender. Using the latter two recommenders we are able to compare `new' news items with user profiles. The news items that have the highest similarity with the user profile are recommended to the user.

The main contribution of this paper is twofold. Firstly, this paper proposes a new method for recommending news items, i.e., CF-IDF weighting. Secondly, we present a com-parison of the performance with the TF-IDF traditional rec-ommender through evaluation of the results of our imple-mentation, Athena, i.e., an extension to the Hermes frame-work.

The structure of this paper is as follows. Section 2 presents related work. Subsequently, Sect. 3 elaborates on the Her-mes framework and its implementation, the Hermes News Portal (HNP). Next, the Athena framework and its tra-ditional and semantic recommendation algorithms are dis-cussed in Sect. 4. Then, Sect. 5 gives an overview of our implementation of Athena in HNP. Section 6 presents the results of our evaluation, and last, conclusions and future work directions are presented in Sect. 7.

1.6. RELATED WORK

This paper focuses primarily on the semantic extension of a TF-IDF recommendation approach. First we introduce different term/concept weighting methods and then we show how these have been applied in existing recommender sys-tems.

1.6.1 TF-IDF

There are many term weighting methods available, such as for example probabilistic weighting, term frequency (TF) weighting, inverse document frequency (IDF) weighting, TF-IDF weighting, variations of TF-IDF weighting, etc. [32]. The main term weighting method that is focused on specif-ically in the work presented in this paper is the traditional TF-IDF weighting scheme. A classic approach in comparing documents is the use of TF-IDF together with the cosine similarity measure. TF-IDF is a statistical method used to determine the relative importance of a word within a docu-ment in a collection (or corpus) of documents.

Before calculating the TF-IDF values, the stop words are being filtered from the document. After stop word removal, the remaining words are stemmed by a stemmer. There are multiple stemmers available like the Krovetz [23], Lovins [24] and the Porter stemmer [29]. A stemmer reduces words back to their root word, for example the words `processor' and `processing' are reduced to `process'. The TF-IDF measure can be determined by first calculating the term frequency (TF), which indicates the importance of a term t_i within a document d_j. By computing the inverse document frequency (IDF), the general importance of the term in a set of documents can be captured. The TF-IDF weight is the multiplication of TF and IDF.

1.6.2 CF-IDF

There has been some previous work on the use of TF-IDF with concepts (similar to CF-IDF). In [5] a conceptual index-ing method based on WordNet [16], a large lexical database, is proposed. This approach represents document contents by the semantic network [37] called document semantic core. The documents are mapped on the WordNet semantic net-work and converted from a set of terms to a set of concepts. After that, the extracted concepts are weighted like in the classical index term case, using the weighting schema's TF-IDF and Okapi BM25 [31]. This method differs from ours in the detection of concepts. It does not take into account synonyms and it lacks a word sense disambiguation proce-dure present in our method [10]. Furthermore we do a more thorough comparison with TF-IDF as we perform an exten-sive evaluation including the Student t-test, the Area Under the Curve (AUC) and Cohen's Kappa coefficient. Yan and Li [38] propose a Word Sense Disambiguation (WSD) [1] method called Term Co-Occurrence Graph (TCOG), which uses WordNet to create a text representation model. In or-der to represent a text, a set of WordNet concepts with a CF-IDF weight is used. This idea is extended by taking into account also higher level concepts (e.g., concept `boat' has as higher level concepts `vessel' and `vehicle'). The authors compare their method against TF-IDF and an adapted Lesk algorithm [4]. The main difference between our approach and TCOG is the purpose of the research. While we focus on a recommendation system, TCOG is meant for text clas-sification with respect to topics. Furthermore, as is the case with [5], we perform a more thorough evaluation.

1.7 CONTENT-BASED RECOMMENDERS

In content-based approaches to news recommending, ar-ticles are recommended according to a comparison between their contents and the user profiles. The user profiles con-tain information about the users' content-based preferences. Both of these components have data-structures which are created using features extracted from text. A weighting scheme is often used to assign high weights to the most dis-criminating features/preferences, and low weights to the less informative ones.

1.7.1 Traditional Content-Based Approaches

Traditional content-based approaches are purely content-based without any semantics. Concepts get weights assigned that are obtained without semantic knowledge of underlying relations between the concepts. User interests are often measured with machine learning algorithms, like Nearest Neigh-bor or Naive Bayes.

In the traditional content-based approaches we review, ar-ticles are processed with TF-IDF by taking all terms (but the stop words) into account. The article is stored in a weighted vector of terms, and compared with a user profile by using a similarity measure. The main difference between the re-lated approaches and the method proposed in this paper, CF-IDF, is the way how we represent an article. CF-IDF considers a news item as a weighted vector of key concepts instead of terms. This makes it a more `intelligent' recom-mender: since it already knows the most important terms in the document, there are no `noise' terms which can pollute the outcome. The similarity measure used for comparing the article with the user profile is the cosine similarity.

News Dude [7] is a personal news recommending agent that uses TF-IDF in combination with the Nearest Neighbor algorithm and uses the full text of an article. News Dude first considers the short-term interests to look for similar items and if this does not return satisfiable results, long-term interests are considered.

The next related work is Daily Learner [8]. This is an adaptive news service which allows users to personalize the news to their own taste. First a user gives his preferences of what type of news he is interested in. Based on this user profile, the system then delivers those stories that best match this user's interests. A new article is processed with TF-IDF, and represented as a vector. Then this article is compared with the user profile (also a vector with TF-IDF weights), using cosine similarity. Finally, the user explicitly provides feedback using four ratings (interesting, not inter-esting, more information, already known). Short-term inter-ests are determined by analyzing the N most recently rated stories, based on the Nearest Neighbor Algorithm. Long-term interests are modeled with the Naive Bayes Classifier.

YourNews [2] is another example of a content-based news recommendation system. It is a personalized news system, which intends to increase the transparency of adapted news delivery. It allows the user to view and edit his interest pro-file. To support this, YourNews highlights the key terms in news items. The news items are represented as weighted vec-tors of terms. The weight of each term is calculated using TF-IDF. Before creating those vectors, the text is filtered from stop words and each word is reduced to its stem using a Krovetz Stemmer [23]. The user profile is represented as a weighted vector of terms extracted from the user's view his-tory and similarities between user profiles and news articles are computed using the cosine similarity measure.

Personalized Recommender System [26] (PRES) is a news personalization system that applies content-based filtering. PRES also uses the combination of TF-IDF and the cosine similarity measure. Every time a new news item is browsed, the system updates existing weights assigned to terms using a certain diminishing factor. This way PRES aims to keep the interests up-to-date, allowing changes over time. The diminishing factor is determined via experimentation.

Traditional TF-IDF recommending approaches consider the full text of the news articles. However, as the authors of [9] made a comparison with different lengths of docu-ments, the performance decreases as documents get larger. CF-IDF does not consider the full text, but only the con-cepts that exist in the knowledge base. With the semantic knowledge about the concepts it is possible to consider more than just the text at hand. The strength of the CF-IDF al-gorithm depends on the quality of the knowledge base.

1.7.2 Semantic Content-Based Approaches

Semantic content-based approaches aim to recommend news items by combining content-based techniques with do-main semantics. Weights for concepts take into account the semantic knowledge about these concepts. Each of the re-viewed recommenders has a different approach of applying the semantic knowledge provided by the ontology. The CF-IDF recommender only records the concepts to calculate weights. The approach proposed in [21], which was created in the same environment as our CF-IDF approach, calculates a similarity based on not only the concepts themselves but also based on the directly and indirectly related concepts, which are described in an ontology.

OntoSeek [20] is a content-based approach which aims to retrieve information from online yellow pages and product catalogs. It matches content with the help of the large Sen-sus [22] ontology, which comprises a simple taxonomic struc-ture of approximately 70,000 nodes. OntoSeek does not em-ploy a user profile. Instead, OntoSeek uses lexical conceptual graphs to represent queries and resource descriptions, i.e., a tree structure where nodes are nouns from the descriptions and arcs are concepts inferred by the corresponding nouns. The ontology is used for classifying items, and to match an item with a query. The user is required to disambiguate the meaning of his queries. This process is performed by the user interface that tries to identify the concept provided and asks the user to choose between potential solutions.

Quickstep [27] is one of two proposed recommendation approaches [28] for online academic publications where user profiling is based on an external research paper topic on-tology. The papers are represented using term vectors.

All the terms in the text are considered and stemmed using the Porter stemmer [29]. After this processing, the term vec-tor weights are computed using the term frequency weight-ing method (TF). The classification of papers is done using a k-Nearest Neighbor type classifier and a boosting algo-rithm. The user profile is created automatically and real-time, based on the vector representations of papers down-loaded by a user. Finally, Quickstep generates recommenda-tions by calculating the correlation (recommendation confi - dence) between the users' current field of interest and the papers which are classified to be in this field of interest. Recommendations are presented to the user sorted by the recommendation confidence. Similar to Quickstep, our ap-proach CF-IDF is maintaining the user profile by observing the user real-time, providing an up-to-date profile. Another similarity with Quickstep is the use of the vector space model to compare news items and the user profile. The most im-portant difference between our approach and Quickstep is the essence of our approach: the use of key concepts for representing news items.

The authors of [12] propose system, a news-based recommendation system which uses Semantic Web technolo-gies to describe and relate news items and user preferences in order to recommend items to a user. To represent news con-tents and user preferences the authors make use of concepts which appear in a set of domain ontologies. This project looks very similar to the Hermes News Personalization framework. Both approaches classify news items to gain key concepts, and work with a domain ontology. For rec-ommending, This News Classifier makes use of 3 different semantic methods for recommendations: content-based, collaborative filtering, and a hybrid approach [11, 13]. The latter two are not discussed since this paper focuses primarily on content-based approaches. The semantic content-based recommendation approach employs a certain similarity measure that utilizes the semantic preferences (weighted concepts gained by observing and profiling user behavior) of the user and the semantic annotations (the key concepts weighted by the classification) of an item. In our approach we follow similar procedures, i.e., we create a vector of the user profile by com-puting the CF-IDF weights of all distinct concepts found in all read news items. Subsequently, we create a vector of CF-IDF weights belonging to the concepts found in a `new' news item and we compare the two vectors using cosine similar-ity. The main difference between the semantic content-based approaches and our approach is the aim of the approach it-self. CF-IDF is mainly created to proof that a term-based recommender can be significantly improved with the help of the semantic annotations, whereas the content-based ap-proach in this project is mainly used for comparison with the hybrid recommendation approach.

CHAPTER – 2

LITERATURE REVIEW

2.1 LITERATURE REVIEW

The review is of the published literature that identifies the common characteristics of successful online communities, particularly those transferable to a Ontology platform. Identifying these characteristics will assist the Come in project in its overall aim of designing and implementing a learning platform for Ontology Based News Recommendation System.

Recommending news items or other documents based on the user's interest has attracted the attention of many re-searchers. Several adaptive Web-based news services have been developed which focus on personal recommendation of news items. These systems vary in application domain, plat-form, development methodology, levels of adaptivity, etc. We identify four categories in recommendation systems, con-tent-based, semantic-based, collaborative, and hybrid sys-tems. In this paper, we limit the discussion to content-based and semantic-based recommendation methods.

YourNews [2] is a personalized news system, that employs a content-based approach, which intends to increase the transparency of adapted news delivery by allowing the user to adapt the user profile. Another content-based approach is News Dude [3], which is a personal news recommending agent, that utilizes TF-IDF in combination with the Near-est Neighbor algorithm in order to recommend news items to the user. [4] states, supported by Singhal's findings [5], that the performance of TF-IDF, which is employed in YourNews and NewsDude, decreases as the length of the article, and the number of words, increases. In addition to this, by ignoring the semantics of a text, news items that are semantically related to the news items in the user profile, fail to be recommended by the system.

[6] provides a practical approach to measure the relatedness or similarity between RSS news items. Their method is based on the semantic relatedness between RSS items.

As in our approach, they determine the relationships between words, using WordNet [7]. Their focus is on the linguis-tic neighborhood of a word, in which general relationships as synonymy, hyponymy, and meronymy between words are considered. The difference with our approach is that we make use of an ontology. Besides the general relationships between words, the ontology covers specific relationships like is-competitor-of, has-product, etc. Despite this difference, their method is applicable in our context, and therefore we will compare both approaches.

In [8] ontological user profiling is employed for recommending academic research papers. While is-a relationships are rich in semantics, we find this approach limited, as it fails to consider other types of concept relationships. The authors propose a classification algorithm, based on the k-Nearest Neighbor classifier, that assigns topics to papers. In our approach, GATE [9] is employed to classify the content of an article by using several language processing techniques. This enables the system to not only recommend full articles, but also possibly recommend a snippet of an article. An-other difference lies in the construction of the user profile, as in [8], the user can adjust the profile. However, as [2] explains, adjusting the user profile might harm the quality of the recommendations, so in our approach the user is not allowed to change the profile. Recommendations are made by combining collaborative filtering techniques with limited semantic-based recommendations, that only employ is-a re-lations, while our system solely employs semantic-based rec-ommendation techniques that utilize more types of relation-ships between concepts.

User Profile Construction

Recommending news items starts with building a user pro-file. A user profile can be defined by keeping track of which articles the user has read so far. Those articles will provide us with information about the user’s interests. The user pro-file is constructed in different ways. For concept equivalence, binary cosine, and Jaccard, the profile is a set of concepts from the articles the user has read. The semantic related-ness approach creates a vector with the distinct concepts from the user profile and assigns a weight to each concept. The ranked recommendation method also uses a vector of distinct concepts from the read articles and assigns a rank to each concept. The difference in user profile construction between the latter two approaches, is the method used to compute the corresponding weights.

Semantic-Based Recommendation

In traditional forms of text comparison, all words in the text are considered. In addition to this, there is no relation between different words. For instance, it is not possible to determine the relation between Google and Microsoft. But a user who is interested in news regarding his stocks in Google, might also be interested in news about Microsoft, because it is a competitor of Google. Using an ontology that covers those relations might therefore be useful in recommending new articles. To illustrate how we accomplished this, we will first discuss a few simple methods and then conclude with a complex method.

According to Cantador, I. BelloginA. & Castells, P. in 2008:

News@hand is a news recommender system that makes use of semantic technologies to provide several on-line news recommendation services. News contents and user preferences are described in terms of concepts appearing in a set of domain ontologies. Based on the similarities between item descriptions and user profiles, and the semantic relations between concepts, content-based and collaborative recommendation models are supported by the system. In this paper, we evaluate a model that personalizes the order in which news articles are shown to the user according to his long-term interest profile, and other model that reorders the news items lists taking into account the current semantic context of interest of the user. The combination of those models is investigated showing significant improvements on the experimental tasks performed.

According to Stuart E. Middleton David C. :

Tools for filtering the World Wide Web exist, but they are hampered by the difficulty of capturing user preferences in such a dynamic environment. We explore the acquisition of user profiles by unobtrusive monitoring of browsing behaviour and application of supervised machine-learning techniques coupled with an ontological representation to extract user preferences. A multi-class approach to paper classification is used, allowing the paper topic taxonomy to be utilised during profile construction. The Quickstep recommender system is presented and two empirical studies evaluate it in a real work setting, measuring the effectiveness of using a hierarchical topic ontology compared with an extendable flat list .

According to Bhogalb, A. Macfarlanea, & P. Smitha

This paper examines the meaning of context in relation to ontologybased query expansion and contains a review of query expansion approaches. The various query expansion approaches include relevance feedback, corpus dependent knowledge models and corpus independent knowledge models. Case studies detailing query expansion using domain-specific and domain-independent ontologies are also included. The penultimate section attempts to synthesise the information obtained from the review and provide success factors in using an ontology for query expansion. Finally the area of further research in applying context from an ontology to query expansion within a newswire domain is described.

According to Iván Cantador, Alejandro Bellogín and Pablo Castells

We present News@hand, a news recommender system which applies semantic-based technologies to describe and relate news contents and user preferences in order to produce enhanced recommendations. The exploitation of conceptual information describing contents and user profiles, along with the capability of inferring knowledge from the semantic relations defined in the ontologies, enabling different content-based collaborative recommendation models, are the key distinctive aspects of the system. The multi-domain portability, the multi-media source applicability, and addressing of some limitations of current recommender systems are the main benefits of our proposed approach.

Wang and Kong (2007) is a personalised recommender system which tries to limit the problems of collaborative recommender systems by ontologically using semantic information from the categorical characteristics of an item. The similarities between user pairs is calculated by a weighted mean method that calculates three similarity measures: the similarity of user evaluation histories (using the Pearson correlation coefficient on usage information of the system in terms of a user-item evaluation data matrix); the similarity of these user's demographic data (calculated with a weighted mean); and the users similarity in interest or preference based on the semantic similarities of the items retrieved and/or evaluated. At the same time, the system incorporates an offline-user cluster mechanism to limit the scalability problem.

Khosravi, Farsani and **Nematbakhsh** (2006) suggest a methodology for personalised recommendations in the context of e-commerce. This is a procedure to recommend products to potential clients. The proposed algorithm is based on modelling information on products and users with OWL (Ontology Web Language). The process starts by classifying the products and consumers with OWL, which enables the analysis of product-client similarity. In a second phase, active consumers are selected, keeping in mind previous recommendations (the system does not make recommendations to a client if the number of previous recommendation does not pass a threshold). The product and client classification is used to create a matrix of product-client evaluations. The algorithm recommends some products from each class within the classes of products based on the number of evaluations in the matrix.

Another model used in the field of e-commerce is one presented by Ziegler, Lausen and Schmidt-Thieme (2004). The system is based on the collaborative-recommender paradigm through content (Pazzani, 1999) using a product taxonomy from which the user profiles are defined (without users needing to provide explicit valuations). The active user profile is used to discover users with similar interests, whose valuations help the system generate recommendations.

Jung and collaborators (2005) propose a recommender system based on personal information which they claim suits the Semantic Web context.. The model is based on the representation of Web services and user profiles with RDF triples (Resource Description Framework). Each company wanting to provide Web services registers its data in the information repository, where the system converts the data into documents in RDF format. The search module extracts the repository's information and sends it to the document retrieval agent. The agent accesses the space with the corresponding name and retrieves the RDF documents relative to the required Web services. These documents are sent to the information integration agent, where they are merged in a single RDF document containing the relevant information. Finally, the information retrieval agent extracts the most relevant RDF triples in accordance with the user profile and offers the user the stored objects that coincide with those triples.

Other systems are defined with decentralised structures like P2P networks. For example, the model presented by Díaz-Avilés (2005), where the information is not available in a centralised repository, but in each of the network's components. The items or objects are modelled through a common ontology that uses all members of the network. The selection of the network's components is done dynamically and the recommendations are generated using a nearest neighbour-based recommender algorithm that is locally executed in each of the network components.

An original approximation is Cantador´s and Castells´ (2006) proposal to develop a multi-layer semantic social network model that can define the system from different perspectives, all from common interests shared by the network's members. From a series of generated user profiles using ontological concepts, and keeping in mind their common preferences, the system is capable of marking out the domain's different concept groups. From these groups, we can identify a set of users with similar interests that interrelate at different semantic levels (according to their preferences). This method allows us to find implicit social networks that may help to define both content-based and collaborative-based recommender systems.

2.2 PROJECT CATEGORY

This project is a "Web Application”. It uses the concept of OOPs, RDBMS and Multimedia for Conferencing and Internet technologies.

The Project aims at creating a website using ASP.NET and SQL Server for a web based platform. It uses some features of .NET, to speed up a few sections of this project.

I. The User’s layer: tools and languages used → .NET (ASP.NET,VB.NET), HTML, DHTML, and JavaScript

Editor: Visual Studio 2005

There is a front end which is web-pages which are displayed before the user and user directly interact with the software. Web pages are developed by using various design tools like Flash, Photo Shop etc and documentation languages like HTML and DHTML.

II. The Presentation layer: tools and languages used →ASP.NET

Editor: Visual Studio 2005, Dreamweaver

At this layer all the services related with presentations like format check, mailing interface, parsing, deployment descriptor etc. this layer is developed in asp.net.

III. The Integration Level: tool/technology used →SQL Server

This tier is dedicated for database connectivity and connection management with the database. This is tier is developed in .net and the database software to be used sql server.

IV. The Data Resources level: [SQL Server 2005]

At the back end, there is a data base which maintains data. This is to be developed in SQL server. In this server tables are created and all primary key –foreign key constraints are implemented. The data base follows all the integrity constraints.

- □ Platform used: Microsoft Windows 2000 Advanced Server

Features of Windows 2000 Advanced Server which may contribute to this project

- This is server edition and contains all the features of server.
- Easy to use and configure different components of the system.

- Provide good base to develop enterprise software.
- Provide multi-user facilities to distribute and reintegrate different components of the project easily.

□ Architecture implemented: .Net Framework 3.0

Justification behind the recommendation of .Net

□ *Enterprise nature:* The project is web-based and Enterprise applications can provide services to end-users via the internet or private networks.

□ *Diversity of information needs:* In an enterprise, information is created and consumed by various users in a number of different forms, depending of specific needs. It is vary common to find that each business activity may process the same information in a different form.

□ *Complexity of business processes:* Processes involved in this project need complex information capture, processing, and sharing. Very often, we encounter complex logic to capture and process information. This leads to complex technical and architectural requirements for building enterprise applications.

□ *Diversity of applications:* Due to the complex nature of enterprise business processes, it is common to find that an enterprise consists of a large number of applications each built at various times to fulfill different needs of various business processes. This commonly leads to the presence of applications built using different architectures and technologies. One of the challenging need of our project is to make such applications that can talk to each other so that business processes can be implemented seamlessly.

This project is designed to meet the requirements of Cooperative Societies. For designing the system we have used simple data flow diagrams.

Overall the project teaches us the essential skills like:

❑ Using system analysis and design techniques like data flow diagram in designing the system.

❑ Understanding the database handling and query processing using SQL Server._

INTRODUCTION TO .NET

What is .NET?

- A vision of how information technology will evolve
- A platform that supports the vision
- A business model of software as a service

1. A Vision.

- Web sites will be joined by Web services
- New smart devices will join the PC
- User interfaces will become more adaptable and customizable
- Enabled by Web standards

2. A Platform.

- The .NET Framework
- ASP.NET
- .NET Enterprise Servers
    - Database, Messaging, Integration, Commerce, Proxy, Security, Mobility, Content Management
- .NET Building Block Services
    - Passport
    - .NET My Services (“Appin”)
- Goal: make it incredibility easy to build powerful Web applications and Web services

3. A business model.

- Software as a service
- Subscription-based services
- Application hosting, e.g. bCentral

Interoperability: Web languages and protocols must be compatible with one another independent of hardware and software.

Evolution: The Web must be able to accommodate future technologies. Encourages simplicity, modularity and extensibility.

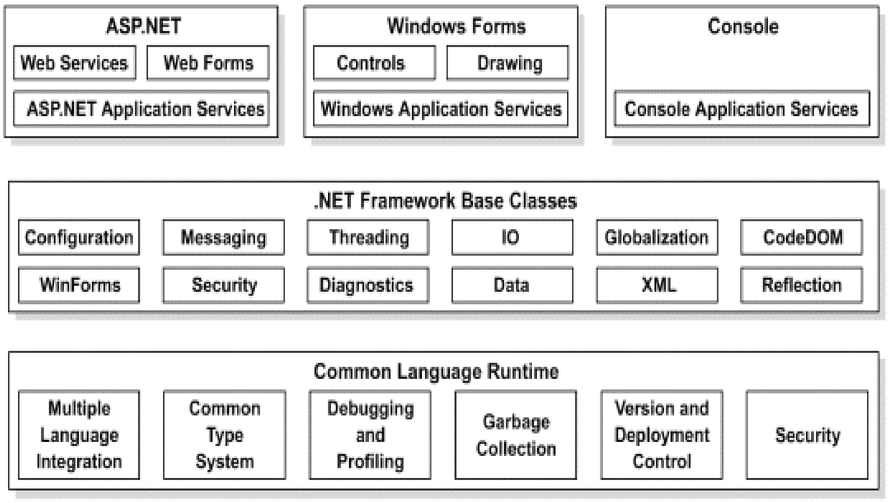
Decentralization: Facilitates Scalability and Robustness.

Web Services

- A programmable application component accessible via standard Web protocols
- The center of the .NET architecture

- Exposes functionality over the Web
- Built on existing and emerging standards are HTTP, XML, SOAP, UDDI, WSDL, …

<u>The .NET Framework</u>



What is the .NET Framework?

- A set of technologies for developing and using components to create:
    - Web Forms
    - Web Services
    - Windows Applications
- Supports the software lifecycle
    - Development
    - Debugging
    - Deployment
    - Maintenance

Application Architectures

- The Different types of Applications may vary from single-tier desktop applications (applications that follow the single-tier architecture) to multi-tier applications (applications that follow the two-, three, or n-tier architecture)
- Single-tier architecture

  - A single executable file handles all functions relating to the user, business, and data service layers.
- Two-tier architecture
  - Divides an application into the following two components:
    - ❖ Client
    - ❖ Server
- Three-tier architecture
  - All the three service layers reside separately, either on the same machine or on different machines.
- n-tier architecture
  - Uses business objects for handling business rules and data access.
  - Has multiple servers handling business services.

The .NET Initiative

- The introduction of the Internet and its rapid growth in the recent past has led to the development of a number of new Technologies.
- One of the most important requirements of such applications is the ability to interchange information across platforms and to benefit from the functionality provided by other applications.
- In the current scenario, although applications serve organization-specific requirements, they are not interoperable. Microsoft has introduced the .NET initiative with the intention of bridging the gap in interoperability between applications.

- The .NET initiative offers a complete suite for developing and deploying applications, which consists of the following:
    - NET products: Microsoft has already introduced Visual Studio .NET, which is a tool for developing NET applications by using programming languages such as Visual Basic, C#, and Visual C++.
    - NET services: Microsoft is coming up with its own set of Web services, known as My Services. These services are based on the Microsoft Passport Authentication service, the same service that is used in Hotmail.

Explanation of the .NET Framework

- Is a collection of services and classes?
- Exists as a layer between .NET applications and the underlying operating system.
- Encapsulates much of the functionality, such as debugging and security services.
- The following figure depicts the components of the .NET Framework:

The .NET Framework Base Classes or the .NET Class Framework

- Consists of a class library that works with any .NET language, such as Visual Basic .NET and C#.
- Provides classes that can be used in the code to accomplish a range of common programming tasks.
- Comprises

  - Namespaces: Namespaces help you to create logical groups of related classes and interfaces that can be used by any language targeting the .NET Framework.
  - Assembly: An assembly is a single deployable unit that contains all the information about the implementation of classes, structures, and interfaces.
- The Common Language Runtime
  - Provides functionality such as exception handling, security, debugging, and versioning support to any language that targets it.
  - Can host a variety of languages and offer a common set of tools across these languages, ensuring interoperability between the codes.

- The following diagram depicts the process of compilation and execution of a .NET application:

- Provides the following features:
  - Automatic memory management
  - Standard type system
  - Language interoperability
  - Platform independence
  - Security management
  - Type safety

Advantages of the .NET Framework

- Some advantages of the .NET Framework are:
  - Consistent programming model
  - Multi-platform applications

- Multi-language integration
- Automatic resource management
- Ease of deployment

ADO.NET

- Is a model used by Visual Basic .NET applications to communicate with a database for retrieving, accessing, and updating data?
- Uses a structured process flow to interact with a database.

ADO .NET Data Access

Most applications need data access at one point of time making it a crucial component when working with applications. Data access is making the application interact with a database, where all the data is stored. Different applications have different requirements for database access. ASP.NET uses ADO.NET (Active X Data Object) as it's data access and manipulation protocol, which also enables us to work with data on the Internet. Let's take a look why ADO.NET came into picture replacing ADO.

Evolution of ADO.NET

The first data access model, DAO (data access model) was created for local databases with the built-in Jet engine which had performance and functionality issues. Next came RDO (Remote Data Object) and ADO (Active Data Object) which were designed for Client Server architectures but soon ADO took over RDO. ADO was a good architecture but as the language changes so is the technology within it. With ADO, all the data is contained in a record set object which had problems when implemented on the network (Internet) and penetrating firewalls. ADO was a connected data access which means that when a connection to the database is established the connection remains open until the application is closed, which raises concerns about database security and network traffic. And also as databases are becoming increasingly important and as they are serving more people a connected data access model makes us think about its use. For example, an application with connected data access may do well when connected to two clients, the same may do poorly when connected to 10 and might be unusable when connected to 100 or more. Also, open database connections use system resources to a maximum extent making the system performance less effective.

Why ADO.NET?

To cope up with some of the problems mentioned above, ADO.NET came into existence. ADO.NET addresses the above mentioned problems by maintaining a disconnected database access model which means that when an application interacts with the database the connection is opened to serve the request of the application and is closed as soon as the request is completed. Likewise if a database is updated, the connection is opened long enough to complete the Update operation and is closed. By keeping connections open for only a minimum period of time ADO.NET conserves system resources and provides maximum security for databases and also has less impact on system performance. Also, ADO.NET when interacting with database uses XML by converting all the data into XML and using it for database related operations making them more efficient.

Features of ADO.NET

- Disconnected data architecture — Applications connect to the database only while retrieving and updating data.
- Data cached in datasets — ADO.NET is based on a disconnected data structure. Therefore, the data is retrieved and stored in datasets.
- Data transfer in XML format — ADO.NET uses XML for transferring information from a database into a dataset and from the dataset to another component.
- Interaction with the database is done through data commands.

ADO.NET Object Model Key Components of the ADO.NET Model

- **Data Provider**
  - Is used for connecting to a database, retrieving data, and storing the data.
- **Is of two types:**
  - OLE DB data provider
  - SQL Server data provider

Components of a Data Provider

- **<u>Connection</u>**
  - Used to establish a connection with a data source
  - Some commonly used properties and methods:
    - ❖ ConnectionString property
    - ❖ Open()method
    - ❖ Close()method
    - ❖ State property
- **<u>Data adapter</u>**
  - Creates a dataset and updates the database.
  - Handles data transfer between the database and the dataset through its properties and methods.
  - Displays the data through the process of table mapping.
  - Are of two types:
    - ❖ SqlDataAdapter
    - ❖ OleDbDataAdapter

- **<u>Data command</u>**

- Is a SQL statement or a stored procedure that is used to retrieve, insert, delete, or modify data from a data source.
- Is an object of the OleDbCommand or SQLCommand class.

35 17 **Data reader**

- Is used to retrieve data from a data source in a read-only and forward-only mode.
- Stores a single row at a time in the memory.
- Commonly used methods:
    - ❖ Read()
    - ❖ Close()
    - ❖ NextResult()

35 17 **Dataset**

- Is a disconnected, cached set of records that are retrieved from a database?
- Is present as a DataSet class in the System.Data namespace.
- Has its own object model.

TOOL/PLATFORM, HARDWARE AND SOFTWARE REQUIREMENT SPECIFICATION

HARDWARE SPECIFICATIONS

The hardware on which the project developed has the following configuration:

Main Memory	:	256 MB RAM
Microprocessor	:	P-IV and Higher versions
Hard Disk Drive	:	80 GB
Printer	:	HP- DeskJet
Disk Drives	:	1.44 MB FDD
CD ROM	:	HP

SOFTWARE SPECIFICATIONS

Operating System	:	**Windows XP**
RDBMS	:	**SQL SERVER**
Front-End-Tool	:	**ASP.** NET

FRONT END:

- C# .NET is one of the languages that are directed towards meeting the objectives of the .NET initiative of creating distributed applications.
- C# .NET is a powerful object-oriented language that provides features such as abstraction, encapsulation, inheritance, and polymorphism.

Features of Visual Basic .NET

$^{35}_{17}$ Some of the key features of Visual Basic .NET are as follows:

- Inheritance
- Constructors and destructors
- Overloading
- Overriding
- Structured exception handling
- Multithreading

2.3 ALGORITHM IMPLEMENTED

For instance when you read something about concept c_1 which is related to concept c_2 and concept c_3 you increase not only your knowledge in concept c_1 but also in the other two concepts.

Even though it is used in a different research field (adaptive hypermedia), the main idea can be applied also here. Each concept is assigned a value, this value we call the rank. For example, when a user reads about Google, he might also be interested in its competitors, like Yahoo!, but also in news about its CEO, Eric Schmidt. Both are considered to be in direct relation to the concept Google. Therefore we increase the rank for Google, Yahoo!, and Eric Schmidt. Unrelated concepts, i.e., concepts that are not directly connected to the current concept, also need to be addressed. This means, if a user profile consists of concepts c_1 and c_2, and the next article the user reads, contains concept c_3, which is directly related to c_1, but not related to c_2, we increase the rank of c_1, and decrease the rank of c_2. By decreasing the rank for such a concept we make the user profile adaptive to the user’s main interest.

The set of related keywords to concept c_i is defined as:

$$r(c_i) = \left\{ c^i_1, c^i_2, \cdot\cdot\cdot, c^i_k \right\} . \qquad (13)$$

R is described as the union of all related concepts to the concepts in the user profile:

$$R = \bigcup_{u_i \in U} r(u_i) . \qquad (14)$$

And finally U_R is defined as the set of all concepts and cor-responding related concepts, this is called the extended user profile:

$$U_R = U \cup R . \qquad (15)$$

The extended user profile is used in order to be able to increase the interest of the user in certain concepts that are not in the user profile, but are related to the concepts in the user profile.

To calculate the final ranks for each concept, we organize the concepts in a matrix. This is done because we have to assign a rank to each concept in the extended user profile for each concept the user has read about. Reading about concept c_1 increases its value with 1.0. If concept c_2 is di-rectly related to concept c_1, then its value is increased with 0.5. If there is a concept, concept c_3, in the extended pro-file which is neither equal to concept c_1 nor is it related to concept c_1, its value is decreased with 0.1. These constants were determined by experimenting with values ranging from 0 to 1 with a step of 0.1. Applying this procedure results in a matrix with rank values. The columns contain the items from the extended user profile (U_R) and the rows contain the items from the user profile (U). Table 1 shows a rank matrix, where $e_i \in U_R$ and $u_i \in U$. Summing the values of the cells in a column of the matrix, and repeating this process for each column, results in a vector with the final ranks for each concept, in the extended user profile.

Table 1: Rank matrix

	e_1	e_2	…	e_q
u1	r11	r12	…	r11
u2	r21	r22	…	r2q
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
u_m	rm1	rm2	…	rmq

The user might have read one or more articles about a con-cept. Logically, the user is presumed to be more interested in concepts that are found in several articles. The number of articles the user has read about concept u_i, is called the weight w_i,

$$W = \{w_1, w_2, \cdot\cdot\cdot, w_m\}\ . \qquad (16)$$

Now we can calculate the value for each cell in the above matrix. This is done as follows:

$$\text{ri,j} \;=\; +0^{+1}.\quad \text{if}^{\text{if}}\, e^{e\,j}_{j}\ {}^{=}\!=u^{u\,i}_{i},$$

$$\text{wi} \quad \mathbf{8} \quad 5^{0} \quad e_j \qquad r(u_i) \quad . \qquad (17)$$

$$0. \quad \text{otherwise}$$

$$\times\ {}^{<} \quad 1 \quad {}^{6} \qquad \in$$

$$\mathbf{:} -$$

The final rank for each concept from the extended user pro-file, can be computed by taking the sum of the values of the corresponding column in the matrix:

$$\text{Rank}(ej\,) = X_{i=1}^{m} rij \;. \qquad (18)$$

Those sums are stored in a vector V_U . Each concept in the extended user profile now has a rank. Before we can compare the user profile with an unread news article, we need to ensure that the range of the ranks is [0,1]. The normalization is done as follows:

$$V_U\,[v_i] = \frac{v_i - \min(v_u)}{\max(v_u) - \min(v_u)}\,, \qquad (19)$$

where $v_i \in V_U$ and $v_u \in V_U$. With this normalization we can compare the extended user profile to a new article that needs to be classified. The new article consists of a set of concepts, specified as A:

$$A = \{a_1, a_2, \cdot\cdot\cdot, a_t\}\;. \qquad (20)$$

For this article we define a vector containing the ranks. This vector is defined as V_A:

$$V_A = (s_1, s_2, \cdot\cdot\cdot, s_t)\,, \qquad (21)$$

$$si = \begin{cases} \text{Rank}(ei) & \text{if } e_i \in A \\ 0 & \text{if } e_i \notin A \end{cases} \;. \qquad (22)$$

Each concept from the extended user profile that appears in the article is assigned the same rank as the one in V_U . The remaining concepts are assigned zero. Concepts appearing in the article but not in the profile are ignored. In the current work we assume that all concepts found in a news item are equally important.

To compare the article with the user profile we propose to compute the extent to which the article fits the profile by dividing the sum of the ranks of concepts in the article by the sum of the ranks of the concepts in the user profile:

$$\text{Similarity}(V_A, V_U) = \frac{\sum_{v_a \in V_A} v_a}{\sum_{v_u \in V_U} v_u}. \qquad (23)$$

The article with the highest similarity measure fits best with the user profile. The cut-off value for news item interestingness was fixed to 0.5, after experimenting

2.4 INPUT AND OUTPUT OF THE PROJECT

Input of the project

- User profiles
- Login form
- Feedback
- Change password
- Adding RSS feeds
- Admin Login form
- News Categories
- Personalized Keywords

Output of the project

- User’s details
- Latest news
- Personilized news
- Generalized News
- Feedback details

CHAPTER – 3

SYSTEM STUDY AND ANALYSIS

3.1 PRELIMINARY INVESTIGATION

System development, a process consisting of two major steps of system analysis and design, start when management or sometimes system development personnel feel that a new system or an improvement in the existing system is required. The system development life cycle is classically thought of as the set of activities that analysts, designers and users carry out to develop and implement an information system. The system development life cycle consists of the following activities:

- Preliminary investigation
- Determination of system requirements
- Design of system
- Development of software
- System testing
- Implementation, evaluation, and maintenance

A request to take assistance from information system can be made for many reasons, but in each case someone in the organization initiates the request is made, the first system activity the preliminary investigation begins. This activity has three parts:

1) Request clarification
2) Feasibility study
3) Request approval

Request clarification: Many requests from employees and users in the organizations are not clearly defined, therefore it becomes necessary that project request must be examined and clarified properly before considering systems investigation.[6]

3.2 SYSTEM DEVELOPMENT LIFE CYCLE

The systems development life cycle (SDLC) describes a set of steps that produces a new computer information system. The SDLC is a problem-solving process. Each step in the process delineates a number of activities. Performing these activities in the order prescribed by the SDLC will bring about a solution to the business situation. The SDLC process consists of the following phases:

1. **Preliminary investigation**—the problem is defined and investigated.
2. **Requirements definition**—the specifics of the current system as well as the requirements of the proposed new system are studied and defined.
3. **Systems design**—a general design is developed with the purpose of planning for the construction of the new system.
4. **Systems development**—the new system is created.
5. **System installation**—the current operation is converted to run on the new system.
6. **Systems evaluation and monitoring**—the newly operational system is evaluated and monitored for

the purpose of enhancing its performance and adding value to its functions.

7. Looping back from a later phase to an earlier one may occur if the need arises.

Each phase has a distinct set of unique development activities. Some of these activities may span more than one phase. The management activity tends to be similar among all phases.

The SDLC is not standardized and may be unique to a given organization. In other words, the names and number of phases may differ from one SDLC to the next. However, the SDLC discussed here is, to a large extent, representative of what is typically adopted by organizations.

At each phase certain activities are performed; the results of these activities are documented in a report identified with that phase. Management reviews the results of the phase and determines if the project is to proceed to the next phase.

The first two phases of the SDLC process constitute the systems-analysis function of a business situation. The following discussion will concentrate on phase one (Preliminary Investigation) and phase two (Requirements Definition) of the outlined SDLC process.

3.2.1 Defining a System

A collection of components that work together to realize some objective forms a system. Basically there are three major components in every system, namely input, processing and output.

In a system the different components are connected with each other and they are interdependent. For example, human body represents a complete natural system. We are also bound by many national systems such as political system, economic system, educational system and so forth. The objective of the system demands that some output is produced as a result of processing the suitable inputs.

3.2.2 System Life Cycle

System life cycle is an organizational process of developing and maintaining systems. It helps in establishing a system project plan, because it gives overall list of processes and sub-processes required for developing a system.

System development life cycle means combination of various activities. In other words we can say that various activities put together are referred as system development life cycle. In the System Analysis and Design terminology, the system development life cycle means software development life cycle.

Following are the different phases of software development cycle:

- System study

- Feasibility study
- System analysis
- System design
- Coding
- Testing
- Implementation
- Maintenance

3.2.3 The Different Phases Of Software Development Life Cycle Are Shown Below.

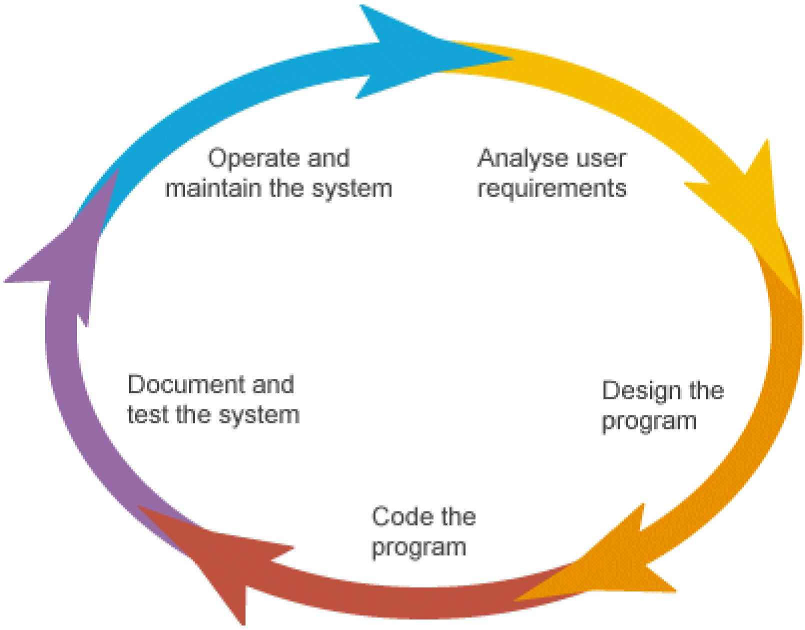
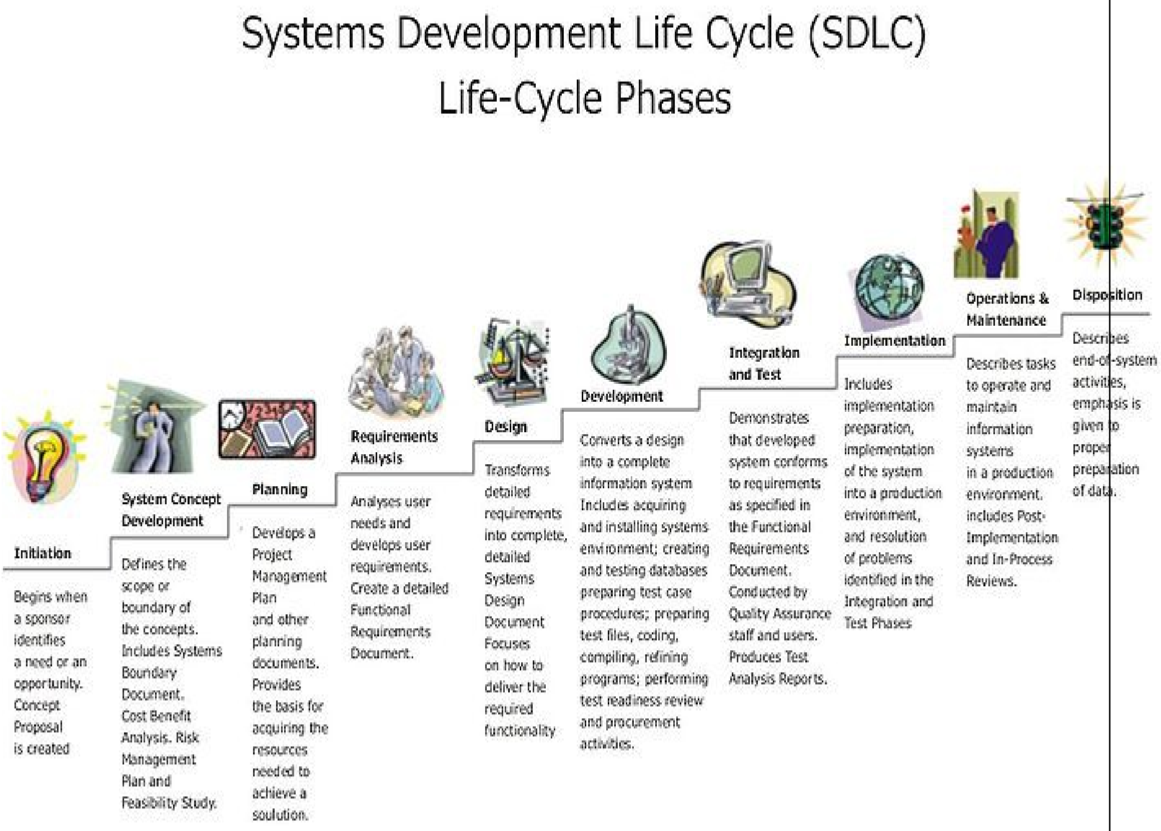

Operate and maintain the system
Analyse user requirements
Document and test the system
Design the program
Code the program

Figure 3.1: SHOWING GENERAL LIFE CYCLE PROCESS AND PERCENTAGE OF TIME DEVOTED

The goal of system analysis is to determine where the problem is in an attempt to fix the system. This step involves breaking down the system in different pieces to analyze the situation, analyzing project goals, breaking down what needs to be created and attempting to engage users so that definite requirements can be defined.

Requirements analysis sometimes requires individuals/teams from client as well as service provider sides to get detailed and accurate requirements; often there has to be a lot of communication to and from to understand these requirements. Requirement gathering is the most crucial aspect as many times communication gaps arise in this phase and this leads to validation errors and bugs in the software program.[7]



Initiation

Begins when a sponsor identifies a need or an opportunity. Concept Proposal is created

System Concept Development

Defines the scope or boundary of the concepts. Includes Systems Boundary Document. Cost Benefit Analysis. Risk Management Plan and Feasibility Study.

Planning

Develops a Project Management Plan and other planning documents. Provides the basis for acquiring the resources needed to achieve a soulution.

Requirements Analysis

Analyses user needs and develops user requirements. Create a detailed Functional Requirements Document.

Design

Transforms detailed requirements into complete, detailed Systems Design Document Focuses on how to deliver the required functionality

Development

Converts a design into a complete information system Includes acquiring and installing systems environment; creating and testing databases preparing test case procedures; preparing test files, coding, compiling, refining programs; performing test readiness review and procurement activities.

Integration and Test

Demonstrates that developed system conforms to requirements as specified in the Functional Requirements Document. Conducted by Quality Assurance staff and users. Produces Test Analysis Reports.

Implementation

Includes implementation preparation, implementation of the system into a production environment, and resolution of problems identified in the Integration and Test Phases

Operations & Maintenance

Describes tasks to operate and maintain information systems in a production environment. includes Post-Implementation and In-Process Reviews.

Disposition

Describes end-of-system activities, emphasis is given to proper preparation of data.

.

Figure 3.2 System Development Life Cycles

Before designing any system it is important that the nature of the business and the way it currently operates are clearly understood. The detailed examination provides the specific data required during designing in order to ensure that all the client's requirements are fulfilled. The investigation or the study conducted during the analysis phase is largely based on the feasibility study. Rather it would not be wrong to say that the analysis and feasibility phases overlap. High-level analysis begins during the feasibility study. Though analysis is represented as one phase of the system development life cycle (SDLC), this is not true. Analysis begins with system initialization and continues until its maintenance. Even after successful implementation of the system, analysis may play its role for periodic maintenance and up gradation of the system.

One of the main causes of project failures is inadequate understanding, and one of the main causes of inadequate understanding of the requirements is the poor planning of system analysis.

Analysis requires us to recall the objectives of the project and consider following three questions:

- ✓ What type of information is required?
- ✓ What are the constraints on the investigation?
- ✓ What are the potential problems that may make the task more difficult?

The three major parts of the system are:

➢ Providing Information:

The system is effectively used to provide large variety of information to the interested customer. The major purpose of the site is to easily provide access to records of various automobile such as car with quick update to latest modifications in the records. This thing is not at all possible in printed material, which are updated only once a few weeks. It also gives information about the general usage of the system for first time visitors. The system itself works as a information provider for automobile organization.

3.3 FEASIBILITY STUDY

The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution. The analyst conducting the study gathers information using a variety of methods, the most popular of which are:

- Interviewing users, members, students, and office bearers.
- Developing and administering questionnaires to interested stakeholders, such as potential users of the information system.
- Observing or monitoring users of the current system to determine their needs as well as their satisfaction and dissatisfaction with the current system.

- Collecting, examining, and analyzing documents, reports, layouts, procedures, manuals, and any other documentation relating to the operations of the current system.
- Modeling, observing, and simulating the work activities of the current system.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components. These components are:

1. **Economic feasibility**—The economic viability of the proposed system. The proposed project's costs and benefits are evaluated. Tangible costs include fixed and variable costs, while tangible benefits include cost savings, increased revenue, and increased profit. A project is approved only if it covers its cost in a given period of time. However, a project may be approved only on its intangible benefits such as those relating to government regulations, the image of the organization, or similar considerations.
2. **Technical feasibility**—the possibility that the organization has or can procure the necessary resources. This is demonstrated if the needed hardware and software are available in the marketplace or can be developed by the time of implementation.

3. **Operational feasibility**—the ability, desire, and willingness of the stakeholders to use, support, and operate the proposed computer information system. The stakeholders include management, employees, users, and suppliers. The stakeholders are interested in systems that are easy to operate, make few, if any, errors, produce the desired information, and fall within the objectives of the organization.

Steps in feasibility Analysis are:

1. Identify deficiency by pinpointing, missing functions, unsatisfactory performance, Excessive cost of operations.
2. Set goals to remove these deficiencies.
3. Goals must be quantified, realizable within the constraints of an organization broken down into sub goals agreeable to all concerned.
4. Set goals not only to remove deficiencies but also to effectively meet competition..

Broad solutions will consist of:

1. Specifications of information to be made available by the system.

2. Description of what will be done manually and what the computer will do.

3. Specification of new computing equipment needed or specification of
expansion of an existing computer.

COST AND BENEFIT ANALYSIS

Developing an IT application is an investment. Since after developing that application it provides the organization with profits. Profits can be monetary or in the form of an improved working environment. However, it carries risks, because in some cases an estimate can be wrong. And the project might not actually turn out to be beneficial.

A cost benefit analysis is done to determine how well, or how poorly, a planned action will turn out. Although a cost benefit analysis can be used for almost anything, it is most commonly done on financial questions. Since the cost benefit analysis relies on the addition of positive factors and the subtraction of negative ones to determine a net result, it is also known as running the numbers

Cost benefit analysis helps to give management a picture of the cost, benefits and risks. It usually involves comparing alternate investments.

Cost benefit determines the benefits and savings that are expected from the system and compares them with the expected costs.

In performing cost and benefit analysis it is important to identify cost and benefits factors. Cost and benefits can be categorized into the following categories:

1. **Development Costs –** Development costs is the costs that are incurred during the development of the system. It is one time investment.
2. **Operating Costs –** Operating Costs are the expenses required for the day to day running of the system. Examples of Operating Costs are Wages, Supplies and Overheads.
3. **Hardware/Software Costs –** It includes the cost of purchasing or leasing of computers and it's peripherals. Software costs involves required S/W costs.
4. **Personnel Costs –** It is the money spent on the people involved in the development of the system.
5. **Facility Costs –** Expenses that are incurred during the preparation of the physical site where the system will be operational. These can be wiring, flooring, acoustics, lightning, and air-conditioning.
6. **Supply Costs –** These are variable costs that are very proportionately with the amount of use of paper, ribbons, disks, and the like.

- **BENEFITS**

We can define benefits as

Profit or Benefit = Income – Costs

Benefits can be accrued by:

- Increasing income, or

- Decreasing costs, or both

Technical feasibility

Today, very little is technically impossible. Consequently, technical feasibility looks at what is practical and reasonable. Technical feasibility addresses three major issues:

1. Is the proposed technology or solution practical?

2. Do we currently possess the necessary technology?

3. Do we possess the necessary technical expertise, and is the schedule reasonable?

Is the Proposed Technology or Solution Practical?

The technology for any defined solution is normally available. The question whether that technology is mature enough to be easily applied to our problems. Some firms like to use state-of-the-are technology, but most firms prefer to use mature and proven technology. A mature technology has a larger customer base for obtaining advice concerning problems and improvements.[8]

Do We Currently Possess the Necessary Technology?

Assuming the solution's required technology is practical, we must next ask ourselves, is the technology available in our information systems shop? If the technology is available, we must ask if we have the capacity. For instance, will our current printer be able to handle the new reports and forms required of a new system?

If the answer to any of these questions is no, then we must ask ourselves, Can we get this technology? The technology may be practical and available, and, yes, we need it. But we simply may not be able to afford it at this time. Although this argument borders on economic feasibility, it is truly technical feasibility. If we can't afford the technology, then the alternative that requires the technology is not practical and is technically infeasible!

Strength and Weaknesses of SDLC	
Strengths	**Weaknesses**
Control.	Increased development time.
Monitor Large projects.	Increased development cost.
Detailed steps.	Systems must be defined up front.
Evaluate costs and completion targets.	Rigidity.
Documentation.	Hard to estimate costs, project overruns.
Well defined user input.	User input is sometimes limited.
Ease of maintenance.	
Development and design standards.	
Tolerates changes in MIS staffing.	

Table 3.1 Strengths and Weaknesses of SDLC

3.4 IMPORTANCE OF ONLINE ONTOLOGY BASED NEWS RECOMMENDATION SYSTEM

There are several attributes in which the computer based information works. Broadly the working of computer system is divided into two main groups:

- Transaction System
- Decision Support System

Transaction System:

A transaction is a record of some well-defined single and usually small occurrence in a system. Transactions are input into the computer to update the database files. It checks the entering data for its accuracy. This means that numeric data appears in numeric field and character data in character field. Once all the checks are made, transaction is used to update the database. Transaction can be inputted in on-line mode or batch mode. In on-line mode, transactions are entered and updated into the database almost instantaneously. In batch mode, transactions are collected into batches, which may be held for a while and inputted later.

Decision Support System:

It assists the user to make analytical decision. It shows the various data in organized way called analysis. This analysis can be made to syrdy preferences and help in making decisions.

Computer system works out best with record maintenance. It will tell you which customer would get how much pending/reports statements. It will also help to search the information about a particular person by simply entering his telephone number. User can store information as per requirement, which can be used for comparison with other reports.

3.5 PRINCIPLES OF SYSTEM ANALYSIS

Principles:

1. Understand the problem before you begin to create the analysis model.
2. Develop prototypes that enable a user to understand how human machine interaction will occur.
3. Record the origin of and the reason for every requirement.
4. Use multiple views of requirements like building data, function and behavioral models.
5. Work to eliminate ambiguity.

A Complete Structure:

The limited time and resources have restricted us to incorporate, in this project, only the main activities that are performed in news sites, but utmost care has been taken to make the system efficient and user friendly.

For the optimum use of practical time it is necessary that every session is planned. Planning of this project will include the following things:

- Topic Understanding.
- Modular Break – Up of the System.
- Processor Logic for Each Module.
- Database Requirements.

Topic Understanding:

It is vital that the field of application as introduced in the project may be totally a new field. So as soon as the project was allocated to me, I carefully went through the project to identify the requirements of the project.

Modular Break –Up of the System:

- Identify The Various Modules In The System.
- List Them In The Right Hierarchy.
- Identify Their Priority Of Development
- Description Of The Modules:

3.6 SYSTEM DESIGN

The design document that we will develop during this phase is the blueprint of the software. It describes how the solution to the customer problem is to be built. Since solution to complex problems isn't usually found in the first try, iterations are most likely required. This is true for software design as well. For this reason, any design strategy, design method, or design language must be flexible and must easily accommodate changes due to iterations in the design. Any technique or design needs to support and guide the partitioning process in such a way that the resulting sub problems are as independent as possible from each other and can be combined easily for the solution to the overall problem. Sub problem independence and easy combination of their solutions reduces the complexity of the problem. This is the objective of the partitioning process. Partitioning or decomposition during design involves three types of decisions:

- Define the boundaries along which to break;
- Determine into how money pieces to break; and

Identify the proper level of detail when design should stop and implementation should start. Basic design principles that enable the software engineer to navigate the design process suggest a set of principles for software design, which have been adapted and extended in the following list:

Free from the suffer from "tunnel vision." A good designer should consider alternative approaches, judging each based on the requirements of the problem, the resources available to do the job.

The design should be traceable to the analysis model. Because a single element of the design model often traces to multiple requirements, it is necessary to have a means for tracking how requirements have been satisfied by the design model.

The design should not repeat the same thing. Systems are constructed using a set of design patterns, many of which have likely been encountered before. These patterns should always be chosen as an alternative to reinvention. Time is short and resources are limited! Design time should be invested in representing truly new ideas and integrating those patterns that already exist. The design should "minimize the intellectual distance" between the software and the problem as it exists in the real world. That is, the structure of the software design should (whenever possible) mimic the structure of the problem domain.[10]

The design should exhibit uniformity and integration. A design is uniform if it appears that one person developed the entire thing. Rules of style and format should be defined for a design team before design work begins. A design is integrated if care is taken in defining interfaces between design components.

The design activity begins when the requirements document for the software to be developed is available. This may be the SRS for the complete system, as is the case if the waterfall model is being followed or the requirements for the next "iteration" if the iterative enhancement is being followed or the requirements for the prototype if the prototyping is being followed. While the requirements specification activity is entirely in the problem domain, design is the first step in moving from the problem domain toward the solution domain. Design is essentially the bridge between requirements specification and the final solution for satisfying the requirements.

The design of a system is essentially a blueprint or a plan for a solution for the system. We consider a system to be a set of components with clearly defined behavior that interacts with each other in a fixed defined manner to produce some behavior or services for its environment. A component of a system can be considered a system, with its own components. In a software system, a component is a software module.

The design process for software systems, often, has two levels. At the first level, the focus is on deciding which modules are needed for the system, the specifications of these modules, and how the modules should be interconnected. This is what is called the system design or top level design. In the second level, the internal design of the modules, or how the specifications of the module can be satisfied, is decided. This design level is often called detailed design or logic design. Detailed design essentially expands the system design to contain a more detailed description of the processing logic and data structures so that the design is sufficiently complete for coding.

Because the detailed design is an extension of system design, the system design controls the major structural characteristics of the system. The system design has a major impact on the testability and modifiability of a system, and it impacts its efficiency. Much of the design effort for designing software is spent creating the system design.

The input to the design phase is the specifications for the system to be designed. Hence, reasonable entry criteria can be that the specifications are stable and have been approved, hoping that the approval mechanism will ensure that the specifications are complete, consistent, unambiguous, etc. The output of the top level design phase is the architectural design or the system design for the software system to be built. This can be produced with or without using a design methodology. Reasonable exit criteria for the phase could be that the design has been verified against the input specifications and has been evaluated and approved for quality.

A design can be object oriented or function oriented. In function oriented design, the design consists of module definitions, with each module supporting a functional abstraction. In object oriented design, the modules in the design represent data abstraction (these abstractions are discussed in more detail later). In the function oriented methods for design and describe one particular methodology the structured design methodology in some detail. In a function oriented design approach, a system is viewed as a transformation function, transforming the inputs to the desired outputs. The purpose of the design phase is to specify the components for this transformation function, so that each component is also a transformation function. Hence, the basic output of the system design phase, when a function oriented design approach is being followed, is the definition of all the major data structures in the system, all the major modules of the system, and how the modules interact with each other.

Once the designer is satisfied with the design he has produced, the design is to be precisely specified in the form of a document. To specify the design, specification languages are used. Producing the design specification is the ultimate objective of the design phase. The purpose of this design document is quite different from that of the design notation. Whereas a design represented using the design notation is largely to be used by the designer, a design specification has to be so precise and complete that it can be used as a basis of further development by other programmers. Generally, design specification uses textual structures, with design notation helping in understanding

3.8 ER - DIAGRAM

3.9 ACTIVITY DIAGRAMS

How activity diagrams are drawn?

Activity diagrams are mainly used as a flow chart consists of activities performed by the system. But activity diagram are not exactly a flow chart as they have some additional capabilities. These additional capabilities include branching, parallel flow, swimlane etc.

Before drawing an activity diagram we must have a clear understanding about the elements used in activity diagram. The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities we need to understand how they are associated with constraints and conditions.

Once the above mentioned parameters are identified we need to make a mental layout of the entire flow. This mental layout is then transformed into an activity diagram.

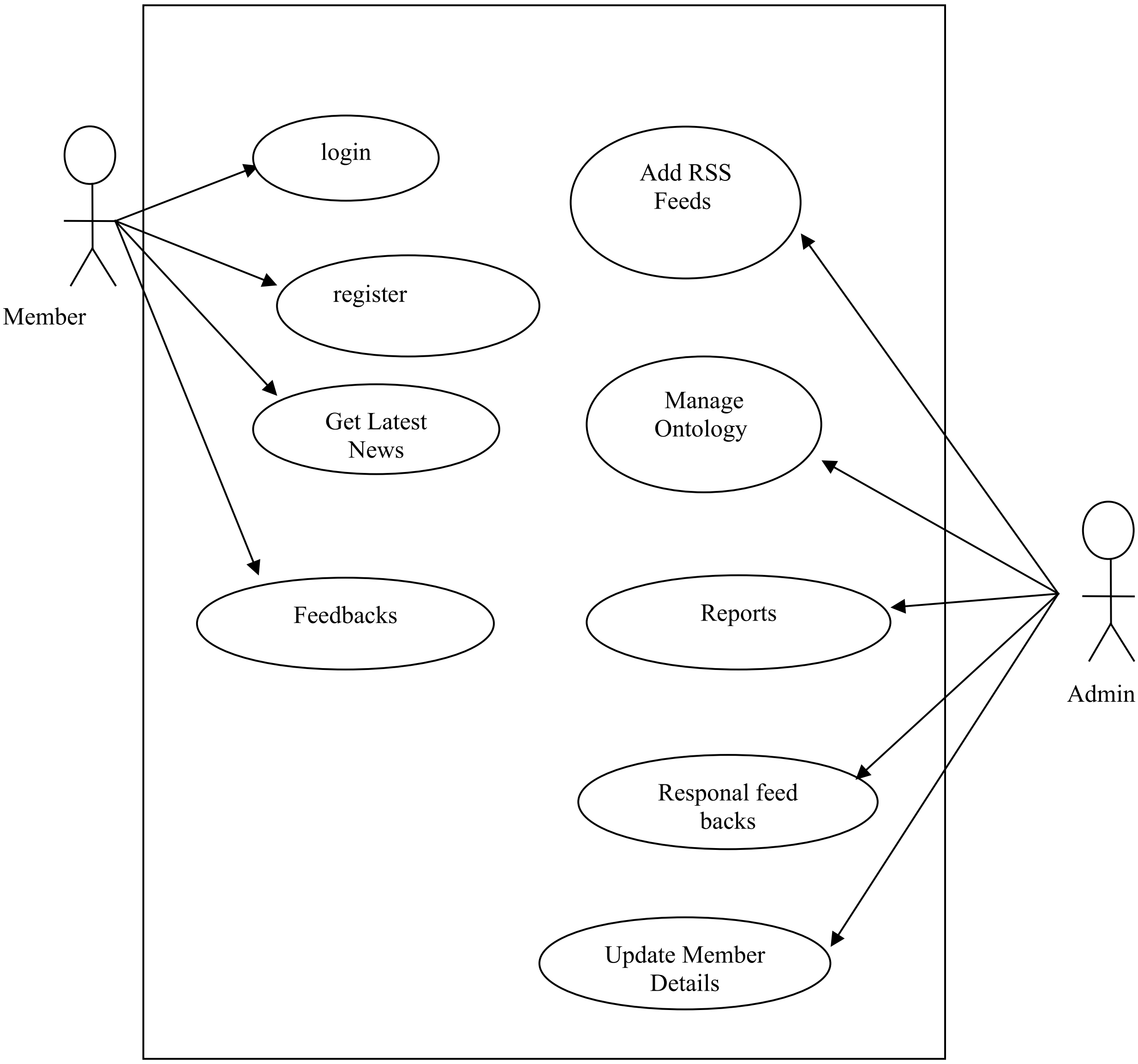

Member
login
register
Get Latest News
Feedbacks
Add RSS Feeds
Manage Ontology
Reports
Responal feed backs
Update Member Details
Admin

Fig. 3.12

3.11 DATA MODELING

Introduction to data dictionary:

Data dictionaries are an integral component of structured analysis, since data flow diagrams by themselves do not fully describe the subject of the investigation . The data flow diagrams provide the additional details about the project/system.

Data Dictionary (Definition):

A data dictionary is a catalog- a repository- of the elements in a system. These elements center on the data and the way they are structured to meet user requirements and organization needs. A data dictionary consists of a list of all the elements composing the data flowing through a system. The major elements are data flows , data stores , and processes. The data dictionary stores details and descriptions of these elements.

3.12 PROCESS LOGIC FOR EACH MODULE

Process Logic of Module is a tool that may be useful in planning and evaluating projects. Our logic model contains four components with Inputs-Outputs-Outcomes being central to the built in response to the model:

- **Situation:** The context and need that gives rise to a project or initiative; logic modules are built in response to an existing situation.

- **Inputs:** The resources, contributions, and investments that are made in response to the situation. Inputs lead to Outputs.

- **Outputs:** The activities, products, methods, and services that reach people and users. Outputs lead to outcomes.

- **Outcomes:** The results and benefits for individuals, groups, agencies, communities and/or systems.

3.13 SCHEDULING

Scheduling of a software project does not differ greatly from scheduling of any multi- task engineering effort. Therefore, generalized project scheduling tools and techniques can be applied with little modification to software projects.

Program evaluation and review technique (PERT) and critical path method (CPM) are two project scheduling methods that can be applied to software development. Both techniques are driven by information already developed in earlier project planning activities.

Estimates of Effort

- A decomposition of the product function.
- The selection of the appropriate process model and task set.
- Decomposition of tasks.

Interdependencies among tasks may be defined using a task network. Tasks, sometimes called the project Work Breakdown Structure (WBS) are defined for the product as a whole or for individual functions.

Both PERT and CPM provide quantitative tools that allow the software planner to (1) determine the critical path-the chain of tasks that determines the duration of the project; (2) establish "most likely" time estimates for individual tasks by applying statistical models; and (3) calculate "boundary times" that define a time window" for a particular task.

Boundary time calculations can be very useful in software project scheduling. Slippage in the design of one function, for example, can retard further development of other functions. It describes important boundary times that may be discerned from a PERT or CPM network: (I) the earliest time that a task can begin when preceding tasks are completed in the shortest possible time, (2) the latest time for task initiation before the minimum project completion time is delayed, (3) the earliest finish-the sum of the earliest start and the task duration, (4) the latest finish- the latest start time added to task duration, and (5) the total float-the amount of surplus time or leeway allowed in scheduling tasks so that the network critical path maintained on schedule. Boundary time calculations lead to a determination of critical path and provide the manager with a quantitative method for evaluating progress as tasks are completed.

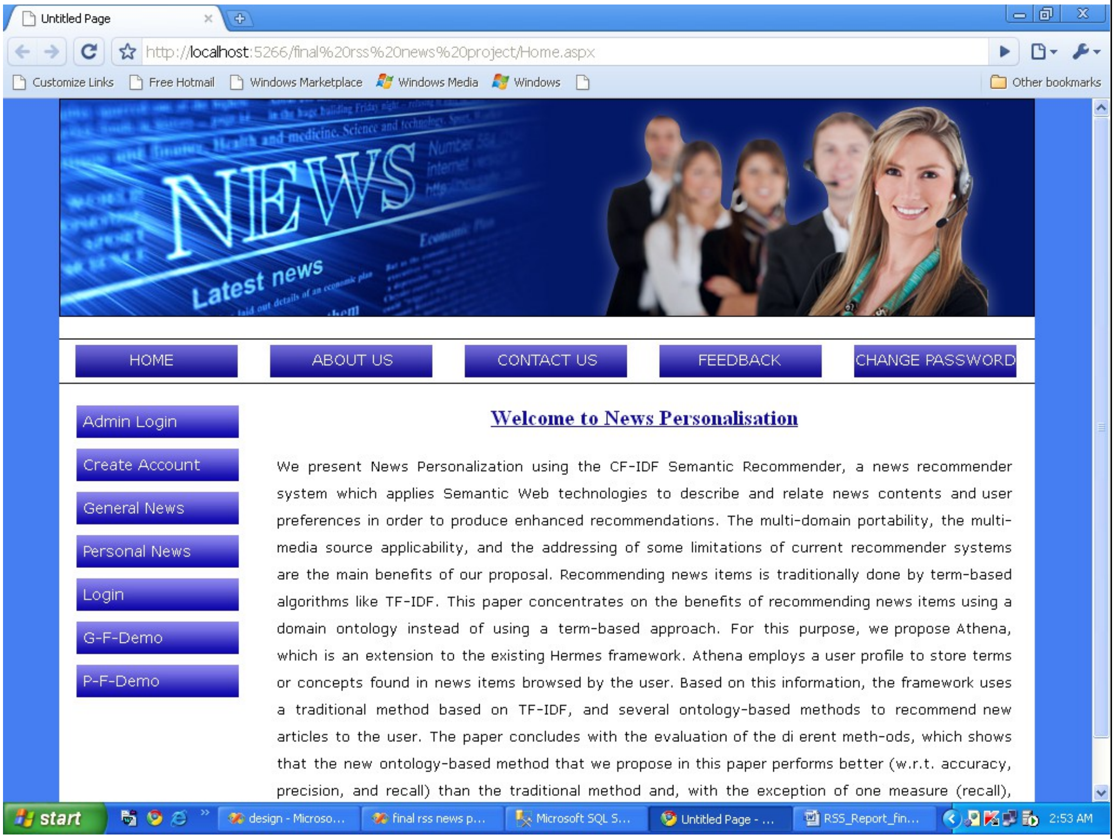
Both PERT and CPM have been implemented in a wide variety of automated tools that are available for the personal computer. Such tools are easy to use and take the scheduling methods described previously available to every software project manager.

CHAPTER-4

SYSTEM DEVELOPMENT

4.1 SOURCE CODE

Home.aspx



Welcome to News Personalisation

We present News Personalization using the CF-IDF Semantic Recommender, a news recommender system which applies Semantic Web technologies to describe and relate news contents and user preferences in order to produce enhanced recommendations. The multi-domain portability, the multi-media source applicability, and the addressing of some limitations of current recommender systems are the main benefits of our proposal. Recommending news items is traditionally done by term-based algorithms like TF-IDF. This paper concentrates on the benefits of recommending news items using a domain ontology instead of using a term-based approach. For this purpose, we propose Athena, which is an extension to the existing Hermes framework. Athena employs a user profile to store terms or concepts found in news items browsed by the user. Based on this information, the framework uses a traditional method based on TF-IDF, and several ontology-based methods to recommend new articles to the user. The paper concludes with the evaluation of the different meth-ods, which shows that the new ontology-based method that we propose in this paper performs better (w.r.t. accuracy, precision, and recall) than the traditional method and, with the exception of one measure (recall),

4.2 CODE EFFICIENCY

Reviewing of Code efficiency for a module is carried out after the module is successfully compiled and all the syntax errors eliminated. Code efficiency review is extremely cost-effective strategies for reduction in coding errors in order to produce high quality code. Normally, two types of efficiency are carried out on the code of a module - code optimization and code inspection. The procedure and final objective of these two efficiency techniques are very different as discussed below.

4.3 OPTIMIZATION OF CODE

Code optimization is an informal code analysis technique. In this technique, after a module has been coded, it is successfully compiled and all syntax errors are eliminated.

Some members of the development team are given the code a few days before the optimization meeting to read and understand the code. Each member selects some test cases and simulates execution of the code by hand (i.e. trace execution through each statement and function execution). The main objectives of the optimization are to discover the algorithmic and logical errors in the code. The members note down their findings to discuss these in a optimization meeting where the coder of the module is also present.

Even though a code optimization is an informal analysis technique, several guidelines have evolved over the years for making this naïve technique more effective and useful. Of course, these guidelines are based on personal experience, common sense, and several subjective factors. Therefore are based on personal experience, common sense, and several subjective factors. Therefore, guidelines should be considered as examples rather than as rules to be applied dogmatically. Some of these guidelines are the following:

The team performing the code optimization should not be either too big or too small. Ideally, it should consist of three to seven members.

TESTING

CHAPTER- 5

TESTING

5.1 TESTING PHASE

One of the purposes of the testing is to validate and verify the system. Verification means checking the system to ensure that it is doing what the function is supposed to do and Validation means checking to ensure that system is doing what the user wants it to do.

No program or system design is perfect; communication between the user and the designer is not always complete or clear, and time is usually short. The result is errors and more errors. Theoretically, a newly designed system should have all the pieces in working order, but in reality, each piece works independently. Now is the time to put all the pieces into one system and test it to determine whether it meets the user's requirements. This is the best chance to detect and correct errors before the system is implemented. The purpose of system testing is to consider all the likely variations to which it will be subjected and then push the system to its limits. If we implement the system without proper testing then it might cause the problems.

1. Communication between the user and the designer.
2. The programmer's ability to generate a code that reflects

exactly the system specification.

3. The time frame for the design.

Theoretically, a new designed system should have all the pieces in working order, but in reality, each piece works independently. Now is the time to put all the pieces into one system and test it to determine whether it meets the requirements of the user. The

CHAPTER - 6

SYSTEM IMPLEMENTATION

6.1 SYSTEM IMPLEMENTATION MAINTENANCE AND REVIEW

As we know, creating software is one thing and the implementation of the created software is another. The process of implementing software is much difficult as compared to the task of creating the project. First we have to implement the software on a small scale for removing the bugs and other errors in the project and after removing them we can implement the software on a large scale. Before we think in terms of implementing the Software on a large basis, we must consider the Hardware requirements.

Whenever we develop software or project a certain hardware and software is being used by the programmer for developing the project. The hardware and software to be used by the programmer for developing the project should be such that it would result in the development of a project, which would satisfy all the basic needs for which the project has been created by the programmer. The Hardware should be such that cost constraints of the Client should also be taken into account without affecting the performance.

6.2 HARDWARE EVALUATION FACTORS

When we evaluate computer hardware, we should first investigate specific physical and performance characteristics for each hardware component to be acquired. These specific questions must be answered concerning many important factors. These hardware evaluation factors questions are summarized in the below figure.

Notice that there is much more to evaluating hardware than determining the fastest and cheapest computing device. For e.g. the question of possible obsolescence must be addressed by making a technology evaluation. The factor of ergonomics is also very important. Ergonomics is the science and technology that tries to ensure that computers and other technologies are "user-friendly", that is safe, comfortable and easy to use. Connectivity is another important evaluation factor, since so many computer systems are now interconnected within wide area or local area telecommunications networks.

<u>Hardware Evaluation Factors:-</u>

1) Performance
2) Cost
3) Reliability
4) Availability
5) Compatibility
6) Modularity
7) Technology
8) Ergonomics
9) Connectivity
10) Environmental requirements
11) Software
12) Support

6.3 SOFTWARE EVALUATION FACTORS

Software can be evaluated according to many factors similar to the hardware evaluation. Thus the factors of performance, cost, reliability, compatibility, modularity, technology, ergonomics, and support should be used to evaluate proposed software acquisitions. In addition, however, the software evaluation factors are summarized in below figure. For e.g. some software packages require too much memory capacity and are notoriously slow, hard to use, or poorly documented. They are not a good selection for most end users, even if offered at attractive prices.

Software evaluation factors:

1. **EFFICIENCY:** is the software a well-written system of computer instructions that does not use much memory capacity or CPU time?
2. **FLEXIBILITY:** can it handle its processing assignments easily without major modifications?
3. **SECURITY:** does it provide control procedures for errors, malfunctions and improper use?
4. **LANGUAGE:** do our computer programmers and users write it in a programming language that is used?
5. **DOCUMENTATION:** is the s/w well documented? Does it include helpful user instructions?
6. **HARDWARE:** does existing hardware have the features required to best use this software?

7. Other characteristics of hardware such as its performance, what about the cost, how much is reliable and etc.

6.4 CONVERSION AND TRAINING

An important aspect of is to make sure that the new design is implemented to establish standards. The term implementation has different meanings, ranging form the conversion of a basic application to a complete replacement of a computer system. Implementation is used here to mean the process of converting a new or revise system into an operational one. Conversion is one aspect of implementation. Conversion means changing form one system to another. The objective is to put the tested system into operation while holding costs, risks, and personnel irritation to a minimum. It involves creating computer-compatible files, training the operation staff, and installing terminal and hardware. A critical aspect of conversion is not disrupting the functioning of the organization.

When a new system is used over and old, existing and running one, there are always compatibility errors. These errors are caused because of the lack of equipment or personnel to work the new system. Running any specified system at an organization does require some or other hardware or, in this case, software requirement as well.

Conversion is one aspect of implementation review & software maintenance.

There are three types of implementation:

1. Implementation of a computer system to replace a manual system. The problems encountered are converting files, training users, creating accurate files and verifying printouts for integrity.
2. Implementation of a new computer system to replace an existing one. This is usually a difficult conversion. If not properly planned there can be many problems. Some large computer systems have taken as long as year to convert.
3. Implementation of a modified application to replace an existing one, using the same computer. This type of conversion is relatively easy to handle, provided there are no major changes in the files.

6.5 TRAINING NEEDS

Training needs refer to the gaining of knowledge required for running the system.

First of all the system is a computer based system therefore the person should have good knowledge about computer and its working. He should know how to use software's on the computer.

For a better usage and working of the software the organization should appoint a person who has good knowledge of all the required softwares. The organization gets a person trained through different institutes present in the market. The training should be as per the above requirements.

6.6 COST ESTIMATION OF THE PROJECT

Cost in a project is due to the requirements for software, hardware, and human resources. Hardware resources are computer time, terminal time and memory required for the project. Software resources include the tools and compilers needed during development. The bulk of cost of software development is due to human resources needed. Cost estimates are determined in terms of person-months (PM).

Total No. Of Persons Involved In This Project:

1. Administrator
2. Senior Programmer
3. Junior Programmers
4. On line Users.

Since this Project will complete in 4 months

COST ESTIMATE: (Salary of Project Manager + Salary of Senior Programmer + 2 * Salary of Junior Programmer) * 2

6.7 . GANTT & PERT CHART

Gantt chart

Gantt charts mainly used to allocate resources to activities. The resources allocated to activities include staff, hardware, and software. Gantt charts (named after its developer Henry Gantt) are useful for resource planning. A Gantt chart is special type of bar chart where each bar represents an activity. The bars are drawn along a timeline. The length of each bar is proportional to the duration of the time planned for the corresponding activity. A Gantt chart is a type of bar chart that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project.

Gantt chart is a project scheduling technique. Progress can be represented easily in a Gantt chart, by coloring each milestone when completed. The project will start in the month of December and end after 4 months at the end of March.

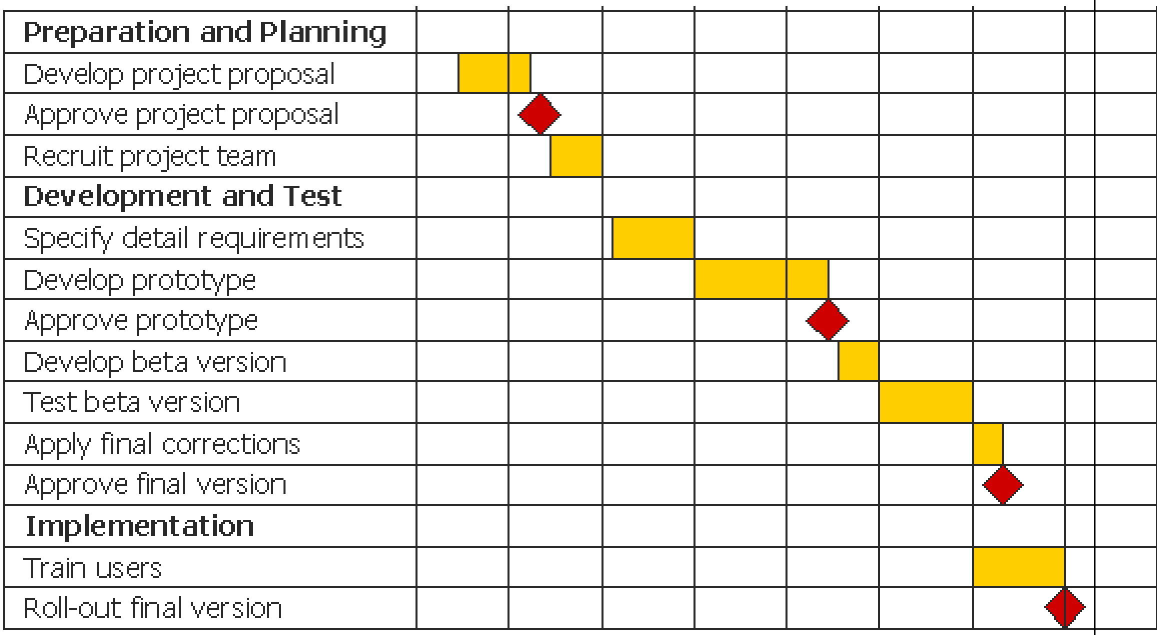

Preparation and Planning
Develop project proposal
Approve project proposal
Recruit project team
Development and Test
Specify detail requirements
Develop prototype
Approve prototype
Develop beta version
Test beta version
Apply final corrections
Approve final version
Implementation
Train users
Roll-out final version

Fig 6.1

Pert Chart

PERT (Project Evaluation and Review Technique) charts consist of a network of boxes and arrows. The boxes represent activities and the arrows represent task dependencies. A PERT chart is a project management tool used to schedule, organize, and coordinate tasks within a project. PERT stands for Program Evaluation Review Technique, a methodology developed by the U.S. Navy in the 1950s to manage the Polaris submarine missile program. A similar methodology, the Critical Path Method (CPM) was developed for project management in the private sector at about the same time.

PERT chart represents the statistical variations in the project estimates assuming a normal distribution. Thus in a PERT chart instead of making a single estimate for each task, pessimistic, likely, and optimistic estimates are also made. The boxes of PERT charts are usually annotated with the pessimistic, likely, and optimistic estimates for every task. Since all possible completion times between the minimum and maximum durations for every task have to be considered, there are many critical paths, depending on the permutations of the estimates for each task. This makes critical path analysis in PERT charts very complex. A critical path in a PERT chart is shown by using thicker arrows.

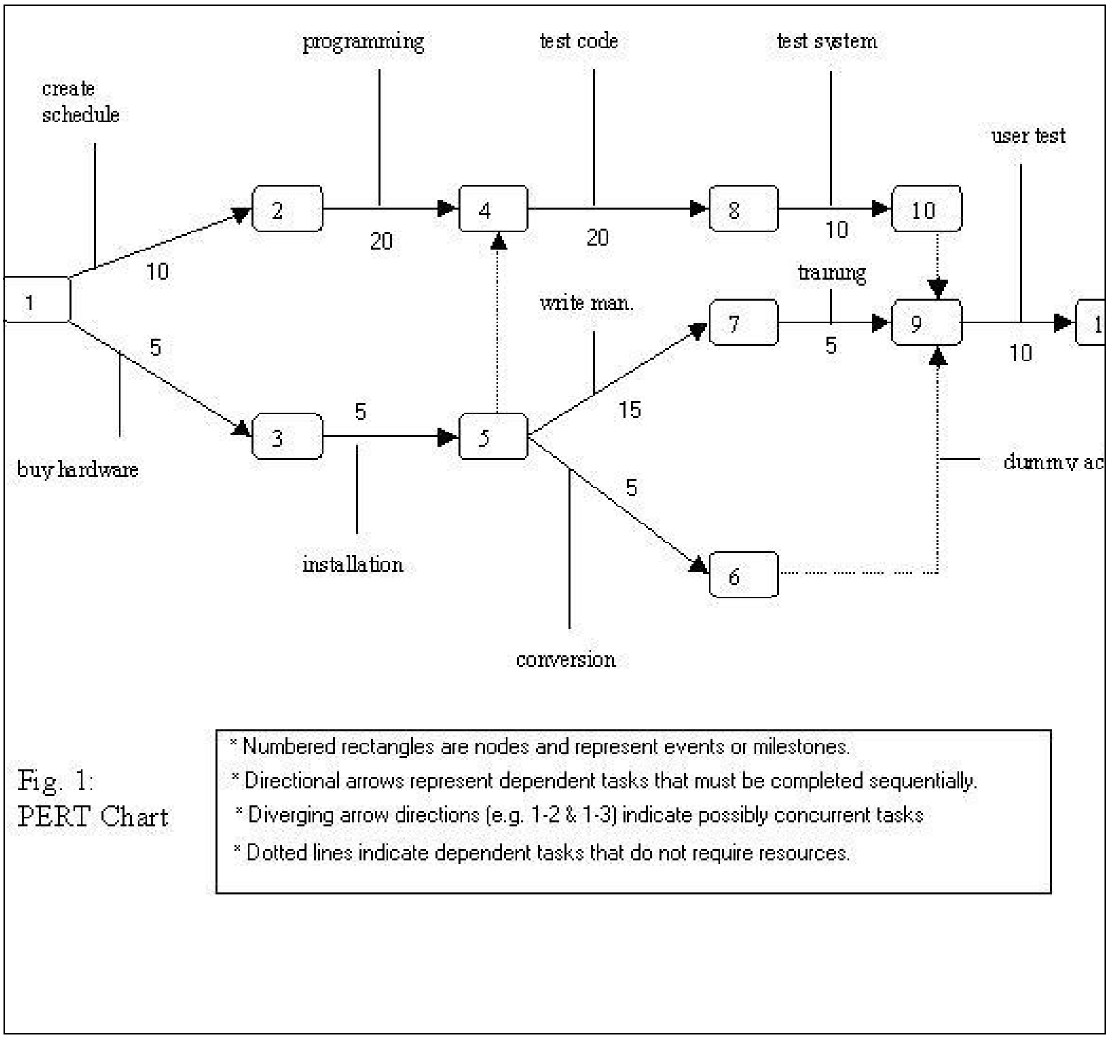


Fig. 1:
PERT Chart

* Numbered rectangles are nodes and represent events or milestones.

* Directional arrows represent dependent tasks that must be completed sequentially.

* Diverging arrow directions (e.g. 1-2 & 1-3) indicate possibly concurrent tasks

* Dotted lines indicate dependent tasks that do not require resources.

Fig 6.2

FIGURE B: PERT Chart representation of the Ontology Based News Recommendation System

PERT charts are a more sophisticated form of activity chart. In activity diagrams only the estimated task durations are represented. Since the actual durations might vary from the estimated durations, the utility of the activity diagrams is limited.

6.8. SECURITY AND VALIDATION CHECKS

In this project we have used following validation checks.

- While entering the data into the form it will check for the name of the user is properly filled & it should not be null.
- henever we enter the data for the new user will automatically check the details from the database tables and also generate the connection number automatically.
- Entered text / number should not exceed the limit (width).
- lmost for all fields we have used the validation for example if name of the fields requires the text type of data then it will check for the string and if the data is numeric then it will check if the number entered is proper numeric or not.

Software's Vulnerability to Attack

Software development is not yet a science or a rigorous discipline, and the development process by and large is not controlled to minimize the vulnerabilities that attackers exploit.

The security of software is threatened at various points throughout its life cycle, both by inadvertent and intentional choices and actions taken by "insiders"—individuals closely affiliated with the organization that is producing, deploying, operating, or maintaining the software, and thus trusted by that organization—and by “outsiders” who have no affiliation with the organization. The software’s security can be threatened

- **During its development:** A developer may corrupt the software—intentionally or unintentionally—in ways that will compromise the software’s dependability and trustworthiness when it is operational.
- **During its deployment (distribution and installation):** If those responsible for distributing the software fail to tamperproof the software before shipping or uploading, or transmit it over easily intercepted communications channels, they leave the software vulnerable to intentional or unintentional corruption. Similarly, if the software’s installer fails to “lock down” the host platform, or configures the software insecurely, the software is left vulnerable to access by attackers.
- **During its operation:** Once COTS and open source software has gone operational, vulnerabilities may be discovered and publicized; unless security patches and updates are applied and newer supported versions (from which the root causes of vulnerabilities have been eliminated) are adopted, such software will become increasingly vulnerable. Non-commercial software and open source software (OSS) may also be vulnerable, especially as it may manifest untrustworthy behaviors over time due to changes in its environment that stress the software in ways that were not anticipated and simulated during its testing.

- **During its sustainment:** If those responsible for addressing discovered vulnerabilities in released software fail to issue patches or updates in a timely manner, or fail to seek out and eliminate the root causes of the vulnerabilities to prevent their perpetuation in future releases of the software, the software will become increasingly vulnerable to threats over time. Also, the software's maintainer may prove to be a malicious insider, and may embed malicious code, exploitable flaws, etc., in updated versions of the code.[14]

The Challenge of Building Secure Software

1. **Dependability:** Dependable software executes predictably and operates correctly under all conditions, including hostile conditions, including when the software comes under attack or runs on a malicious host.
2. **Trustworthiness:** Trustworthy software contains few if any vulnerabilities or weaknesses that can be intentionally exploited to subvert or sabotage the software's dependability. In addition, to be considered trustworthy, the software must contain no malicious logic that causes it to behave in a malicious manner.

Survivability (also referred to as "Resilience"): Survivable—or resilient—software is software that is resilient enough to (1) either resist (i.e., protect itself against) or tolerate (i.e., continue operating dependably in spite of) most known attacks plus as many novel attacks as possible, and (2) recover as quickly as possible, and with as little damage as possible, from those attacks that it can neither resist nor tolerate

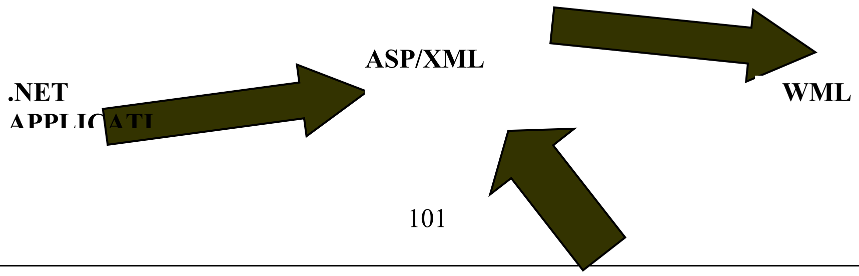
6.9 SCOPE OF FUTURE APPLICATION

It is directly dependent on the lay stone of the project that is we will have to design a system which when the time passes having a better system initially should not become a joke later.

This allows the development team to freeze the scope for one increment so that an operational web application release can be created. The next increment may scope changes suggested by a review of the preceding increment, but once the second increment commences, scope is again frozen temporarily. This approach enable the web application team to work without having to accommodate a continual stream of change but still recognizes the continuous evolution characteristics of most web application.

Besides that, the following basic quality in the software always safeguards the future scope of the software.

- Currently it is running for ASP.Net, I am updating it so that it can have **more subjects** so that everybody having different knowledge can test his/her capabilities.
- The project can also be upgraded to run on **networking protocols.**
- Compatible with **.Net plate-form** to easily handle with other **MS Programming Language**.



SQL-SERVER
(Database Management System)

Fig 6.3

6.10. CONCLUSION

This project is designed to meet the shown that the ranked recom-mender, our ontology-based recommender, performs better than a traditional recommender system based on TF-IDF for accuracy, precision, and recall, and equally good for specificity. It also performs better, or equally good, with respect to accuracy, precision, and specificity than the other considered ontology-based recommenders. Nevertheless, the recall is lower than some of the implemented ontology-based recommenders.

For designing the system we have used simple data flow diagrams.

Overall the project teaches us the essential skills like:

- ❑ Using system analysis and design techniques like data flow diagram in designing the system.
- ❑ Understanding the database handling and query processing using SQL Server

REFERENCES

1. Ahn, J., Brusilovsky, P., Grady, J., He, D., Syn, S. Y. (2007). *Open User Profiles for Adaptive News Systems: Help or Harm?* Proceedings of the 16th International Conference on World Wide Web (WWW 2007), pp. 11-20. Banff, Alberta, Canada.

2. Billsus, D. (1999). *A Personal News Agent that Talks, Learns and Explains*. Proc. of the 3rd International Conference on Autonomous Agents (Agents 1999), 268-275. Seattle, WA, USA.
3. Billsus, D., Pazzani, M. J. (2000). *User Modeling for Adaptive News Access*. User Modeling and User-Adapted Interaction, 10, pp. 147-180.
4. Cantador, I., Bellogín, A., Castells, P. (2007). *Modelling Ontology-based Multilayered Communities of Interest for Hybrid Recommendations*. Proc. of the 1st Int. Workshop on Adaptation and Personalisation in Social Systems: Groups, Teams, Communities (SociUM 2007). Corfu, Greece.
5. Cantador, I., Castells, P. (2006). *Multilayered Semantic Social Networks Modelling by Ontology-based User Profiles Clustering: Application to Collaborative Filtering*. Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2006), pp. 334-349. Podebrady, Czech Republic.
6. Cantador, I., Castells, P., Vallet, D. (2006). *Enriching Group Profiles with Ontologies for Knowledge-Driven Collaborative Content Retrieval*. Proc. of the 1st Int. Workshop on Semantic Technologies in Collaborative Applications (STICA 2006), pp. 358-363. Manchester,

UK.

7. Castells, P., Fernández, M., Vallet, D. (2007). *An Adaptation of the Vector-Space Model for Ontology-based Information Retrieval*. IEEE Transactions on Knowledge and Data Engineering, 19 (2), pp. 261-272.

8. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M. (1999). *Combining Content-Based and Collaborative Filters in an Online Newspaper*. Proceedings of the ACM SIGIR'99 Workshop on Recommender Systems. Berkeley, CA, USA.

9. Das, A., Datar, M., Garg, A., Rajaram, S. (2007). *Google News Personalisation: Scalable Online Collaborative Filtering*. Proceedings of the 16th International Conference on World Wide Web (WWW 2007), pp. 11-20. Banff, AB, Canada.

10. Jones, G. J. F., Quested, D. J., Thomson, K. E. (2000). *Personalised Delivery of News Articles from Multiple Sources*. In J. Borbinha, T. Baker (Eds.), ECDL 2000, pp.340-343. Springer-Verlag.

11. Lang, K. (1995). *NewsWeeder: Learning to filter Netnews*. Proceedings of the 12th International Conference on Machine Learning, pp: 331-339.

12. Maes, P. (1994). *Agents that Reduce Work and Information Overload*. Communications of the ACM, 37(7), pp. 31-40.

13. Nadjarbashi-Noghani, M., Zhang, J., Sadat, H., Ghorbani, A. A. (2005). *PENS: A Personalised Electronic News System*. Proceedings of the 3rd Annual Communication Networks and Services Research Conference (CNSR'05), pp. 31-38.

14. Vallet, D., Castells, P., Fernández, M., Mylonas, P., Avrithis, Y. (2007). Personalised Content Retrieval in

Context Using Ontological Knowledge. IEEE Transactions on Circuits and Systems for Video Technology, 17 (3), pp. 336-346.

15. Bra, P.D., Aerts, A.T.M., Houben, G.J., Wu, H.: Making General-Purpose Adaptive Hypermedia Work. In: WebNet 2000 Conference, AACE (2000) 117–12

16. Guzman-Lara, S.: KStem Java Implementation. University of Massachusetts Amherst (2007) from: http://ciir.cs.umass.edu/cgi-bin/downloads/downloads.cgi.

17. Cunningham, H.: GATE, General Architecture for Text Engineering. Computers and the Humanities 36 (2002) 223–254

18. Salton, G., Buckley, C.: Term Weighting Approaches in Automatic Text Retrieval. Information Processing and Management 24(5) (1988) 513–523

19. Woerndl, W.; Schueller, C. y Wojtech, R. (2007). "A Hybrid Recommender System for Context-aware Recommendations of Mobile Applications." Proc. of the IEEE 3rd International Workshop on Web Personalisation, Recommender Systems and Intelligent User Interfaces (WPRSIUI'07).

20. Xin, Z. et al. (2005). "Information Push-Delivery for User-Centered and Personalized Service." En: "Anonymous Fuzzy Systems and Knowledge Discovery." Berlin, Heidelberg: Springer, pp. 594-602.

21. Yu, Z. et al. (2007). "Ontology-Based Semantic Recommendation for Context-Aware E-Learning." Proc. of the 4th Conference on Ubiquitous Intelligence and Computing, v.4611, Berlin, Heidelberg: Springer, pp. 898-907.

22. Ziegler, N. (2004). "Semantic Web Recommender Systems." Proc. of the joint ICDE/EDBT Workshop.

23. Ziegler, N.; Lausen, G. y Schmidt-Thieme, L.

(2004). "Taxonomy driven computation of product recommendations." Proc. of the 30th ACM Conference on Information and Knowledge Management, pp. 406-415.