Knowledge Management
for
Engineering Support

A Thesis submitted in partial fulfillment of the
requirements for the award of the degree of

Master of Technology

in

Artificial Intelligence

By

V. Sandeep

Department of Computer and Information Sciences
University of Hyderabad
Hyderabad, India.

June - 2009

CERTIFICATE

This is to certify that the project report entitled "**Knowledge management for Engineering support**", submitted for partial fulfillment of the requirements for the award of Degree of Master of Technology in Artificial Intelligence (M.Tech (AI)) to the University of Hyderabad is a record of bonafide project work carried out by **Mr. V. Sandeep** (07MCMI08) at India Software Labs, IBM India Pvt. Ltd., Hyderabad, for a period of one year during June 2008 to June 2009 under the guidance of Mr. Vijai Bhasker Terala, Senior Manager, Connectivity team, IBM (IBM, India Software Labs, Hyderabad).

Dr. Vineet P Nair.,
Project Supervisor,
DCIS
University of Hyderabad.

Mr. Vijai Bhasker Terala,
Senior Manager, Connectivity
IBM India Software Labs,
Hyderabad.

Prof. Arun Agarwal
Head of the Department, DCIS,
University of Hyderabad,

Prof. T. Amaranath,
Dean (School of MCIS),
University of Hyderabad,

ACKOWLEDMENTS

I sincerely thank **Prof. Arun Agarwal**, Head, DCIS, University of Hyderabad for giving me an opportunity to work with IBM.

I thank **Dr. Rajeev Wankar**, Reader, DCIS, University of Hyderabad, who could make my internship programme at IBM possible.

Dr. Vineet P Nair, Lecturer, DCIS, my research adviser, translated my wish to write a thesis into a series of concrete, well-planned sub-goals. I thank him for his support, guidance and timely advice throughout.

I thank all the faculty members of DCIS, UOH for their support. I wish to thank them for being very patient, understanding and helpful.

I would like to express my deepest gratitude to **Dr. Kavi Narayana Murty**, Professor, DCIS, UOH for introducing me to the wonderful subject of Text Engineering.

I am very thankful to IBM India Pvt, Ltd., for offering me an Internship at India Software Labs, Hyderabad.

Mr. Vijai Bhasker Terala, my manager during my internship tenure, taught me a lot about professional attitude towards work in addition to guiding me through the nitty-gritty's of project.

I express my sincere thanks to **Mr. Shaikh Quader** and **Mr. Yi Fan** without whom it would have been impossible to complete my work in a meaningful manner. I thank them for their encouragement and continuous guidance.

I would like to thank **Dr. Sree Rama Murty**, **Dr. Krishna Kummamuru and Dr. Prasad Deshpandey** for reviewing my design proposals and providing me with valuable inputs that helped to shape my project better.

My present status in life, this Masters' included, is a result of the extraordinary will, efforts and sacrifices of my family.

V. Sandeep

Table of Contents

Abstract

"Power comes from transmitting information to make it productive, not by hiding it"
- Drucker

Knowledge management is a term commonly used with the context of an organizational structure. Prior studies in knowledge management reveal that organizations powered by knowledge management systems or schemes compete better, increase responsiveness and innovativeness. The context of knowledge management in this project is specific to industries that roll out products and provides solutions to customers.

The process of resolving customer issues is a cyclic process (a definite chain of events) involving many role players. This mechanism is known as "Engineering support".

There exist many task specific applications in the Industry that attempt to resolve this issue however they face the problems of compatibility and robustness. Managing information present in these information sources so that the right information can be given to the right person at the right time is the goal of this project.

This project strives to achieve the following objectives:

- Present a detailed study of Existing Knowledge management scenario in product development organizations.
- Propose an ideal model for knowledge management for industries based on state of the art knowledge management techniques.
- Present a gap analysis study between the existing system of knowledge management and the proposed ideal system for knowledge management.
- Propose designs for applications (called bridge applications) that address each of these gaps mentioned in the previous stage. (Applications that integrate with the existing systems and facilitate its transformation into an ideal model without disturbing the harmony of existing systems)

- Identify off the shelf components that can be integrated in order to implement the proposed models.
- Implementing a concept in the existing system to make the process run better. (Implementing the concept of similar pages in the already existing Enterprise search application)

Chapter 1

Introduction to thesis

This chapter introduces the basic terminology and concepts to be used in the thesis. The chapter begins with a detailed description of reasons for taking up this work, the motivational factors. Later in the chapter, objectives / milestones of the work are discussed in full detail, followed by organization of thesis.

1.1 Motivation

Data growth rates over the past decade have been tremendous. This growth in data is often known as a data overflow or data flood. Data is generated from various sources like banks, telecommunications, business transactions, scientific data related to astronomy, web, text, e-commerce etc. [2] [3]

The graph below shows that the size of the largest database got tripled over the last two years. [5]

This trend in organizational data enforces a necessity to manage the content available and elicit knowledge from it. More specifically, the organizational content has to be properly

managed in order to improve the efficiency, productivity and competitiveness of the organizations [2] [3].

This process of streamlining the otherwise haphazard organizational content is known as Knowledge management [1] [4]. Knowledge management is defined as identifying Information sources in the organization and leveraging collective knowledge of the organization.

Knowledge management is an industrial buzz word and multiple organizations have come up with knowledge management frameworks. These platforms give services which can be configured (customized) to suit specific needs of an organization.

In spite of the existence of significant knowledge management frameworks, organizations still complain of unable to achieve the desired results. The primary aim of a knowledge management system has to be delivering the right information to the right person at the right time.

A survey at Cranfield university [5] revealed that the employees across multiple organizations feel that the data that they require is very well present in the organization however, to identify the source to retrieve data at that time is not possible.

In addition to these factors, employee attrition (Most of the industrial processes are people dependent) and product specificity (Knowledge management products developed are too specific and incompatible to integrate) makes the matters worse for knowledge management in Industries.

Previous studies of knowledge management have all been specific to a particular industrial process however no studies did a completely analysis of the entire end to end process. (Considering the industrial process as a sequence of events)

The work proposed here makes an attempt to address the issue mentioned above. This work deals with providing an effective knowledge management framework for a multi national organization. More specifically, the work mentioned here deals with providing a knowledge management solution for End to End customer support process in this organization.

Similar works for the applicability of knowledge management techniques for entire support process have been rare world wide. Hence a detailed comparative study of existing systems for knowledge management of customer support with ideal systems is taken as the foundation to kick start this work. A detailed study of each of these aspects is provided in the chapters to follow. The next section would explain the main objectives or mile stones that are expected to be achieved in order to complete this work.

1.2 Description of objectives

This section discusses the objectives of this work. This work deals with studying the customer support cycle of the given organization. A detailed survey of the tools/applications being used for knowledge management in this support process has to be presented enumerating the reasons for their inefficiency. Usually works of this nature are done based on a prior study or results already published. However, the approach followed here is slightly different. As the entire support process is viewed instead of a particular process in the customer support cycle, it was necessary to provide a scalable and robust solution that is process independent. This entire work could be split as being able to achieve four objectives and they are as follows:

1. **Presenting a detailed model of knowledge management for the customer support process in product development industries. Providing a gap analysis study between the existing system for knowledge management and the ideal system for knowledge management for the customer support process.**

As mentioned above, in order to understand the fallacies in knowledge management process, it is necessary to clearly specify the existing model for customer support process. During this process an attempt was made to answer the following questions.

- Who are the necessary parties involved for the customer support process?
- What are the necessary sequences of events that take place in order to complete a successful customer support cycle?
- What is the ideal expected behavior of the customer support process?
- What are the tools used at each level of the customer support lifecycle that promise to assure knowledge management for the organizations?
- In what ways does the existing system deviate from its ideal behavior?
- What are the gaps between the existing system for knowledge management and the ideal system, also mentioning the ways to bridge these gaps?

Chapter 3 describes in detail regarding this existing system of knowledge management and its fallacies, the ideal knowledge management scenario and the gaps that exists between the two models. This chapter would give insight on how the existing system deviates from its ideal behavior and enumerates the reasons for its inefficient behavior.

2. Analyzing gaps in the previous objective and tailoring applications that address these gaps.

This objective deals with analysis of reasons that lead to the inefficiency of the existing system for knowledge management. In this objective the following questions are addressed:

- Which gaps are interrelated? Why?
- Are there applications (already existing, off the shelf) that address these interrelated gaps?

- How can applications be tailored so that they do not hamper the harmony of existing systems? More specifically how can bridge applications be developed that integrate with the existing knowledge management systems with ease?

Chapter 4 gives a detailed description of applications that are identified to address the gaps. (Results of the gap analysis study) A detailed description of the overlap between the tasks they perform is analyzed and applications are customized to be task specific yet inter operable.

3. **Identify off the shelf components that can be integrated to implement the proposed model, implementing a concept in the existing system to make the process run better. (Implementing the concept of similar pages in the already existing Enterprise search application)**

The input from the previous stage is taken as the proposed model of knowledge management for end to end customer support process. The designs of the bridge applications are considered and a quick survey is performed to see if these bridge applications correspond to already existing off the shelf components.

The rationale behind this objective is to integrate those off the shelf components so that the proposed model can be implemented.

Along with this aspect, the present objective also aims at improving the efficiency of an Enterprise search application by providing the feature of similar pages.

It is observed during the first phase of this objective that a core component of this process is an Enterprise search engine (that can be tailored to our needs). Based on this observation, an off the shelf Enterprise search engine is configured to respond to the tasks at hand.

An elaborate discussion of this process is mentioned is chapter 5. Chapter 5 explains in detail of the methodology used to implement the similar pages feature, implementation details and results.

The section to follow would give an overview about the kind of research methodologies in general and would also describe the methodology used to perform the aforementioned tasks.

1.3 Research Methodology

*"**All research ultimately has a qualitative grounding**"*

- Donald Campbell

"***There's no such thing as qualitative data. Everything is either 1 or 0***"

- Fred Kerlinger

Research aims for the advancement of knowledge and the theoretical understanding of concepts [12]. Research is often powered by the researcher's curiosity and intuition. It is therefore an exploratory process [13]. It is generally carried on without a specific practical end in mind. However, it may have unexpected results that may lead to practical applications. Research is a considered to be a subset of invention [12].

Typically the structure of research is constant [14]. The following steps are usually part of research:

- Formation of the topic
- Hypothesis
- Conceptual definitions
- Operational definitions
- Gathering of data
- Analysis of data
- Test, revising of hypothesis
- Conclusion, iteration if necessary

Research broadly falls under three categories namely:

- **Exploratory research** [12]

  Newer problems are identified in this methodology and efforts are put to solve them.

- **Constructive research** [12]

  This is more traditional research in which solutions to a problem are developed.

- **Empirical research** [12]

  This methodology tests the feasibility of a solution using empirical evidences.

Another common classification of research is describing it as qualitative and quantitative research [17]. Quantitative research deals with:

- The generation of models, theories and hypotheses
- The development of instruments and methods for measurement
- Experimental control and manipulation of variables
- Collection of empirical data
- Modeling and analysis of data
- Evaluation of results

In Quantitative research, evidence is evaluated, theories and hypotheses are refined, technical advances are made, and so on. In addition to this, Quantitative research using statistical methods is common. It typically begins with the collection of data based on a theory or hypothesis, followed by the application of statistical methods [ref].

In contrast, qualitative research focuses on understanding the richness, depth, and complexity of phenomena [14] [15]. Qualitative research means "any kind of research that produces findings not arrived at by means of statistical procedures or other means of

quantification" [16]. A few variants of qualitative research are used for the studies of the following:

- ✓ Case study
- ✓ Grounded theory
- ✓ Phenomenology
- ✓ Ethnography
- ✓ Historical

The work presented here can be described in two important phases. The first phase consists of detailed study of the existing models and performing gap analysis studies, proposing design models for applications that can integrate with the existing systems and rolling out a complete knowledge management framework for the entire customer support process. This phase is undoubtedly qualitative.

However, the second phase consists of implementing a feature in the existing model that enhances its performance. This phase consists of evaluating the algorithm with data collected from different information sources of the organization. Results will be computed and tabulated for further analysis. This phase of work is regarded as quantitative analysis or methodology of research.

1.4 Organization of Thesis

This section describes the organization of the chapters in this thesis. The entire thesis is organized into seven chapters.

Chapter 1 deals with presenting the motivation for the work, followed by the description of objectives of this work. This chapter also introduces the common research methodologies and comments on the research methodology taken up for this work.

Chapter 2 provides with a detailed survey of the standard knowledge management techniques. This is followed by explaining the meaning of customer support process in

Industries. This chapter also presents existing work in this direction, namely, tools used for specific tasks that claim to address the knowledge management issues of organizations.

Chapter 3 provides information about the existing model of customer support process in the organization being considered. This chapter also proposes an Ideal knowledge management scenario for the same process and provides with a detailed gap analysis study that brings out the deficiencies of the current system.

Chapter 4 analyses these gaps and makes an attempt to tailor applications that act as bridges which nullify the detrimental effect caused by these gaps. A detailed description of each of these applications is provided along with their working details and manner in which it gets integrated with the existing system.

Chapter 5 revisits chapter 4 and provides the entire knowledge management framework for the support process. In this chapter, the core issues of data management and data organization are discussed.

In chapter 6, a hierarchical organization of Industrial data is presented as the effective means of achieving progress with respect to knowledge management. (Retrieving the right information at the right time)

Chapter 7 provides the implementation details of the feature of similar pages in the existing Enterprise search engine. The methodology used to find out similar pages and architecture used to implement the feature is also provided in this chapter. Results are tabulated and important conclusions are listed. This chapter also gives a brief description of the direction in which this project heads in the future.

Chapter 2

Background: Knowledge management and Engineering support cycle

This chapter introduces the concepts of knowledge management. The chapter begins with exploring the frameworks of knowledge and the means to access knowledge. The later sections explain in detail about the customer support process. A brief description of tools used specifically for the knowledge management process in the Industry scenario is also presented. State of the art research that points out key factors for the failure of knowledge management systems is also presented.

2.1 Concise summary of Knowledge management

Knowledge Management (KM) comprises a range of practices used in an organization to identify, create, represent, distribute and enable adoption of insights and experiences [2] [7] [8]. Such insights and experiences comprise knowledge, either embodied in individuals or embedded in organizational processes or practice.

Many large companies and non-profit organizations have resources dedicated to internal KM efforts, often as a part of their 'Business Strategy', 'Information Technology', or 'Human Resource Management' departments. [10]

KM efforts typically focus on organizational objectives such as improved performance, competitive advantage, innovation, the sharing of lessons learned, and continuous improvement of the organization [3] [4].

KM efforts can help individuals and groups to share valuable organizational insights, to reduce redundant work, to avoid reinventing the wheel per se, to reduce training time for new employees, to retain intellectual capital as employee’s turnover in an organization, and to adapt to changing environments and markets.

2.1.1 Insight into Knowledge: Frameworks

Different frameworks for distinguishing between knowledge exist. One proposed framework for categorizing the dimensions of knowledge distinguishes between tacit knowledge and explicit knowledge. [2]

Tacit knowledge represents internalized knowledge that an individual may not be consciously aware of how he or she accomplishes particular tasks.

At the opposite end of the spectrum, explicit knowledge represents knowledge that the individual holds consciously in mental focus, in a form that can easily be communicated to others.

Early research suggested that a successful KM effort needs to convert internalized tacit knowledge into explicit knowledge in order to share it [1], but the same effort must also permit individuals to internalize and make personally meaningful any codified knowledge retrieved from the KM effort.

A second proposed framework for categorizing the dimensions of knowledge distinguishes between embedded knowledge of a system outside of a human individual [8] (e.g., an information system may have knowledge embedded into its design) and embodied knowledge representing a learned capability of a human body's nervous and endocrine systems.

A third proposed framework for categorizing the dimensions of knowledge distinguishes between the exploratory creation of "new knowledge" (i.e., innovation) vs. the transfer or exploitation of "established knowledge" within a group, organization, or community. Collaborative environments such as communities of practice or the use of social computing tools can be used for both knowledge creation and transfer. [2] [4] [9]

2.1.2 Insight into Knowledge: Knowledge access

Knowledge may be accessed at three stages: before, during, or after KM-related activities. Different organizations have tried various knowledge capture incentives, including making content submission mandatory and incorporating rewards into performance measurement plans [1]. Considerable controversy exists over whether incentives work or not in this field and no consensus has emerged. [6]

One strategy to KM involves actively managing knowledge (push strategy). In such an instance, individuals strive to explicitly encode their knowledge into a shared knowledge repository, such as a database, as well as retrieving knowledge they need that other individuals have provided to the repository [9].

Another strategy to KM involves individuals making knowledge requests of experts associated with a particular subject on an ad hoc basis (pull strategy) [3]. In such an instance, expert individual(s) can provide their insights to the particular person or people needing this.

A few knowledge management strategies that companies usually follow are listed below: [1]

- Rewards (as a means of motivating for knowledge sharing)
- Storytelling (as a means of transferring tacit knowledge)
- Cross-project learning
- Knowledge mapping (a map of knowledge repositories within a company accessible by all)
- Communities of practice
- Best practice transfer
- Collaborative technologies (groupware, etc)
- Knowledge repositories (databases, etc)

- measuring and reporting intellectual capital (a way of making explicit knowledge for companies)
- knowledge brokers (some organizational members take on responsibility for a specific "field" and act as first reference on whom to talk about a specific subject)
- Social software (wikis, social bookmarking, blogs, etc)

2.1.3 Insight into Knowledge management: Motivational factors for Knowledge management

Given below is a list of considerations (thought not exhaustive) driving a KM effort: Typically organizations are lured (under statement) to processes that adapt knowledge management for the below reasons: [9] [11]

- ✓ Making available increased knowledge content in the development and provision of products and services
- ✓ Achieving shorter new product development cycles
- ✓ Facilitating and managing innovation and organizational learning
- ✓ Leveraging the expertise of people across the organization
- ✓ Increasing network connectivity between internal and external individuals
- ✓ Managing business environments and allowing employees to obtain relevant insights and ideas appropriate to their work
- ✓ Solving intractable or wicked problems
- ✓ Managing intellectual capital and intellectual assets in the workforce (such as the expertise and know-how possessed by key individuals)

Overall, it is a proven fact that organizations that succeed in implementing well defined knowledge management frameworks are proven market leaders in terms of efficiency, productivity and innovativeness.

With the consciousness of this buzz word in the industry, organizations have used multiple task specific applications, which mean that organizational content can be very well identified as information sources. However, these organizations are still not in a position to exploit the above mentioned advantages of knowledge management.

In short Knowledge management for an organization can be considered as identifying information sources and leveraging it to the needs of the organization [1]. More specifically, aim of knowledge management should be to provide the right information to the right person at the right time.

2.2 Engineering support: What do we mean by this?

"Life cycle of a product or process actually begins with its support to the customers and the degree of satisfaction they show back to the organization" – Anonymous

Organizations are basically categorized based on the kind of services they provide. At the top of organizational categorization we find that organizations are classified as product based or process based companies.

The basic difference between a product based company and a process based company is that product based companies analyze the external environment of the customers and come up with a product that meets their needs.

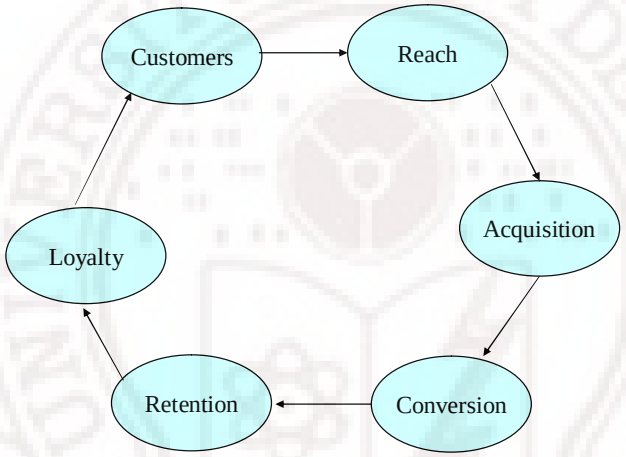
A process based company on the other hand responds to the customers by providing them well defined and structured services based on their need. More specifically, in process based companies the customer approaches the company with his needs and a customized solution will be given to him.

However, in both these organizational behaviors, an important phase is to provide assistance to their respective customers. This phase becomes inevitable because of the following reasons:

1) Customers' requirements change over time.
2) There is always a mismatch between the customers' version of requirements and developers design of the product or service.
3) Over time the developed product (process) needs maintenance in order to assure its robustness and flexibility.

A typical customer support life cycle can be described with the figure described below:

Customer support life cycle



(Jim Sternly and Matt cutler, 1998)

As described in the aforementioned figure, the organizations' success is determined by how effective each of these phases is completed. An acquired customer has to be converted into a loyal customer and support becomes very crucial for this reason.

The support process of an organization is divided into three crucial phases namely:

1) Pre Installation support
2) Installation support
3) Post Installation support

During these phases the customer expects that the organization shows timely response to their queries and sorts out their problems. Usually many customers are satisfied if the organizations provide support to properly install the service. (Make the product or process usable)

The figure below describes these phases along with the percentage of issues that an organization receives at each phase.

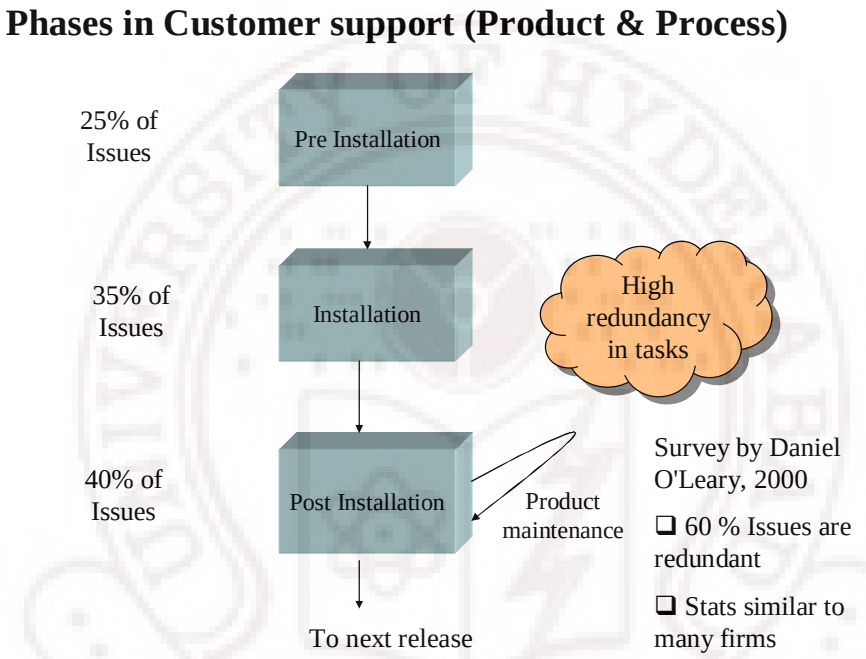


Figure: Phases in customer support

Customer Support in broader sense usually points to the support the organization gives the customer during the post installation.

The above figure also shows the statistical survey presented by Daniel O'Leary [6] who gives statistical evidence for the percentage of issues faced at each of these phases.

According to the author, 25% of issues [6] are faced by the customers during the pre Installation phase. These include information about the system specifications or requirements, platform information, customer requirement specifications etc.

The authors' survey revealed that about 35% of issues [6] are faced by the customers during the installation phase. This is a commonplace considering the fact that most of the customers are not adept at dealing with high end products and services.

Added to that if the product is new in the market, then the organizations have to carry out demonstrations to show its customer population the specifics about its installation.

As mentioned above, the customer support cycle begins at the post installation stage. Post installation support determines the success of a product o a process for the organization.

Post installation support involves responding to customers by accommodating their new requirements, handling problems with respect to a scenario where the service behaves abnormally, improving the service by constantly adding new features to the service and shaping it to be even more flexible and robust to the customers' needs [6]. This cycle is usually termed as a release of the product or process. A product usually consists of multiple releases.

In order to provide support to the customers, the organizations need to have a well defined structure internally. This structure should ensure that the issue (faced by customer) reaches the respective employee responsible to fix it. In addition to this functionality the organization must also take care of the fact that the customers need to have a single window access to report their problems.

The organizations typically maintain a three tier hierarchy for the support process. The specifics about this process will be elaborated in the next chapter. Huge content is generated due to the interactions that happen for customer support. Organizations use specialized systems to track, maintain and retrieve back (as need be) the content for each interaction. A list of various systems (though not exhaustive) used by the organizations is provided below:

1) Customer Relationship Management Systems
2) Defect Management Systems
3) Content Management Systems
4) Document Management Systems
5) Application Lifecycle Management Systems

A brief description of each of these applications and organizations that provide efficient tools for each of the application is provided in the section to follow.

2.2.1. Customer Relationship Management Systems

Customer relationship management (CRM) consists of the processes a company uses to track and organize its contacts with its current and prospective customers [18]. CRM software is used to support these processes; information about customers and customer interactions can be entered, stored and accessed by employees in different company departments. Typical CRM goals are to improve services provided to customers, and to use customer contact information for targeted marketing [19].

From the outside, customers interacting with a company perceive the business as a single entity, despite often interacting with a number of employees in different roles and departments.

CRM is a combination of policies, processes, and strategies implemented by an organization to unify its customer interactions and provide a means to track customer information [11]. It involves the use of technology in attracting new and profitable customers, while forming tighter bonds with existing ones.

CRM includes many aspects which relate directly to one another:

- ✓ Front office operations — Direct interaction with customers, e.g. face to face meetings, phone calls, e-mail, online services etc.

- ✓ Back office operations — Operations that ultimately affect the activities of the front office (e.g., billing, maintenance, planning, marketing, advertising, finance, manufacturing, etc.)

- ✓ Business relationships — Interaction with other companies and partners, such as suppliers/vendors and retail outlets/distributors, industry networks (lobbying groups, trade associations). This external network supports front and back office activities.

- ✓ Analysis — Key CRM data can be analyzed in order to plan target-marketing campaigns, conceive business strategies, and judge the success of CRM activities (e.g., market share, number and types of customers, revenue, profitability).

There are basically four types of CRM's, [18]

- ➢ Operational CRM
- ➢ Analytical CRM
- ➢ Sales Intelligence CRM
- ➢ Collaborative CRM

The following table lists the top software vendors for CRM projects completed in 2006 using external consultants and system integrators, according to a 2007 Gartner study.

Vendor	**Percentage of implementations**
Siebel (Oracle)	41%
SAP	8%
Epiphany (Infor)	3%
Oracle	3%
PeopleSoft (Oracle)	2%
Salesforce.com	2%
Amdocs	1%

Chordiant	1%
Microsoft	1%
SAS	1%
Others	15%
None	22%

2.2.2. Defect Management Systems

Defects management system is a "Defect Tracking System" which maintains a database of problem reports. Defects management system allows individuals or groups of developers and consultants working on same project to keep track of outstanding defects in their project effectively.

Defects management system can track defects and design changes, communicate with teammates, submit and review patches, and manage quality assurance.

The defect management systems help the organizations to gain benefits that include: [20]

- DMS improves communications between customers, support staff and developers.
- Increases the quality of the projects.
- Increases overall productivity.
- Facilitates easy bug tracking.
- Strives to eliminate redundancy in tasks.

The table below lists down frequently used defect tracking / management tools. [20]

Tool name	Platform	Tool vendor/site	Comments
Issue Tracking Anywhere	Web browser	Dynamsoft	Web-based bug tracking system designed for issue / work item tracking, bug tracking, customer support and project management
RADAR	web browser	Cosmonet Solutions	defect management and analysis tool

Tool name	Platform	Tool vendor/site	Comments
SpiraTest	web browser	Inflectra	Manages requirements, tests, bugs and issues in one environment, with complete traceability from inception to completion
yKAP	Windows	DCom Solutions	Defect and issue tracking and collaboration tool
VisionProject	web	Visionera	Web based defect tracking and support center tool
SoftwarePlanner		Pragmatic Software	Manages requirements, project tasks, issues, test cases
SilkCentral Issue Manager		Borland	Issue management system
RMTrack	web	RMTrack	Issue management software
Quality Assurance Studio	Windows LotusNotes	Objentis	Testing and defect tracking tool
ProblemTracker	web	NetResults Corporation	Web-based collaboration software for bug tracking, change management, support, and help desk
PR-Tracker		Softwise Company	Defect tracking tool
Ozibug	web	Tortuga Technologies	Web based defect tracking tool
Issue View	Windows	Issue View	Defect tracking tool
Bug Tracking	web	Elementool	Web based defect tracking tool
Deskzilla		ALMWorks	Desktop client for Bugzilla
DefectTracker	web, Windows	Pragmatic Software	Defect tracking software
Defect Manager	web, Windows	Tiera Software	Tracks, bugs, defects, calls, tasks and enhancements
CustomerFirst	Windows	rti Software	Defect tracking and help desk software
Census	web	MetaQuest	Census is a scaleable, Web-based system for bug tracking, defect tracking, enhancements, support calls, timesheets and more
	web	WEBsina	Bugzilla based defect tracking system

Tool name	Platform	Tool vendor/site	Comments
Bugzero			
BugStation		BUGOPOLIS	Bugzilla based defect tracking system
BugSentry	Windows	IT Collaborate	Bug tracking for .NET and COM
BugMonitor	web	BugMonitor	Web based bug tracking system
BugHost	web	BugHost.com	Web based bug tracking system
Buggy	Windows	Novosys	Tracks bugs, feature requests (enhancements), support requests and to-do.
Bugcentral	web	BugCentral.com	Web-based defect tracking system
Bug-Track.com	web	Bug-Track.com	Web-based defect tracking software
Aspire Swatter		Aspire Managed Solutions	Defect management system
AQdevTeam	Windows	Automated QA	Issue tracking and project management system
PureCM	Windows, Unix, Linus, Mac	PureCM.com	SCM and defect tracking system
ProjectLocker Issue Management	web	ProjectLocker	Web based issue management system
Ken Testman	web browser	http://www.kentestman.com/	Defect management tool, among other things
IssueView		IssueView.Com	Issue management tool
Dragnet		SourceGear	Web-based bug-tracking system
JIRA		Atlassian	Issue management tool
OnTime	Windows	Axosoft	Defect (and feature and task) tracking tool
Active! Focus		Falafel Software	Issue (and requirements and project and risk) management tool
SpeeDEV IM	Windows	SpeeDev	Issue management tool
Team Track	Windows/web	Serena	Issue tracking tool
Rally	ASP/web browser	Rally Software Development Corp.	Defect (and requirements and project) management tool
FogBUGZ	Windows, Unix, Mac	Fog Greek Software	Defect and feature request management tool

Tool name	Platform	Tool vendor/site	Comments
Polaris	Windows	aicas GmbH	Workflow and issue tracking tool
20s Change Coordinator	Excel	20smackers	Manage the process and communication thread associated with change requests
e11 Help Desk Software	Web based	e11online.com	Bug tracking, help desk, etc.
Bugzilla	Linux, Unix, Windows	Bugzilla	Open-source defect management system
ExtraView		sesame technology	Web based defect tracking, help desk and customer support tool
BugBase	Windows	BugBase	Open source bug reporting system
TrackRecord	Windows	Compuware	Defect handling system, change request management
Visual Intercept	Windows	Ellsinore Technologies	Defect management system
Track	Windows 3, NT, 95, OS/2	Soffront, Inc.	Defect management system
TestTrack	Windows, Mac	Seapine Software	Defect management system
Remedy Change Management	Windows	Remedy Corporation	Change management
Remedy Action Request System	Windows	Remedy Corporation	Defect management system
PVCS Tracker	Windows	Merant, Inc.	Defect management system
McCabe TRUEtrack	Windows, Unix	McCabe and Associates	Change/problem/defect tracking
QuickBugs	Windows	Excel Software	Defect management system
DevTrack		TechExcel	Defect management system
PR-Tracker	Windows	Softwise Company	Defect management system over Internet and local network in one database
ClearQuest	Windows, Unix	Rational Software	Defect management system
Gran PM	Web	TrackStudio	Web based defect management system

Tool name	Platform	Tool vendor/site	Comments
Requeste	Windows, Linux	Sysart	Issue management system
BUGtrack	Windows	SkyTech	Defect management system

2.2.3. Content Management Systems

A content management system (CMS) is a computer application used to manage work flow needed to collaboratively create, edit, review, index, search, publish and archive various kinds of digital media and electronic text [21].

CMS' are frequently used for storing, controlling, versioning, and publishing industry-specific documentation such as news articles, operators' manuals, technical manuals, sales guides, and marketing brochures [21] [22].

The content managed may include computer files, image media, audio files, video files, electronic documents, and Web content. These concepts represent integrated and interdependent layers.

There are various nomenclatures known in this area: Web Content Management, Digital Asset Management, Digital Records Management, Electronic Content Management and so on. The bottom line for these systems is managing content and publishing, with a workflow if required.

A CMS may support the following features: [22]

- Identification of all key users and their content management roles.
- The ability to assign roles and responsibilities to different content categories or types.

- Definition of workflow tasks for collaborative creation, often coupled with event messaging so that content managers are alerted to changes in content. (For example, a content creator submits a story, which is published only after the copy editor revises it and the editor-in-chief approves it.)
- The ability to track and manage multiple versions of a single instance of content.
- The ability to capture content (e.g., scanning).
- The ability to publish the content to a repository to support access to the content. (Increasingly, the repository is an inherent part of the system, and incorporates enterprise search and retrieval.) Hence, material can be re-factored for new uses. (E.g., use the same base content in different ways for desktop browsers, mobile browsers, and print output.)

There are four main categories of CMS, with their respective domains of use: [21] [22]

- Enterprise CMS (ECMS)
- Web CMS (WCMS)
- Document management system (DMS)
- Mobile CMS
- Component CMS

The table below depicts the market leaders in Content Management Systems: [21]

Name	**Platform**	**Supported Data**	**Stable Release**
Blue Light CMS	Java	Oracle, MySQL	1.1
Cascade Server	Java	Oracle, MySQL, SQL Server	5.7
CoreMedia CMS	Java	Oracle, SQL Server, PostgreSQL, DB2	2008
Day Communiqué WCM	Java	no database required uses JSR-170-compliant content repository	5.1
FatWire Content Server	Java	Oracle, SQL Server, DB2	7.0.3

Noodle	Java	Oracle, SQL Server, PostgreSQL	6.6.4
TerminalFour Site Manage	Java, J2EE	Oracle, MySQL, SQL Server	6.0
Accrisoft Freedom	PHP	MySQL	6.0

2.2.4. Document Management Systems

A document management system (DMS) is a computer system (or set of computer programs) used to track and store electronic documents and/or images of paper documents [21].

The term has some overlap with the concepts of content management systems and is often viewed as a component of enterprise content management (ECM) systems and related to digital asset management, document imaging, workflow systems and records management systems [22].

In the broadest sense, document management systems can range from a shoebox all the way to an enterprise content management system.

There are several common issues that are involved in managing documents, whether the system is an informal, ad-hoc, paper-based method for one person or if it is a formal, structured, computer enhanced system for many people across multiple offices.

Most methods for managing documents address location, filing, retrieval, security, disaster recovery, retention period, archiving, distribution, workflow, creation, authentication and traceability.

Document management systems commonly provide storage, versioning, metadata, security, as well as indexing and retrieval capabilities. The components of a Document management system are listed below: [21]

- Metadata
- Integration
- Capture
- Indexing
- Storage
- Retrieval
- Distribution
- Security
- Workflow
- Collaboration
- Versioning
- Publishing

Given below is a list of Document management systems that are commonly used by organizations world wide. [22]

- ✓ SoluSoft N2 Document Management and Workflow System
- ✓ Autonomy iManage by Autonomy Corporation (Interwoven Worksite)
- ✓ Cognidox by Cognidox
- ✓ Document Locator by ColumbiaSoft
- ✓ Content Manager by IBM
- ✓ ViewWise by Computhink
- ✓ DocPoint by DocPoint.biz
- ✓ DocumentMall.com by Ricoh-usa.com
- ✓ DMS by MetricStream
- ✓ FileNet by IBM
- ✓ ImageNow by Perceptive Software
- ✓ ImagePlus by IBM
- ✓ Infonic Document Manager by Infonic
- ✓ Invu by Invu

- ✓ ISIS Papyrus by ISIS Papyrus
- ✓ Laserfiche DMS by Laserfiche
- ✓ Objective Solution by Objective Corporation

2.2.5. Application lifecycle management systems

Application lifecycle management (ALM) is the marriage of business management to software engineering made possible by tools that facilitate and integrate requirements management, architecture, coding, testing, tracking, and release management. [23]

There are quite a few advantages of the ALM systems. A few of them are mentioned here: [24]

- o Increases productivity, as the team shares best practices for development and deployment, and developers need focus only on current business requirements
- o Improves quality, so the final application meets the needs and expectations of users
- o Breaks boundaries through collaboration and smooth information flow
- o Accelerates development through simplified integration
- o Cuts maintenance time by synchronizing application and design
- o Maximizes investments in skills, processes, and technologies
- o Increases flexibility by reducing the time it takes to build and adapt applications that support new business initiatives

Although the industry has not yet formally defined what precisely constitutes an ALM tool, the generally accepted categories include: [24]

- Requirements visualization
- Requirements management
- Feature management

- Modeling
- Design
- Project Management
- Change management
- Configuration Management
- Software Information Management (for ALM Tool Integration)
- Build management
- Testing
- Release Management
- Software Deployment
- Issue management
- Monitoring and reporting
- Workflow

Rally, Neuma CM+, Parasoft Concerto, Lighthouse, Artisan Studio - Collaborative modeling tool suite, StarTeam - Change and Configuration Management are the leading application lifecycle management tools available in the market. [23]

In spite of there being so many specialized tools for knowledge management, their impact for processes like customer support has been feeble. The above mentioned tools are specific to a process within the entire Customer support process. Most of the times, these tools are incompatible with each other thereby resulting in information loss. More specifically, the output of one tool should be ideally used by another tool if right information has to be identified by the employees in the organization. As this fails to happen in the real world, these tools though effective in their own terms fail to make an impact for the knowledge management for the entire customer support process.

Chapter 3

Knowledge Management for Engineering Support

This chapter explains in detail about the existing knowledge management scenario in an organization for customer support process. Later in the chapter the reasons for inefficiency of this model will be enumerated. The proposed work will be presented thereof.

3.1 Existing knowledge management system for engineering support

This chapter attempts to ease the process of understanding the proposed knowledge management framework. Later in this chapter a systematic knowledge management (KM) framework for end to end customer support process will be unrevealed.

This section focuses on the Existing systems for knowledge management that comprises of a set of task specific models. These task specific applications are incompatible and often fail to integrate [5]. The reason for this situation is lack of a generalized KM framework that fails to clearly specify the tasks of individual applications.

The purpose of this KM framework is two fold

- Semi automating the customer-support process.
- Eliminating the issue related to multiple application incompatibility.
- Addressing the problem of recurrence/redundant problems.

This section analyses the current model for knowledge management used across many organizations. It also mentions the gaps or pitfalls of the current model by comparing it with a presumed ideal knowledge management scenario.

The current model for knowledge management can be described in figure 1. Generally, many organizations work based on this three tier model for customer support.

In this three tier architecture, the lead role players are the customers, the customer support executives and the developers or the maintenance engineers. [5] [31]

The current model of end to end support in an organization can be understood from the figure below

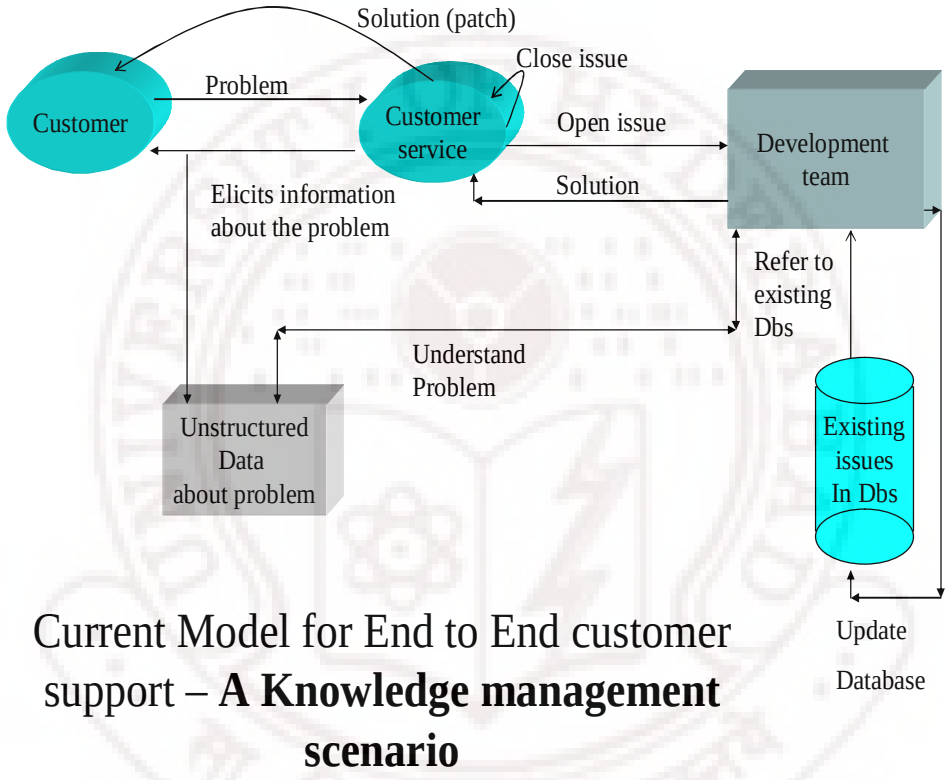


Figure 1: Current model of end to end customer support

The customer reports his problem to a Customer Service Executive. The Customer Service Executive captures additional details of the problem and opens forwards the issue to the maintenance team (development team).

The maintenance team uses the information provided by the support executive and details in the corresponding communication system to understand the problem. Later the development team performs a search on the existing database of issues.

If a similar issue is found, the customer will be provided with similar solution otherwise corresponding code segment will be monitored and changes as required will be made to the code and sent to the customer through customer service executive. This is the most favored model for end to end customer support. Not surprisingly, this model has drawbacks that greatly affects on the performance of the teams involved.

The main reason for the failure of this model is due to its people dependence rather than on documented knowledge [3]. The problem with this model is that the knowledge possessed by an individual is lost once the individual decides to leave the organization or move internally to another team.

In order to overcome this issue, the defects are all documented and maintained in defect management systems. However, as we shall see later in the chapter, these pose more problems than to help.

3.2 Ideal system for knowledge management

Previous attempts to provide an end to end knowledge management system for customer support process have identified that the information content present in the organization should grow based on the interactions that happen between various role players in the customer support process.

The figure given below describes a knowledge management scenario where the parties involved in various transactions communicate through a centralized repository there by facilitating ease of data organization and retrieval for future needs. (Finding records that handled a similar issue etc) Based on this module and an assumption regarding the ideal knowledge management scenario that it depends on documented knowledge we have designed an “Ideal” knowledge management scenario for the support process. The figure below describes the base model and proposed model for ideal behavior of a knowledge management framework. [31]

Techconnect system: KM schema adopted by most of the organizations

(Dorothy, 2003)

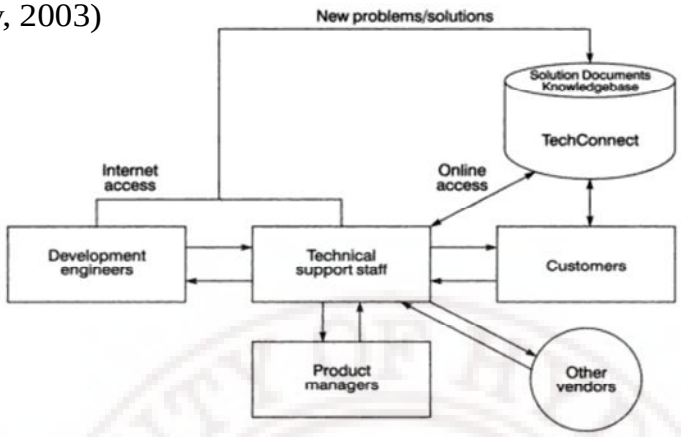


Figure 2: Base model for proposing Ideal Knowledge management scenario

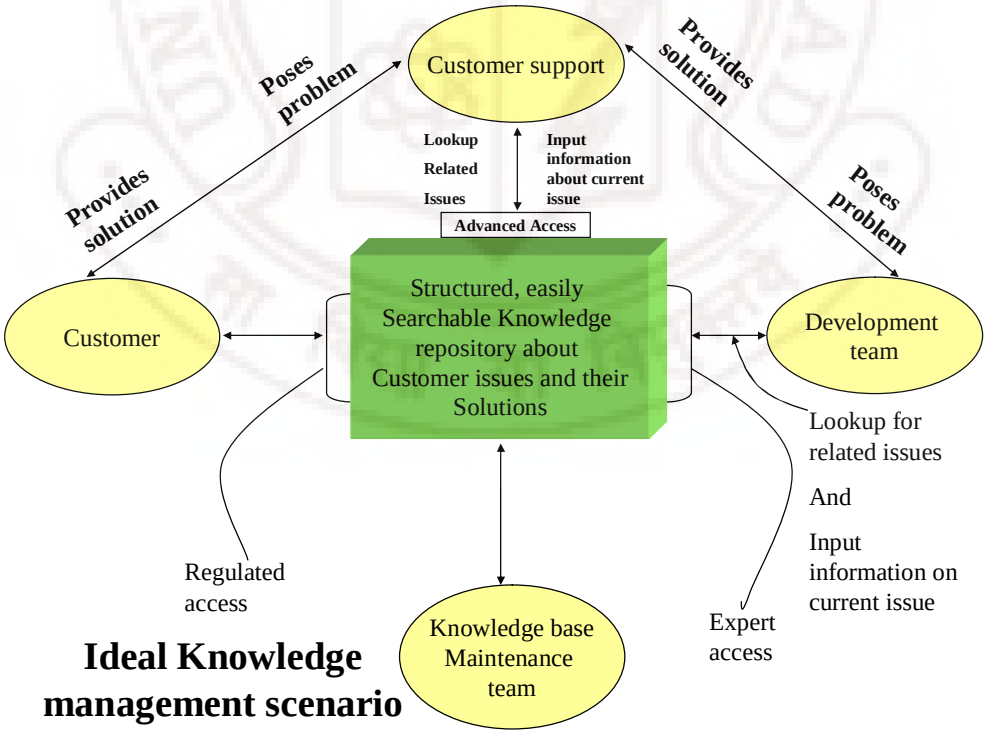


Figure 3: Ideal Knowledge management scenario for Customer support process

We have felt it necessary to consider that there be a centralized knowledge repository that grows as a result of the communication happening between various role players in the process. They in turn interact with the knowledge repository for performing their tasks.

This ideal model reveals that the knowledge of the organization is not tied with individuals but grows with interactions between the role players. In this model the customer will be able to communicate his problem with the customer support executive by accessing the centralized knowledge repository.

The repository has different levels of access to the customer, the customer support executive and the developer. The privileges given to the customer are far less compared to those given to the customer support executive with the developer having total access permission.

This difference in access levels is to help the organization shield its confidential information from its customers and other potential intruders.

The customer support executive has an advanced access to the data repository so that he will be able to search the repository for issues related to the customers' present problem. This action would imply that the customer support executive is empowered with access to the repository which he cannot do in the existing system.

An added advantage due to this action would be that the developer can expect well structured information regarding the issue from the customer support executive. Apart from this, the customer support executive will also be able to provide references to previous issues that were of same nature.

These actions would assure that the time spent by the developer to address an issue would drastically come down thereby improving the performance and productivity.

The next section would give the results of gap analysis study between the existing model for Knowledge management and the ideal model.

3.3 Existing system Vs Ideal System: A Gap analysis study

A gap analysis study is presented below in order to understand the problems with the existing system.

The following differences have been noticed and documented.

- Existing process is people dependent and not documented knowledge dependent. [31]
- No structured way of eliciting information from the customer.
- Customer is not empowered with access to existing knowledge. [5]
- Customer support executives do not contribute to knowledge; not empowered with access to repository. (They could potentially filter many problems at the initial stage reducing the response time)
- No structured way of capturing developer's knowledge.
- Developer/ management team's search process is not guided or vague. (Proper access to resources is absent)
- Absence of a maintenance team for the knowledge repository. (These deal with issues related to updating, addition and deletion of records to the repository and check for its consistency)
- Temporal split of data does not exist. (Older data VS Newer data)

These were identified as the problems to be addressed and fixed in order to improve the performance and productivity of the support processes. Each of these differences will be briefly explained below.

The existing system is people dependent where it should ideally documented knowledge dependent. In order to understand this concept better, consider a scenario where an

employee dealing a critical project moves from an organization into another organization. In this case the knowledge transfer processes used by the organizations are crude and naïve. This hampers the growth of project and also injects infinite delays in responding to user queries.

In contrast if the knowledge possessed by the customer is simultaneously documented (not manual documentation however, using tools that define communication mode for a particular transaction) then irrespective of his presence, knowledge about the project is available anytime to any employee.

The interaction between the customer and customer support executive is not uniform. The details elicited from the customers by support personnel are fairly different from another support executive.

This requirement is not as simple as it appears. It is not just the standardization of the questions to be asked however it is to deal with how the support personnel is able to ask better questions each time. Ideally, the support personnel should be able to trace the root cause of the customer through this process.

Added to this, when the developer views the conversation between the support executive and the customer he should clearly be able to understand the requirement without further delays in interpreting the conversation. This sadly is not the case with the existing systems.

A toast of customers' behavior is his panic when the application fails to behave as expected. This behavior of the customer makes sense as his investment of time; effort and money have gone in order to get the product that can behave as expected.

For naïve issues like installation, configuration of environment, issues dealing with modes of installation and likes it is not ideal that the customer approaches for support. In the existing system, the number of issues like this is definitely a significant number.

This is waste of productive time for both the customer and the developer as they are spending time on issues that cannot be treated as their core competencies.

If the customers are provided access to a centralized knowledge repository that gives them structured and well defined information about the frequently asked questions, manuals, user guides etc that would greatly reduce the number of issues reaching the developer.

Issues that the customers cannot handle themselves can be filtered out and resolved at the customer support level. However, the existing system does not provide the necessary powers to the support personnel to contribute to the knowledge of the repository.

Ideally, the customer support executive should be able to access the knowledge repository with access privileges better than customers so that they can determine similar issues and guide the developer to reach the solution at a quick pace or they can filter out common / redundant tasks and provide the customers' quick fix to their issue.

Equally significant number of problems exists at the developers' end. The developer is not fed with structured information from the support personnel which results in long delays to figure out the exact nature of the problem.

Added to this the developers' knowledge has to be captured in specific tools that can come to rescue when another developer is dealing a similar issue. Tools used in this process are user dependent and most of the times the developers do not document the problems fixed by them.

All these problems can be resolved if a proper data management architecture exists. If each data item can be properly tagged with its meta-data, then issues can be automatically categorized when the user submits the problem. However, the existing systems do not even provide a temporal split of data.

The advantage of having a temporal split is that previous issues can be moved out of the repository and newer issues can be given preference. This would also help in meta-data tagging of the data items based on time. The existing system does not exploit this aspect.

3.4 Proposed Knowledge management framework

Based on the gap analysis study, we come up with a generalized KM framework that addresses the above mentioned issues. In our model we propose three applications namely a support process assistant, an advanced search mechanism and a DMS (Defect Management System) auto fill application and a data organization scheme.

Given below is a table that gives insight about the issues resolved by the applications:

Application name	I1	I2	I3	I4	I5	I6	I7	I8
Customer support Assistant	Yes	Yes	Yes	Yes	No	No	No	No
DMS auto fill application	No	No	No	Yes	Yes	Yes	No	No
Advanced search mechanism and data organization	Yes	No	No	Yes	No	Yes	Yes	Yes

Table 1: Issues addressed VS Applications.

I1-I8 → The problems numbered from 1 to 8 in the existing system.
Yes → The problem is addressed by the application.
No → The application does not address the problem

A common framework that integrates these applications with the existing system is presented is presented below. These applications make the transition of the existing system to the ideal knowledge management scenario easy.

The working and architectural specifications of these applications will be mentioned in the next chapter. We will briefly explain the purpose of each of these below.

Customer Support Assistant application is responsible for assisting the customer support executive by providing with a possible list of questions to be asked to the customer based on his profile and responses to previous questions.

This application ensures that information is elicited from the customer in a structured manner. [27] [28] [29]

Defect management is essential for all software development projects. Automated defect management tools have become commonplace.

The performance of these tools depends on the amount of data that is put in them. Then these applications gain momentum and integrate with the development and maintenance cycles of the project.

However, in many cases the data put is inaccurate or insufficient for various reasons. This finally results in not utilizing these tools to their full potential.

The Defect Management System (DMS) auto fill application uses the online and offline (Emails, chat transcripts and other information sources in the organization) communications between the employees to auto fill the templates of Defect Management System. [25] [26]

The organization of data in the knowledge base facilitates the working of these above mentioned applications [9].

A hierarchical data organization (will be presented later) aims to provide easy search and structured query retrieval. [30]

These applications find place in the existing system, meaning that they will integrate without hampering the harmony of the existing systems.

These are hence known as bridge applications. These applications make the information flow in the organization easy and will act as catalysts to enhance the working of existing frameworks.

The proposed framework of knowledge management for end to end customer support process is presented below

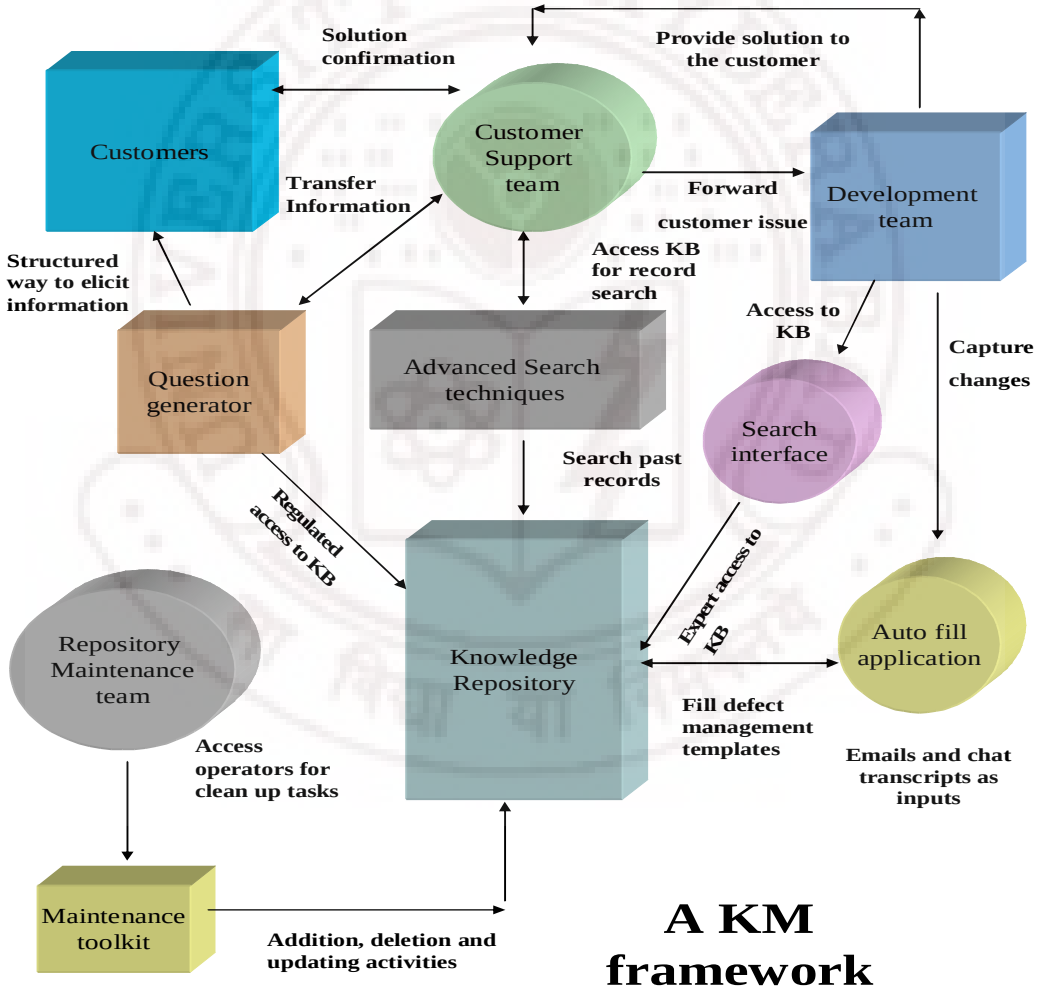


Figure 4: Proposed model of knowledge management

For End to End customer support

Chapter 4

Automated customer support assistant: A design proposal

This chapter presents in detail about the automated customer support assistant. The motivational factors for this application, Chatbots and question answering systems will be presented in the initial sections of the chapter. Later sections would present the detailed architecture (proposed design) and its working.

4.1 Motivation: Chatbots

The previous chapter disclosed the details about the proposed Knowledge management framework. The framework specifically mentioned the importance of bridge applications that address the gaps between the existing knowledge management frameworks and the Ideal knowledge management proposal.

This chapter is dedicated to describe one of the bridge applications' namely, the Customer support assistant.

Support Assistant application is responsible for assisting the customer support executive by providing him with a possible list of questions to be asked to the customer based on his profile and responses to previous questions. This application ensures that information is elicited from the customer in a structured manner.

The support assistant is a help application that assists the customer support executives throughout the organization to ensure standardization in questioning the customer.

Added to this basic functionality, the customer support assistant also prompts the support personnel a list of questions that be asked to the customer [33]. These questions are based on the profile of the customer and the previous issues he faced and the nature of the question asked.

The motivation for designing this application roots from chatbots [32] and question answering systems [34]. A brief overview of the state of the art Chatbots is presented in the next section.

Later sections will deal with the proposed design for this customer support assistant application, followed by detailed description of its working.

4.1.1 Chatbots: State of the art

A chatterbot (or chatbot) is a type of conversational agent, a computer program designed to simulate an intelligent conversation with one or more human users via auditory or textual methods [32].

The computer programmes are also known as Artificial Conversational Entity (ACE) [33] and, though many appear to be intelligently interpreting the human input prior to providing a response.

However, most chatterbots simply scan for keywords within the input and pull a reply with the most matching keywords or the most similar wording pattern from a local database [34]. Chatterbots may also be referred to as talk bots, chat bots, or chatterboxes.

A good understanding of a conversation is required to carry on a meaningful dialog but most chatterbots do not attempt this. Instead they "converse" by recognizing cue words or phrases from the human user, which allows them to use pre-prepared or pre-calculated responses which can move the conversation on in an apparently meaningful way without requiring them to know what they are talking about. [28]

Some modern AI research focuses on practical engineering tasks. This is known as weak AI and is distinguished from strong AI, which would require sapience and reasoning abilities.

One pertinent field of AI research is natural language processing (NLP). Usually, weak AI fields employ specialized software or programming languages created for them. [30]

For example, one of the 'most-human' natural language chatterbots, A.L.I.C.E., uses a programming language called AIML that is specific to its program, and its various clones, named Alicebots. Nevertheless, A.L.I.C.E. is still based on pattern matching without any reasoning. This is the same technique ELIZA, the first chatterbot, was using back in 1966. [27] [32]

Australian company MyCyberTwin also deals in strong AI, allowing users to create and sustain their own virtual personalities online. MyCyberTwin.com also works in a corporate setting, allowing companies to set up Virtual AI Assistants. [32] Another notable program, known as Jabberwacky [27], also deals in strong AI, as it is claimed to learn new responses based on user interactions, rather than being driven from a static database like many other existing chatterbots.

Although such programs show initial promise, many of the existing results in trying to tackle the problem of natural language still appear fairly poor, and it seems reasonable to state that there is currently no general purpose conversational artificial intelligence. This has led some software developers to focus more on the practical aspect of chatterbot technology - information retrieval. Automated Conversational Systems have progressed and evolved far from the original designs of the first widely used chatbots.

In the UK, large commercial entities such as Lloyds TSB, Royal Bank of Scotland, Renault, Citroën and One Railway are already utilizing Virtual Assistants to reduce expenditures on Call Centers and provide a first point of contact that can inform the user exactly of points of interest, provide support, capture data from the user and promote products for sale. [33] [34]

Notable architectures of applications that have proved to be successful Chatbots (Question answering systems) are given below.

State of the art Question Answering systems

Tequesta Question answering system TREC 10

(Christof Monoz, 2002)

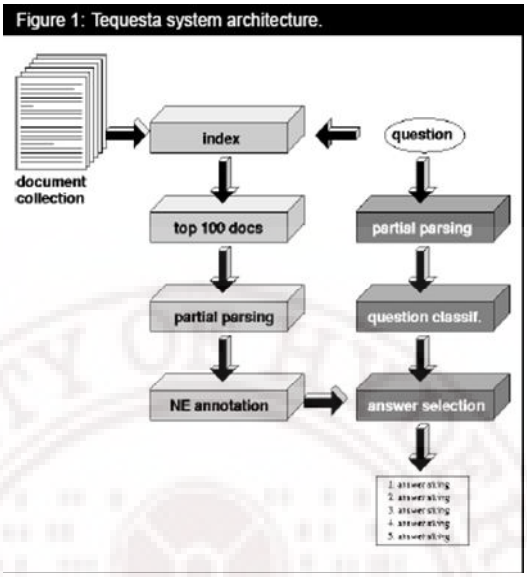


Figure 1: Chatbot application: Tequesta System [28]

Open domain surface based question answering system TREC 9

(Aaron, 2001)

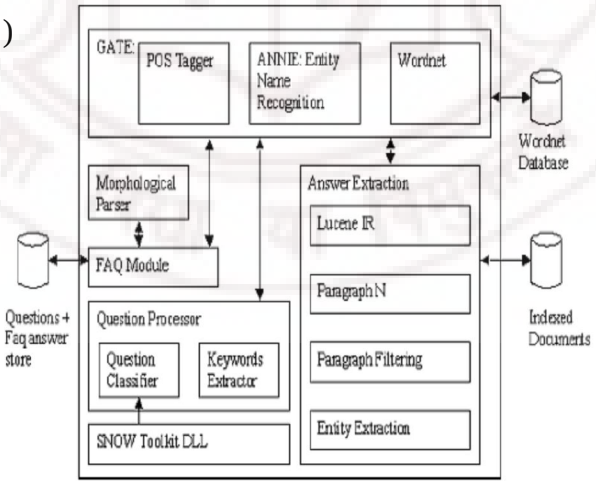


Figure 2: Chatbot application: Open domain surface based Question answering system [34]

Based on these applications, the automated customer support application is designed. The details of the design and working are presented in the next section.

4.2 Proposed design for automated customer support assistant and its detailed working.

The customer support assistant performs three important functions namely:

- Providing an efficient means of eliciting information from the customer.
- Assisting the customer service executives to ask better questions to the customer.
- Providing a higher level problem description to the maintenance team (development team)

The figure below shows the architecture diagram of the customer support assistant.

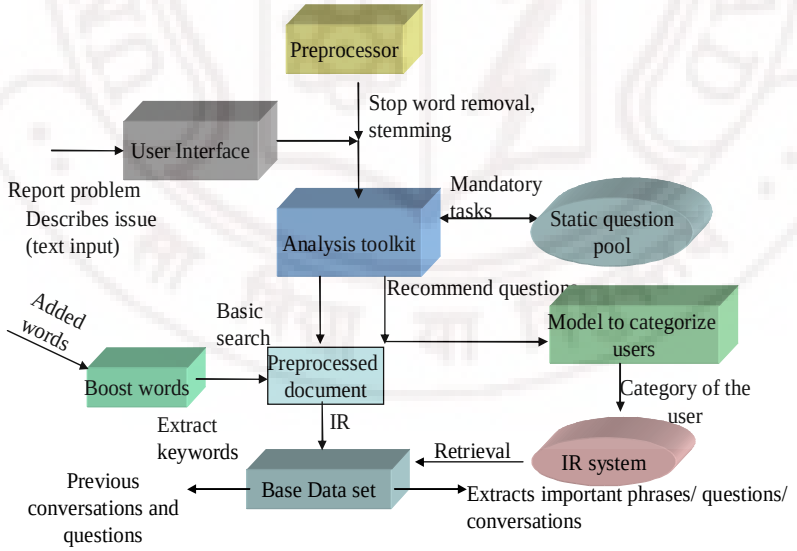


Figure 3: Proposed architecture for Customer support Assistant

The customer support service Assistant application basically works in two modes.

1. The static application mode
2. The question recommender system mode.

The functionality of the application in both the modes is common to some extent. In both modes, the user (Customer) input is taken in, preprocessed (Stop words removed) and sent to the analysis toolkit. The difference in functionality is seen at the analysis toolkit module.

In static mode, the toolkit provides interfaces that allow the customer service executive to query for static or mandatory questions that can be asked to this particular customer/ customer with a specific issue.

The output of this mode will be a possible set of mandatory questions that can be asked in order to the customer [28]. This assures of a standard questioning to the customers irrespective of who the support executive is.

It also ensures that the elicited information is in a structured format so that it is easily comestible by the maintenance team. (Development team)

The second mode of operation is the question recommender mode. The basic concept behind this mode of operation is to study the profile of the customer, categorize him / her and then suggest to the support executive the possible set of questions that can be asked to the customer.

The idea is to capture critical information about the problem which the customer might have missed out in his problem description. It is highly likely that the same customer or a different customer at some other time reported the same problem. The recommender mode captures such conversations and prompts the support executive to ask questions of that kind.

By categorizing the customers based on their pro-files (that include the type of issues they encounter and products they use), [27] [29] customers can be asked with questions that have helped to identify the problem accurately in the previous cases.

The output of this application is a set of possible keywords that are encountered in the previous conversations between the customer and the customer service executives. These keywords are given as input to the base data (Repository of all the enterprise data) and results are extracted [27]. These results are presented to the service executive.

The customer service executive analyses these results and asks the customer questions that have high probability of capturing critical information about the problem.

This application addresses the issues related with structured way of eliciting information from the customer and empowering the customer support executives with access to the knowledge repository.

Chapter 5

Defect management system auto fill application: Design proposal

This chapter presents in detail about the design and working of Defect management system auto fill application. Text mining techniques which are the basis for design of this application are presented in the initial sections. Later sections would present the detailed architecture (proposed design) and its working.

5.1 Defect management system: A survey

Defects management system is a "Defect Tracking System" which maintains a database of problem reports. Defects management system allows individuals or groups of developers and consultants working on same project to keep track of outstanding defects in their project effectively. [50]

Defects management system can track defects and design changes, communicate with teammates, submit and review patches, and manage quality assurance.

The defect management systems help the organizations to gain benefits that include: [41] [50]

- DMS improves communications between customers, support staff and developers.
- Increases the quality of the projects.
- Increases overall productivity.
- Facilitates easy bug tracking.
- Strives to eliminate redundancy in tasks.

Defect management is essential for all software development projects. Automated defect management tools have become commonplace. In addition to the specific tasks they perform, they are highly customizable and flexible.

These tools help the organizations to manage resources, track changes, plan and implement actions and generate reports for analysis.

The performance of these tools depends on the amount of data that is put in them and then these applications gain momentum and integrate with the development and maintenance cycles of the project. [50]

However, in many cases the data put is inaccurate or insufficient for various reasons. This finally results in not utilizing the full potential of these tools.

On the other hand, given an issue (during the maintenance cycle or development cycle), there is a lot of discussion between individuals in a team, across teams; between the support group and the development group, at various levels of hierarchy etc using various online and offline tools. (Online-messaging, e-mails, chats and other information sources like documents and tutorials.)

The application auto-fill will integrate the data/text pertaining to all such valuable discussions with the Defect management tools using text mining techniques. [25]

5.2 Text mining techniques for Record Auto fill

The recent abundance of digital information available electronically has made the organization of textual information into an important task. Text mining is a burgeoning new technology for discovering knowledge from text data. [35]

Text mining or text data mining, the process of finding useful or interesting patterns, models, directions, trends, or rules from unstructured text, is used to describe the application of data mining techniques to automated discovery of knowledge from text [36] [37]. Generally text mining has been viewed as a natural extension of data mining.

This reflects the fact that the advent of text mining relies on the burgeoning field of data mining to a great degree.

However, unlike data mining, which focuses on the well-structured collections that exist in either relational databases or data warehouses, text mining excavates data that is far less structured [37]. Much of today's electronic data resides not in traditional relational databases, but is hidden in the Web and natural-language documents [40].

In comparison with relational databases, natural-language corpora available on the internet are heterogeneous and noisy [26]. The same is the case with Enterprise data. Entries in many textual database fields could exhibit minor variations that can prevent mining algorithms from discovering important regularities. [38]

Variations can arise from typographical errors, misspellings, abbreviations, as well as from other sources [38]. Variations are particularly pronounced in data that is automatically extracted from unstructured or semi-structured documents or web pages.

For example, in data on local job offerings that we automatically extracted from newsgroup postings, Windows operating system is variously referred to as “Microsoft Windows", “MS Windows", “Windows 95/98/ME", etc.

Some previous work has addressed the problem of identifying similar or duplicate records, where it is referred to as record linkage, the merge/purge problem, hardening soft databases [38] [39].

Typically, a fixed textual similarity metric is used to determine whether two values or records are similar enough to be duplicates. In this approach, “Microsoft Windows", “MS Windows", and “Windows 95/98/ME" are mapped to a unique term and recognized as one entity. This is done as a pre-processing step [39] [41].

Similarity of text can be measured using standard bag of words metrics or edit-distance measures [25]. Other standard similarity metrics can be used for numerical and additional data types.

For instance, soft matching rules such as: “If Windows is in the list of required skills for a job, and then knowledge for IIS is also required for that job." are discovered from a set of job announcements. In this case, “Windows" and “IIS" can be matched to similar strings such as “MS Windows" or “IIS Services" respectively [41].
Based on the works mentioned above, this defect management auto fill application takes assistance of text/data mining techniques [42] in order to learn the patterns for each field in the defect management systems’ template. The section to follow will describe the working of this application in more detail.

Text mining techniques churn the unstructured text present in all the communication sessions and present a high level description (Summary, key-word set, related words etc) as the data bed for the tools mentioned above. [26]

This would feed data that is both quantitative and qualitative for the issue. This eliminates the problem of manual filling of records thus improving the performance of defect management tools. [26]

The hypothesis here is that the performance of defect management tools can be substantially improved by integrating data available in online and offline discussions, using text mining techniques.

Generally, the Defect management system templates (DMS templates) are specific to each system and are static.

Each field in the template (an attribute) is read from the template and its operator is chosen from the set of operators.

Here, operators are set of rules that decide the value to be filled for each field. These rules act as conditions for pattern match in the search space.

Text is pulled from online messaging systems, offline mailing systems and discussion forums, CRM's etc.

A Pre-processing model will remove the stop words and performs stemming (Reducing a word to its root form). Corpus is populated with these words and indexed.

5.3 Design proposal: Defect management system auto fill application

An IR (Information retrieval) component allows the user to view all related discussions with respect to a given defect. This corpus is treated as base data for text mining engine.

A Machine Learning toolkit [42] that has a text / word processing plug in with data mining/ Text mining algorithms are run on this corpus to obtain pat-terns specific for each field. Operators are applied on this search space to fill values.

The entire process is described as a three-stage working model in the figure below. In stage 1, partly filled tickets and the entire database of communications are considered as Index and base data for Information retrieval operation respectively.

Searching for unnecessary data is relatively a simpler task than discovering useful patterns. Hence, a model to do the same is built (to remove the unnecessary data from corpus).

This model can be a neural network, a classifier, set of business rules etc. For example, formal communications do have phrases like "Hello", "How are you", "Thank you", "and Bye" etc that are not necessary and can be eliminated.

Once this is done, a support file for each ticket is obtained. These are considered as communication nuggets for each defect. This process is done for each ticket. This result in support files for defects already handled and defects to be handled.

Now, the support files are sorted as support file specific to this (current defect reference) ticket, support files related to a specific customer and support files for all the tickets and customers.

This can be visualized as a specific to general search space. The user will be provided with a certainty adjustment knob that controls the search space.

If the user wants high accuracy while filling of records, his preference would be to search only in the support file specific to that defect.

The user can also go easy by allowing the search space to be specific to all support files for this customer. In rare occasions if only filling of values is necessary then all the support files can be searched.

Accuracy Vs number of fields filled will be determined by the certainty level provided by the user. These support files form the search space for pattern discovery. This is stage 1.

In stage 2, a training set of fully filled tickets are taken. These set of fully filled tickets give a flavor of the type of values filled for each field.

Data mining algorithms will be run to obtain a specific model for each field (Pre-processing is done before the algorithms are run). These models are in reference to the operators for each field mentioned in the previous paragraph. This concludes stage 2 of operation.

In stage 3 partly filled tickets (tickets for auto filling values), the model for each attribute (operator) and the certainty level are taken as inputs and a search is performed for the matching pattern. The matched patterns will be auto filled.

In some cases, there might be dependency on attributes which may not be obvious. In those cases, filling of an attribute value and then searching for another value yields good results. A feed back is provided to take advantage of this fact. This process is done for each attribute.

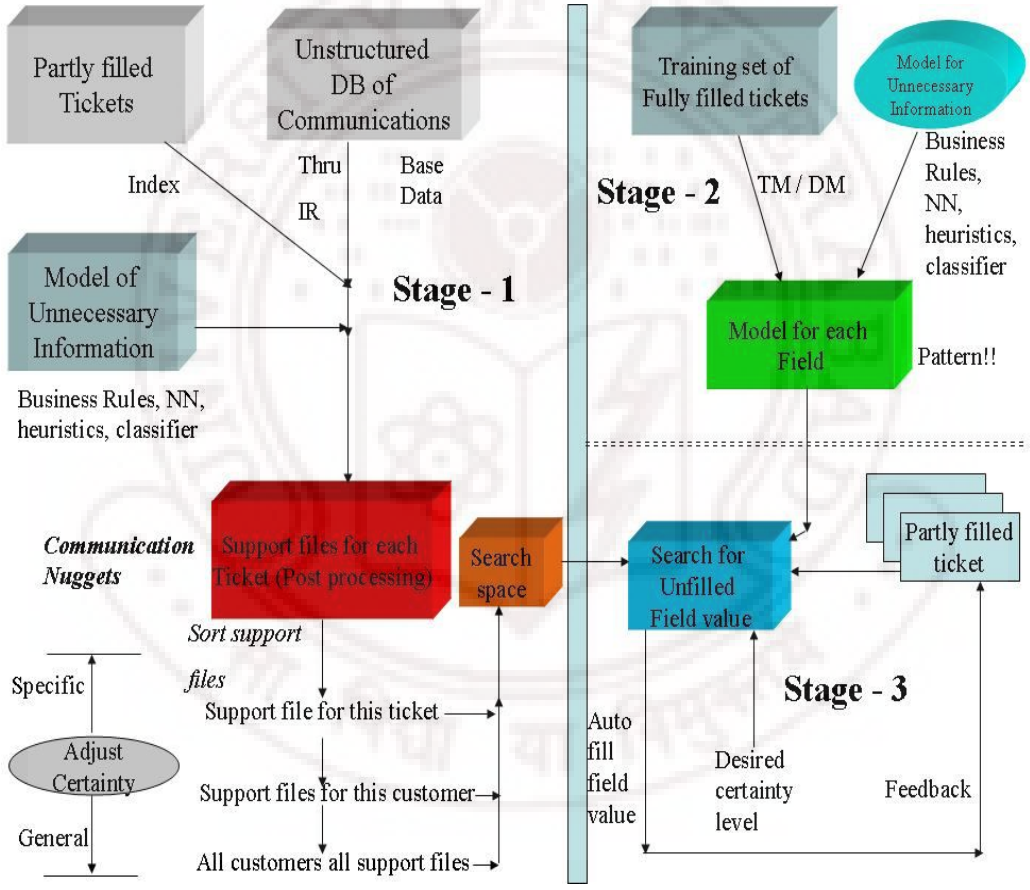


Figure 1: Three stage architecture diagram for record auto fill application

The adjustment parameter can be a classifier with cloud size as the number of nearest neighbors or a fuzzy system that would map the certainty with the support files to search for.

Stage 1 will run once for all the tickets. Stage 2 will run for each attribute of all the tickets and stage 3 will be for each attribute of an individual ticket.

This application would reduce the burden on the developers as it does not impose the developer to fill information related to the issue. The application does that itself.

However, the application expects that the user validates the entries made by the application so that a feedback mechanism is possible. This feedback allows the application to learn the expected input better and come up with more accurate results the next time.

The defect management system auto fill application interfaces with the information sources that are identified as pertinent to the defects pool and the defect management system.

For these reasons, the architecture of this application is made robust enough to be flexible and scalable. Ideally during the implementation phase, the auto fill application will be tailored in such a way that it can be configured to add information resources that are vital for training.

With these features it is hypothesized that the Defect management system auto fill application will be a handy tool.

Chapter 6

Data organization for Enterprise search

This chapter will give a detailed description about the Enterprise search engine design and the proposed data organization structure for the enterprise content. The chapter begins by exploring various information sources that are identified for data coalition to leverage knowledge inputs to the people involved in support process.

6.1 Identifying Information sources

Information sources in the organization are typically various tools used by the company to capture the interactions between various role players in the process.

Enterprise content in this scenario consists of the following information sources.

a) **Customer relationship management data**: The conversations between the customer support executives and developers is stored in this tool. These issues are typically regarding issues faced by the customers for a wide variety of products and releases. The content is mostly text. [44] [45]

b) **Problem management records**: A problem management record is generated when a customer requests a fix to be made to the software problem using a service request. This record gives provision for the customers to attach relevant files to the record so that the developer will be able to understand the problem better. A PMR number is assigned to track the request. A PMR is also referred to as a problem report. A PMR number is also referred to as a report number. [9] [53]

c) **Authorized program analysis report**: An APAR (acronym for authorized program analysis report) is a term used for description of a problem with a program that is formally tracked until a solution is provided. An APAR is created

or "opened" after a customer (or developer) discovers a problem that determines is due to a bug in its code. The APAR is given a unique number for tracking and a target date for solution. When the support group that maintains the code solves the problem, a temporary fix is given to the customer. This temporary fix if accepted by the customer will close the APAR. APAR content is text and numeric data along with links to PMR and CRM. [53]

d) **Defect management System records (DMS)**: A defect management system record consists of well defined fields that are used to describe a problem and its fix. A DMS record is used by the managerial level employees and the support executives to determine if the problem stated by the customer is redundant. DMS records are to be filled by the developers. The content of DMS consists of organizational text (non English terms that make specific sense to the organization) along with references to corresponding technical documents and user manuals. [53]

e) **Technical documents, User guides, Manuals, White papers**: These are documents that are enclosed with each and every product. They contain vital information about usage, problem handling, installation, configuration details that are most of the times the reasons for bugs to encroach. These documents would be helpful for the developers who fix the issues. The data is combination of text and numbers with text dominating the share. [53] [61]

6.2 Data Organization: A hierarchical approach, Design and Working

The organization of data in the knowledge base facilitates the working of these above mentioned applications [9].

A hierarchical data organization will be later presented that supports for easy search and structured query retrieval. [30]

The design of knowledge base is critical as mentioned in the above section. We adopted a hierarchical structure to all data pertaining to the organization [9].

The figure below shows the proposed structure for organizing data. Every organization deals with a product or a process. So at the root, the product / process will have its place.

Many times it is a common experience that recent documents are preferred over the older documents [9]. Added to that, older documents may not have a place in the knowledge base. So it is inferred that a temporal split would make sense. This will also make the life of maintenance team easy.

The temporal split of data here is based on the version number of the product or process. The next split in the hierarchy is based on categories [51]. A category can be defined as a specific group that determines the behavior of the application.

For example, an organization may be dealing with a product that deals with different databases. Then the categories will be these databases.

Each category is further classified as an issue. Here the issues are common problems that are encountered in a particular category. Under each issue are the existing databases that include DMS, the Resource Management tools and general documents.

The General documents include customer Frequently Asked Questions (FAQ), Developer FAQ, User guides, developer guides, developer reference manuals etc.

If relevant information is not found in this hierarchy, there is an option for the user to get connected to the web [48]. This is mostly useful for the developers who often refer to online reference manuals and code tutors.

The DMS documents will be integrated with the auto fill application. The details of auto fill application are presented in the next section.

An application that computes the similarity between issues across categories is put in place so that this will be helpful when an issue similar to a query request is found in category not specified by the user [9]. This can be a valuable piece of information for the developer.

Organizing data in this manner facilitates easy search and structured query retrieval. It would also be easy to provide regulated access to certain users and full access to others.

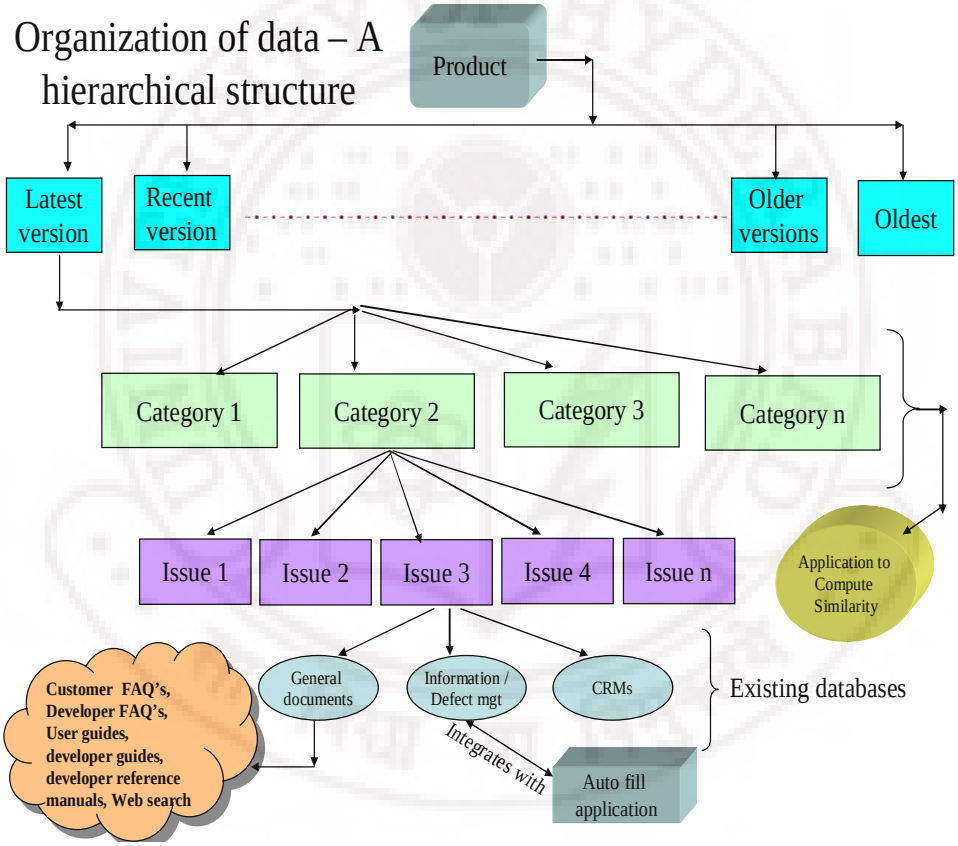


Figure 1: Hierarchical organization of data

6.3 Advanced Search and Query retrieval

All the aforementioned applications rely on how the enterprise content is organized and later integrated while searching. To be more specific, the core functionality of the entire knowledge management framework is to have an enterprise search application that

integrates the enterprise data and presents results to different role players as and when needed. [43]

Based on the hierarchical organization of data, we propose here the architecture of an enterprise search application that handles the task of searching documents from multiple resources. [30]

This application (at the initial level) will be a typical enterprise search engine. It has a query parser, an indexer, a result presentation (basic User Interface) scheme and connectivity to various databases that hold enterprise content (like the content management systems, the defect management systems, Customer resource management tools, other documents etc)

The query parser takes input from the user (any of the players in the support chain), parses it and extracts the keywords from the query (description given by the user).

The keywords are then passed to the Information retrieval system which basically is any enterprise search application. The index for this application is built based on the hierarchical structure [43] proposed in the chapter. The hierarchical structure forms the metadata for the document.

Hence, the enterprise search engine performs a search based on the metadata than actual data. This will improve the efficiency of results retrieved. [51] [52]

The data that will be indexed are various data-bases that contain enterprise content. The search engine indexes the content present based on the metadata specifications.

Once the query is presented, search is performed based on keyword to metadata map [51]. If a match is found, then documents that share the same metadata information (this happens at each level, that is product, version, category, issue etc) are only considered to rank and retrieve as relevant documents.

The results are presented in a segmented manner so that sorting and analysis of results becomes easier [52]. This is important as the needs of all roles in the organization are fairly different. For example: A development engineer may wish to view a defect management system record however a support executive may wish to view a content management / resource management record.

This application ensures that all other applications work in sync. Each and every application depends directly or indirectly on the search mechanism making it the most important ingredient in this knowledge management framework.

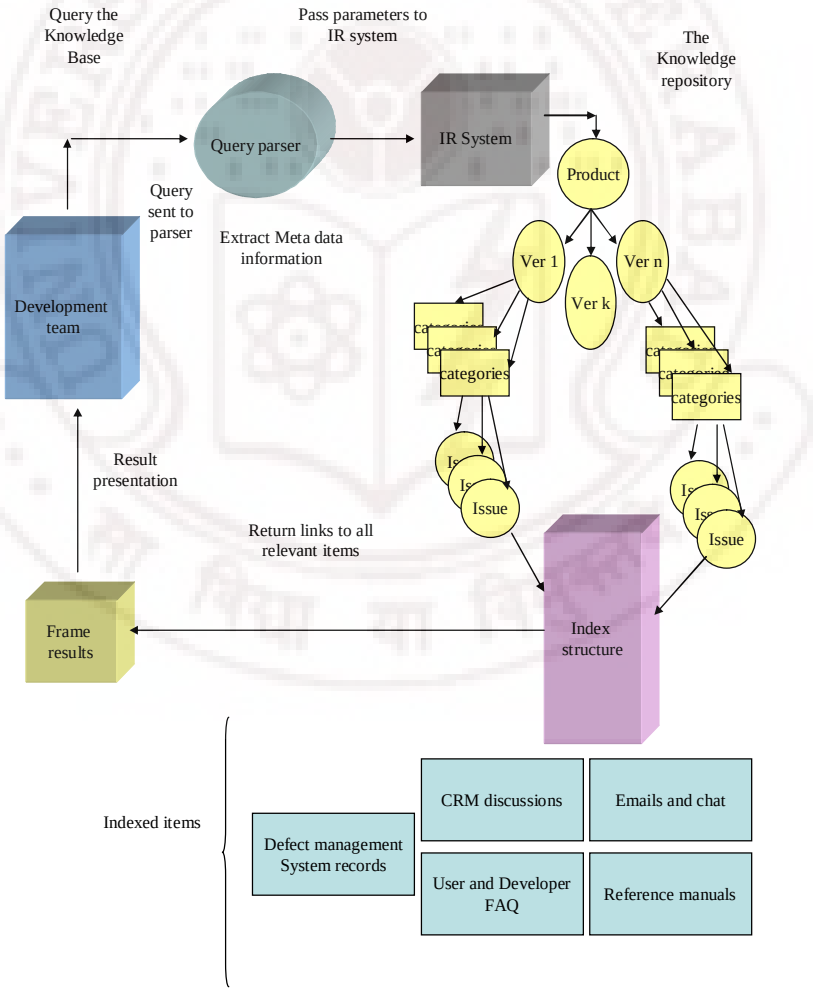


Figure 2: Proposed Enterprise search application

6.4 Future work

Developing these applications and integrating them at a very small scale to demonstrate the concept is the #1 priority task.

Integrating features to the search application to make it dynamic and interactive are being worked on. For example: Evolving this application as a recommender system based on user profiling rather than a simple search engine.

The idea is to provide different results to the users based on their user profiles. This motivation be-hind this idea is that the needs of various roles in an organization vary to a great extent. Hence, providing the right information to the right person at the right time will be the primary goal.

Bridge applications that help in reducing the bur-den/ dependence for data on employees in the organization are being designed.

It is proposed that new features be added to the search application that helps the users to search for related documents (similar pages), add key-words (boost words) to the documents and providing relevance feedback etc.

The aim of this project is to come with a knowledge management toolkit that helps people at the managerial level to manage their resources effectively irrespective of the domain and scale of the enterprise.

Chapter 7

Enterprise Search and Similarity of Records

This chapter will present in detail about the scope of proposed work that has been put to implementation. The chapter begins with an introduction to enterprise search engines followed by state of the art research proposals that specify the importance of similar pages in search technology. Later a detailed description of design proposal for implementing similar pages is produced along with experimental results.

7.1 Enterprise search: A brief overview

The designs proposed in the previous chapter have all heavily relied on a framework that integrates the basic information sources in an enterprise. Added to this a core functionality of this framework would be to provide a search interface for the users so that they can run queries across the information sources and retrieve best possible results.

Hence, the heart of this proposed knowledge management framework would be a search application that can get connected with the information sources of an organization. Once this feat is achieved, this application can then provide scope for integrating the bridge applications so as to evolve into an ideal knowledge management framework for the customer support process.

In order to achieve this, an off the shelf search engine is taken [58]. It is configured for recognizing the information sources so that it can later index this organizational content.

The later sections would give a brief listing of the information sources available in the organization.

The search engine chosen here has the following components namely: Crawler, Indexer, Analyzer and Search application. [57] [58]

Crawler component collects the documents from the database on a continuous basis. This is done in order to ensure that users have access to latest information.

Analysis of the documents is done by the parser component. The parser extracts the documents and performs linguistic analysis. This step is necessary for improving the quality of the retrieved results.

The Indexer runs on a scheduled basis and adds information about new document to the search index. The search application works in sync with the search module of the Enterprise search engine and retrieves results corresponding to a given query.

7.2 Featuring similar pages: Why Similar pages?

Results retrieved by the search engine are dependent on the selection criteria specified in the search module of the Enterprise search engine (usually relevance). It is observed that these retrieved results show a significant degree of variation. [58]

Experimentation on enterprise content for commonly used keywords shows that every 5 out of top 10 on an average, for retrieved results are different. This is a booming 50% variance in results. [57] [59]

Many a time, it is common for the end user to have interest to explore documents that are content wise similar to a selected result [57]. In order to provide the end user with this feature, a model is proposed to find out similar documents for a selected result. The sections to follow would briefly mention about the information sources available in the enterprise and the data being handled.

7.2.1 Data sources

An Enterprise search engine can crawl, index and search content from various information sources. Enterprise content in this scenario consists of the following information sources.

a) **Customer relationship management data**: The conversations between the customer support executives and developers is stored in this tool. These issues are typically regarding issues faced by the customers for a wide variety of products and releases. The content is mostly text. [53] [61]

b) **Problem management records**: A problem management record is generated when a customer requests a fix to be made to the software problem using a service request. This record gives provision for the customers to attach relevant files to the record so that the developer will be able to understand the problem better. A PMR number is assigned to track the request. A PMR is also referred to as a problem report. A PMR number is also referred to as a report number. [53]

c) **Authorized program analysis report**: An APAR (acronym for authorized program analysis report) is a term used for description of a problem with a program that is formally tracked until a solution is provided [53]. An APAR is created or "opened" after a customer (or developer) discovers a problem that determines is due to a bug in its code. The APAR is given a unique number for tracking and a target date for solution. When the support group that maintains the code solves the problem, a temporary fix is given to the customer. This temporary fix if accepted by the customer will close the APAR. APAR content is text and numeric data along with links to PMR and CRM. [53] [61]

d) **Defect management System records (DMS)**: A defect management system record consists of well defined fields that are used to describe a problem and its fix [53]. A DMS record is used by the managerial level employees and the support executives to determine if the problem stated by the customer is redundant. DMS records are to be filled by the developers. The content of DMS consists of organizational text (non English terms that make specific sense to the organization) along with references to corresponding technical documents and user manuals.

e) **Technical documents, User guides, Manuals, White papers**: These are documents that are enclosed with each and every product. They contain vital information about usage, problem handling, installation, configuration details that are most of the times the reasons for bugs to encroach [53] [61] [62]. These documents would be helpful for the developers who fix the issues. The data is combination of text and numbers with text dominating the share.

7.2.2 Approaches to find similar pages

Previous attempts to implement similar pages in search applications are related to web content. The reason for popularity among developers to choose web content is that the graphical structure or the link based structure of the web can be exploited to find similar pages.

Similarity in this context is defined as the documents that share the same link structure. For Instance: www.yahoo.com/labs and www.yahoo.com/jobs are considered to be similar using the basic link level analysis of the web [59].

Another approach for finding similar documents on the web is based on the number of links present in each document [59] [60]. Upon parsing the selected document, either its source is parsed to check for links that it contains or the base document is parsed to check for links it contains. Upon finding the links in either of the documents, these links are posted as similar pages to the selected document.

As mentioned above similarity of two documents on the web is usually done by exploiting the link structure of the web. However, in general sense similarity between two documents means that content wise they are same or speak about the same concept.

Traditional approaches to achieve content wise similarity include analysis of the document based on term level and sentence level. This process includes complete parsing of the document and understanding its behavior [60].

A sentence level analysis of the document involves a linear parsing to determine its part of speech added to a hierarchical parsing that gives information based on the level at which each component (a term or a phrase or a clause) are related to each other [56].

Term level analysis of the document measures the frequency of the term in the document and based on that information assigns weights to the terms [54]. Words that are most important are weighed high in contrast to words that do not contribute to the analysis of document.

In this approach term based analysis is used to compute document similarity. The sections to follow would in detail describe the process of computing similarity, reasons for choosing the term based analysis, proposed architecture, implementation details and obtained results.

7.3 Our approach

As mentioned above, “Similar pages” for Enterprise search is a feature to find out documents that are related to a particular result. The purpose of this concept is to provide results (documents) that are content wise similar and about which the user may not be familiar.

Many a times it is a common phenomena that the user is interested in a particular result and aims to research documents similar to that result, at that instance the similar page feature retrieves results similar to the result without bothering the user for extracting keywords and rerunning the search.

In this approach, terms in the documents are weighted based on the Term Frequency (TF) Inverse Document Frequency (IDF) metric with filters for eliminating unnecessary and uninformative words applied. Based on the term weights, the top five words in each document are computed.

A search query is built based on the existing metadata information of the document and this "Representative keyword set" specific to the document and the same is presented to the Enterprise search application.

In order to enhance performance, the computations are done on the fly. As the process of computing term weights for all the documents is not feasible we have employed a technique that computes term weights based on the top "k" (k is usually top 25 documents in the result set) results for the query given by the user.

7.3.1 Previous approaches favoring term based weighting of documents

Prior studies in Information retrieval and text categorization reveal that improved results are seen when terms in the document are weighted.

Yang [59] describes a few variants of TF-IDF scheme that he has used for term weighting. Ac-cording to his analysis, TF-IDF approach to weight documents is simple yet significant means of representing documents and queries.

Greiff [60] had experimented to find out relation-ship between occurrences of a query term in a document and its relevance to the information need. In his exploratory analysis he had focused on the efficacy of Inverse document frequency along with Term frequency in retrieving relevant documents.

His findings throw light on the fact that the correlation among query terms and document collections is robust.

These aforementioned studies are based on the works of Gerard Salton [54] whose works on term weighting approaches for automatic text retrieval clearly describe the importance of term weighting.

These studies are mainly focused on information retrieval when a search query is presented to the search engine.

Tasturoni [56] proposes a new term weighting scheme that is different from the query based summarization methods that is, rather than using the information in the query, they utilize the similarity information among documents by hierarchical clustering. Document similarity is based a term weighting metric named Information gain.

The idea for implementing similar pages in the Enterprise search application is rooted to these studies. Analogous to weight query terms and documents based on the TF-IDF scheme, we chose to represent all the documents as a set of representative words.

Once this is achieved, weighted terms can be sorted to find out which words best describe the document.

These words act as search query for the search engine. We hypothesize that results retrieved by the search engine when queried with these terms are pages (documents) similar to the original document.

7.4 Detailed description of objectives and Proposed Architecture for Similar page Implementation

The main objectives of our work can be described in terms of four major steps:

- Designing a model to integrate the functionality of similar pages in the existing Enterprise search application.

- Selecting an appropriate document representation scheme and thereby a term weighting strategy for the cause.
- Implementing this model to obtain a weight matrix for each document in the document base.
- Integrating this application with the existing search application.

As a first step to feature similar pages into the Enterprise search application, we designed a model for the same. This initial model represents each document in the document pool as a "bag of words" and computes a weight matrix for every document [55].

The idea of implementing similar pages can be summarized in the following manner: The Enterprise search engine retrieves documents that match the user's query (and preferences).

The user chooses a result that interests him. It is for this document that similar pages will be computed. Each document in the document pool will be associated with a weight matrix. Upon selecting a document, its corresponding weight matrix is retrieved.

The weight matrix of each document is computed before hand based on a term weighting scheme chosen to best fit the purpose. (TF – IDF in our case). The weight matrix consists of (term, weight) pair sorted based on the weight of the term.

Once the weight matrix is retrieved, top five words from the matrix are chosen and a search query is formed with them. This search query is presented to the Enterprise search engine and the results retrieved are presented as similar pages to the selected document.

This process is based on the initial design of the application. A few changes were made in order to enhance the performance. These changes will be described in detail in the later sections.

Added to this default behavior, the user is presented with top ten words in the weight matrix so that he/she can choose the words to form the search query.

The sections to follow would describe in detail about the representation of documents, term weighting, formulating the threshold function and a few implementation details.

Below we present the model that achieves the expected behavior.

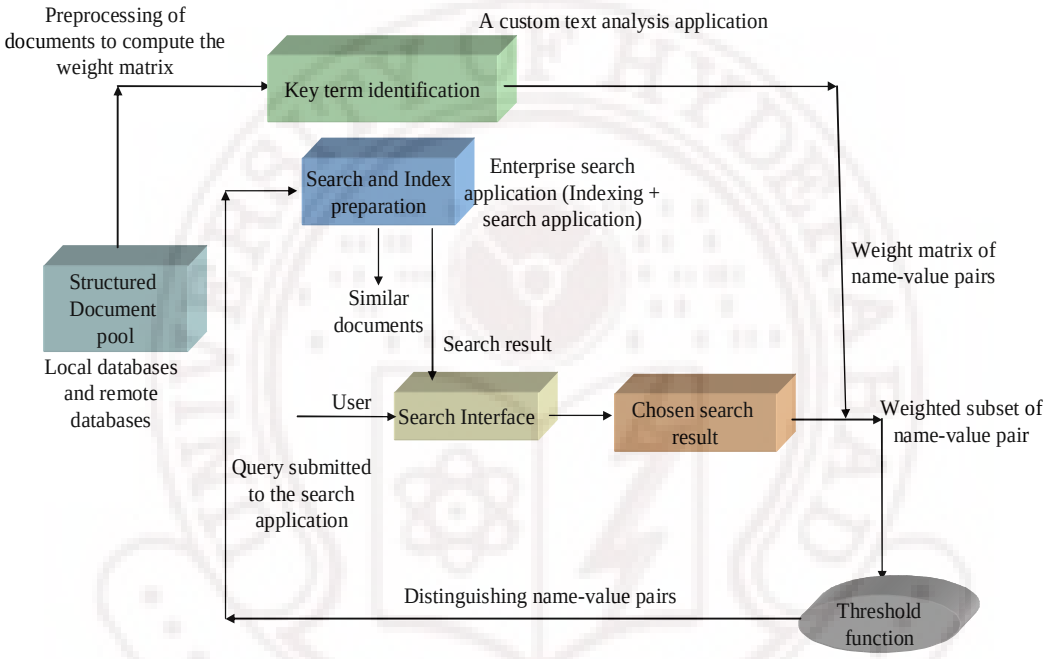


Figure 1: Initial model to implement similar pages

In the model presented above, the documents from the structured document pool (Information sources of the Enterprise) are processed in order to compute the weight matrix for each document.

This is achieved through the custom text analysis application that does pre-processing (eliminating stop words) and computes weights based on the weighting rule. (TF-IDF)

The weight matrix is pre-computed and every document is associated with a weight vector. The search engine module corresponds to the Enterprise search engine that retrieves documents based on user's preferences.

When a similar page request is generated, the weight vector corresponding to the document selected is retrieved. Based on the top five words in the weight vector a new search query is generated that is again sent back to the Enterprise search engine to compute similar pages.

7.4.1 Document representation and Term weighting

In order to obtain documents similar to a particular document, it is essential that a metric be defined that computes similarity between any two documents. [55]

For the task at hand, the document representation for computing term weights was chosen to be a "bag of words" model and the term weighting strategy as the product of Term Frequency (TF) and Inverse Document Frequency (IDF).

Term frequency is defined as the frequency of a term in the document. Inverse Document frequency is defined as the occurrence of the term across documents in the collection.

More specifically IDF is computed as [54],

$$idf(t) = 1 + \log\left(\frac{numDocs}{docFreq+1}\right)$$

Where docFreq denotes the number of documents in the collection that have the presence of term t and numDocs represents the number of documents in the collection.

There are a couple of reasons that motivated us to use this weighting scheme for computing important words in the document.

1. The size of documents is on the lower side than usual web page documents and other text data sources. Upon performing stop word removal the size of data present in the document is further reduced and that leaves us with a set of non English words and a few Enterprise specific terms and only very few literary English terms. Hence, a sentence wise analysis of the document is not possible.

2. The documents also consist of a significant number of numeric terms that can-not be ignored. In order to compute their importance, a term level analysis was inevitable.

A snapshot of available document sources is given below to give the reader an understanding of the kind of data being dealt with.

```
“I have asked customer to send me the "ls -la" output of
the files in the
directory $HOME/sqllib/adm.
-------------
 ACTION PLAN
-------------
Waiting for customer feedback.
+XYZABC, ELLIS WY      -1000999 -L50J/-------P1S1-08/09/28-
06:06 -AT
--------------
 ACTION TAKEN
--------------
Received the output from customer,
total 8976
drwxrwxrwx   2 1000 Oct 19 2003
drwxrwxrwx  19 1999 Dec 11 2003”
```

Figure 2: Snapshot of the data being handled

Each term in a document gets its weight as the product of its term frequency and its inverse document frequency.

7.4.2 Changes to initial design

The initial design viewed term weighting as a preprocessing step. Its principle was to compute the term weights for each and every document in the document repository and store them as a matrix of (document → Weight Vector).

However, in the later phase it was found out that the cost of computing weight vectors for every document was a performance issue. Hence, a dynamic computation of term weights was chosen.

In this model, the term weights will be computed once the Enterprise search engine retrieves the results for a user query.

Top twenty five results of this search (results that are considered to be most relevant based on literature survey) are chosen as the document base for computing the TF-IDF values.

Once the size of the document base is restricted, the computations become faster and term weights will be dynamically generated. The remaining process is intact. The important words in the selected document will be computed and they will be used to form the search query for the Enterprise search engine to compute similar pages.

A threshold function is used to determine terms that deservedly represent the document. A local threshold at the document level and a global threshold at the Intra document level are set.

Local threshold value is the average TF-IDF in the document plus an error function (acceptable error rate) and the global threshold is the average of maximum TF-IDF across the documents.

As the weights are normalized, all the weights are between 0 and 1. Hence a quantitative comparison can be made between values across the documents.

Terms in a document that have their weights above the local threshold and that have difference with global threshold minimum are chosen to be the representative set of the document.

Top ten words based on these values comprise the set of important words of the document. These words will be presented to the user so that he/ she can select the words to form a search query or the default behavior of top five words will be used to form the search query and send it to the search engine.

7.5 Implementation details

The application mainly consists of three modules.

1. The search application.
2. The term weight calculator
3. Query generator and Results

The search application module gets connected to the search engine and retrieves results based on the user query. The output of this document is a set of results ranked by relevance that correspond to the user query.

These results are then sent for further analysis. The obtained results are in XML format. Hence, they are parsed before their content can be read and sent to the term weight calculator module.

Top 20-25 documents are considered for analysis as literature study reveals that there exists high probability for finding best possible match for the query in this range. [59]

These documents are represented as bag of words. Bag of words basically is a representation scheme where each document is preprocessed to remove unnecessary and uninformative content and later the document is nothing but a sequence of words.

A list of stop words is stored and they are removed if they occur in the document. Delimiter is defined to be any symbol that is not a character or number. Hence, all the content numeric, non numeric and composition of both of them (which is common for data of this kind) are considered as terms.

Computing weights for each of this sequence of words is done using the hash map concept in java. Hash maps store the term – frequency pair of each document in the selected list of documents.

A few rules are used for numeric content present in the document. If the numeric content is less than four characters in length then it is not considered for analysis.

The reason behind this is to eliminate the possible inclusion of date and other numerals often found in the documents that do not contribute as weighted terms. Once the term frequency pairs for all the terms across the documents is computed, the algorithm considers the term frequency of each term in the document and then computes the inverse document frequency of the term using the method described above.

The product of these two quantities is generated as the term weight. The process is repeated for each term in every chosen document. The term weight calculator module outputs values from this set which qualify the threshold function.

The threshold function ensures that terms that qualify locally (greater than average TF-IDF values) and compete globally (values closer to average of maximum values across documents) are only considered for forming the representative set of keywords.

Top five words from this set is taken and tagged as the representative set of the document. A sample test was run on five documents and the following results were obtained.

Example Output:

```
Number of documents processed=5

Document 1:
Number of input words: 599
Number of unique words: 141

TF-IDF          Term
1.0             sqllib
0.932           db2
0.917           adm
0.714           dwapinst
0.588           olppinst

Document 2:
Number of input words: 1289
Number of unique words: 340

TF-IDF          Term
1.0             db2
0.705           19547
0.687           17334
0.687           dwapinst
0.669           13345

Document 3:
Number of input words: 133
Number of unique words: 57

TF-IDF          Term
1.0             customer
0.894           db2
0.874           080304
0.656           chiang
0.656           ellis

Document 4:
Number of input words: 137
Number of unique words: 36

TF-IDF          Term
1.0             olppinst
0.874           sqllib
0.809           db2
0.76            adm
0.68            dwapinst

Document 5:
Number of input words: 196
Number of unique words: 92

TF-IDF          Term
1.0             x82
0.686           p1s1
0.6             082334
0.6             UDB
0.4             fixpacks
```

Figure 3: Sample Output on a test case run with five documents for an Input query with text value as db2

Each of these results is later concatenated to form query strings in the query generator module. Once the user selects a particular document to find pages similar to that, this query string is sent to the search engine and retrieved results are shown as similar page results.

While considering term a few heuristics were considered. Content that is composition of numerals and characters is weighted more than only numerals. Prior analyses of the documents revealed that, majority of terms that distinguish enterprise content are these terms.

All weights are dynamically generated and they vary with each query presented.

Once these keywords are computed during the term weight generator module, the user will be able to view these keywords. This option of presenting the user with keywords adds flexibility to the application as it makes the user an arbiter to choose keywords that should be present in the query than proceeding with default behavior.

7.6 Experiments and Results

The implemented algorithm was tested to determine its efficiency. The experimentation process included picking the top ten most frequent search terms in the organization (IBM in this case) and finding out the query strings that are generated based on the representative keyword set.

Top ten results of these top ten keywords are used to perform this experiment. The following are the list of keywords that are mostly frequently used based on the number of hits from the Asia Pacific region of IBM.

Db2, IBM, cmvcdb2, Infosphere, warehouse, AIX, memory leak, Linux, server, Tera-data and Informix.

The precision and recall values for the query strings based on the top ten results for each of these search term are given below:

Serial Number	Keyword	Precision For similar page result	Recall For similar page Result
1	Db2	0.89	0.94
2	IBM	0.675	0.75
3	cmvcdb2	0.68	0.82
4	Infosphere	0.825	0.92
5	Warehouse	0.83	0.87
6	AIX	0.72	0.79
7	Memory leak	0.63	0.68
8	Linux	0.69	0.72
9	Server	0.78	0.82
10	Informix	0.81	0.91

Precision of a given result is computed as the ratio of total number of relevant documents retrieved by the search system to the total number of documents retrieved by the system for a given search query.

Recall is computed as the ratio of total number of relevant documents retrieved by the search system to the total number of relevant documents that are to be retrieved by the system.

The corresponding search query strings for each of the above top ten keywords are given below:

Top ten results for most frequently used keywords at IBM along with corresponding search query strings for similar pages.

1. AIX

TF-IDF	Term
1.0	web
0.799	aix
0.714	administration
0.514	server
0.429	tested

TF-IDF	Term
1.0	aix
0.762	gdlc
0.667	ethernet
0.571	dlc
0.571	etherchannel

TF-IDF	Term
1.0	aix
0.991	wpar
0.582	client
0.571	system
0.542	api

TF-IDF	Term
1.0	aix
0.488	l950
0.488	toya
0.488	yasuhisa
0.439	tl5

TF-IDF	Term
1.0	web
0.799	aix
0.714	administration
0.514	server
0.429	tested

TF-IDF	Term
1.0	aix
0.867	hacmp
0.8	l50j
0.733	5765e5100
0.733	chuck

TF-IDF	Term
1.0	communications
0.939	server
0.865	aix
0.439	version
0.384	bit

TF-IDF	Term
1.0	aix
0.381	version
0.314	support
0.3	based
0.3	netbackup

TF-IDF	Term
1.0	aix
0.554	edition
0.545	manager
0.545	rscsrs
0.545	tivoli

TF-IDF	Term
1.0	aix
0.448	lang
0.448	loc
0.448	utf
0.379	1207

Search Query strings for similar pages:

- web aix administration server tested
- aix gdlc ethernet dlc etherchannel
- aix wpar client system api
- aix l950 toya yasuhisa tl5
- web aix administration server tested
- aix hacmp l50j 5765e5100 chuck
- communications server aix version bit
- aix version support based netbackup
- aix edition manager rscsrs tivoli
- aix lang loc utf 1207

2. CMVCDB2

TF-IDF	Term
1.0	bind
0.426	db2
0.417	dbconvert5
0.417	sql0020w
0.333	option

TF-IDF	Term
1.0	database
0.458	db2
0.383	migrate
0.256	logfilsiz
0.256	parameter

TF-IDF	Term
1.0	cmvc
0.586	serviceability
0.456	tools
0.443	cfg
0.391	utilities

TF-IDF	Term
1.0	security
0.75	contentdocs
0.75	corporate
0.75	nsf
0.75	sas

TF-IDF	Term
1.0	rollforward
0.903	database
0.756	family
0.75	forward
0.75	pending

TF-IDF	Term
1.0	5648a3400
0.857	p1s1
0.571	gfidb2
0.571	tables
0.429	l25r

TF-IDF	Term
1.0	pierre
0.9	hildebrand
0.846	l991
0.831	576520700
0.6	p3s3

TF-IDF	Term
1.0	p2s2
0.863	576520700
0.709	cmvc
0.611	alain
0.611	l993

TF-IDF	Term
1.0	576545400
0.826	p2s2
0.433	l165
0.372	l13m
0.326	charles

Search Query strings for similar pages:

- bind db2 dbconvert5 sql0020w option
- database db2 migrate logfilsiz parameter
- cmvc serviceability tools cfg utilities
- security contentdocs corporate nsf sas
- rollforward database family forward pending
- 5648a3400 p1s1 gfidb2 tables l25r
- pierre hildebrand l991 576520700 p3s3
- p2s2 576520700 cmvc alain l993
- 576545400 p2s2 l165 l13m charles

3. DB2

TF-IDF	Term
1.0	db2
0.787	rwxrwxrwx
0.325	root
0.319	chiang
0.319	ellis

TF-IDF	Term
1.0	adm
0.978	sqllib
0.731	db2
0.66	dwapinst
0.66	olppinst

TF-IDF	Term
1.0	db2
0.543	windows
0.53	bit
0.44	fixpaks
0.395	linux

TF-IDF	Term
1.0	version
0.876	db2
0.499	edition
0.308	client
0.278	udb

TF-IDF	Term
1.0	mvs
0.8	5625db2
0.65	db2
0.35	morgan
0.34	l50j

TF-IDF	Term
1.0	db2
0.791	5765f4104
0.744	braun
0.698	ulrich
0.651	l25c

TF-IDF	Term
1.0	db2
0.778	entry
0.556	db2fmcd
0.556	etx
0.556	inittab

TF-IDF	Term
1.0	db2
0.564	windows
0.55	bit
0.457	fixpaks
0.41	linux

TF-IDF	Term
1.0	db2
0.414	host
0.414	kenichiroh
0.414	l950
0.414	motono

Search Query strings for similar pages:

- db2 rwxrwxrwx root chiang ellis
- adm sqllib db2 dwapinst olppinst
- db2 windows bit fixpaks linux
- version db2 edition client udb
- mvs 5625db2 db2 morgan l50j
- db2 5765f4104 braun ulrich l25c
- db2 entry db2fmcd etx inittab
- db2 windows bit fixpaks linux
- db2 host kenichiroh l950 motono

4. IBM

TF-IDF	Term
1.0	ibm
0.069	024
0.069	installp
0.067	level
0.059	firmware

TF-IDF	Term
1.0	ibm
0.216	5765g0300
0.181	p3s3
0.165	l19l
0.158	anderson

TF-IDF	Term
1.0	ibm
0.363	p2s2
0.349	5765g0300
0.334	l19l
0.174	console

TF-IDF	Term
1.0	ibm
0.909	ibmus
0.636	stg
0.545	global
0.415	services

TF-IDF	Term
1.0	mqsi
0.713	messagebroker
0.412	ibm
0.162	lib
0.157	jar

TF-IDF	Term
1.0	ibm
0.333	phase
0.259	dev
0.238	device
0.209	0514

TF-IDF	Term
1.0	ibm
0.289	8203rout
0.281	aixamf
0.232	defined
0.224	available

TF-IDF	Term
1.0	identity
0.926	ibm
0.903	tivoli
0.611	manager
0.566	www

TF-IDF	Term
1.0	director
0.854	ibm
0.333	japan
0.292	sugawara
0.292	tadashi

TF-IDF	Term
1.0	ibm
0.762	business
0.653	www
0.644	tivoli
0.615	com

Search Query strings for similar pages:

- ibm 024 installp level firmware
- ibm 5765g0300 p3s3 l19l anderson
- ibm p2s2 5765g0300 l19l console
- ibm ibmus stg global services
- mqsi messagebroker ibm lib jar
- ibm phase dev device 0514
- ibm 8203rout aixamf defined available
- identity ibm tivoli manager www
- director ibm japan sugawara tadashi
- ibm business www tivoli com

5. Informix

TF-IDF	Term
1.0	informix
0.417	rwxr
0.218	opt
0.205	usr
0.175	bin

TF-IDF	Term
1.0	informix
0.86	captura
0.756	prodec
0.453	negocios
0.323	pdecbb

TF-IDF	Term
1.0	informix
0.344	rwxr
0.191	1048575
0.184	rawszcyc
0.169	dev

TF-IDF	Term
1.0	informix
0.339	sanchez
0.305	datablade
0.271	5724c5500
0.213	data

TF-IDF	Term
1.0	ect1lnk
0.85	home
0.653	informix
0.339	5180000
0.177	0x40001

TF-IDF	Term
1.0	informix
0.398	rwxr
0.183	5724c6601
0.154	antola
0.154	l24h

TF-IDF	Term
1.0	mix
0.651	bin
0.651	oninit
0.543	informix
0.149	makefll

TF-IDF	Term
1.0	informix
0.485	bin
0.444	rwxr
0.2	root
0.144	rwsr

TF-IDF	Term
1.0	dbspaces
0.937	informix
0.494	ist
0.453	2560000
0.245	dbspace

TF-IDF	Term
1.0	zhjs
0.768	list
0.581	informix
0.551	dbs
0.354	chk01

Search Query strings for similar pages:

- informix rwxr opt usr bin
- informix captura prodec negocios pdecbb
- informix rwxr 1048575 rawszcyc dev
- informix sanchez datablade 5724c5500 data
- ect1lnk home informix 5180000 0x40001
- informix rwxr 5724c6601 antola l24h
- mix bin oninit informix makefll
- informix bin rwxr root rwsr
- dbspaces informix ist 2560000 dbspace
- zhjs list informix dbs chk01

6. Infosphere

TF-IDF	Term
1.0	cdc
0.76	java
0.52	infosphere
0.44	jvm
0.36	engine

TF-IDF	Term
1.0	desktop
1.0	recommended
0.85	heap
0.85	settings
0.75	ibm

TF-IDF	Term
1.0	infosphere
1.0	patch
0.815	metadata
0.77	workbench
0.667	cq155910

TF-IDF	Term
1.0	ibm
0.611	websphere
0.5	infosphere
0.419	information
0.411	module

TF-IDF	Term
1.0	multilingual
0.858	infosphere
0.858	server
0.648	refresh
0.628	information

TF-IDF	Term
1.0	ptf
0.991	mdm
0.75	infosphere
0.75	server
0.674	workbench

TF-IDF	Term
1.0	ibm
0.783	class
0.743	infosphere
0.74	tab
0.74	v14

TF-IDF	Term
1.0	ibm
0.806	class
0.765	infosphere
0.762	tab
0.762	v14

TF-IDF	Term
1.0	ibm
0.741	class
0.7	tab
0.7	v14
0.676	infosphere

TF-IDF	Term
1.0	traceability
0.591	blank
0.545	pdf
0.523	server
0.502	target

Search Query strings for similar pages:

- cdc java infosphere jvm engine
- desktop recommended heap settings ibm
- infosphere patch metadata workbench cq155910
- ibm websphere infosphere information module
- multilingual infosphere server refresh information
- ptf mdm infosphere server workbench
- ibm class infosphere tab v14
- ibm class infosphere tab v14
- ibm class tab v14 infosphere
- traceability blank pdf server target

7. Linux

TF-IDF	Term
1.0	kernel
0.667	environment
0.566	install
0.5	assume
0.5	dir

TF-IDF	Term
1.0	icon
0.75	collection
0.75	mdist2
0.75	object
0.5	administrator

TF-IDF	Term
1.0	bonding
0.714	interface
0.549	linux
0.286	network
0.214	802

TF-IDF	Term
1.0	ste
0.75	tfst
0.589	avi
0.5	access
0.5	ftp

TF-IDF	Term
1.0	presentation
1.0	ste
1.0	tfst
0.785	mp3
0.667	ftp

TF-IDF	Term
1.0	code
0.375	client
0.294	dsmtca
0.294	permission
0.25	install

TF-IDF	Term
1.0	endpoint
0.785	suse
0.667	script
0.598	etx
0.598	start

TF-IDF	Term
1.0	font
0.571	properties
0.303	java
0.243	client
0.235	file

TF-IDF	Term
1.0	tec
0.875	size
0.754	file
0.637	gateway
0.5	tmp

TF-IDF	Term
1.0	winstlcf
0.425	found
0.381	bin
0.35	command
0.333	node

Search Query strings for similar pages:

- kernel environment install assume dir
- icon collection mdist2 object administrator
- bonding interface linux network 802
- ste tfst avi access ftp
- presentation ste tfst mp3 ftp
- code client dsmtca permission install
- endpoint suse script etx start
- font properties java client file
- tec size file gateway tmp
- winstlcf found bin command node

8. Memory Leak

TF-IDF	Term
1.0	informix
0.349	rwxr
0.145	usr
0.142	opt
0.135	bin

TF-IDF	Term
1.0	informix
0.661	captura
0.581	prodec
0.348	negocios
0.249	pdecbb

TF-IDF	Term
1.0	informix
0.288	rwxr
0.147	1048575
0.141	rawszcyc
0.13	dev

TF-IDF	Term
1.0	180
1.0	docview
1.0	uid
1.0	wss
0.939	http

TF-IDF	Term
1.0	ect1lnk
0.85	informix
0.762	home
0.339	5180000
0.177	0x40001

TF-IDF	Term
1.0	5724q36ds
0.867	l183
0.762	p2s2
0.533	grant
0.533	hooks

TF-IDF	Term
1.0	p4s4
0.897	5765f4100
0.61	l18z
0.563	paul
0.535	haus

TF-IDF	Term
1.0	informix
0.373	bin
0.372	rwxr
0.143	root
0.121	rwsr

TF-IDF	Term
1.0	5765f4103
0.69	p2s2
0.365	eveline
0.341	grosse
0.341	l29c

TF-IDF	Term
1.0	5765g0300
0.882	p2s2
0.514	l165
0.481	db2
0.395	perichiappan

Search Query strings for similar pages:

- informix rwxr usr opt bin
- informix captura prodec negocios pdecbb
- informix rwxr 1048575 rawszcyc dev
- 180 docview uid wss http
- ect1lnk informix home 5180000 0x40001
- 5724q36ds l183 p2s2 grant hooks
- p4s4 5765f4100 l18z paul haus
- informix bin rwxr root rwsr
- 5765f4103 p2s2 eveline grosse l29c
- 5765g0300 p2s2 l165 db2 perichiappan

9. Server

TF-IDF	Term
1.0	server
0.841	tsm
0.504	set
0.298	command
0.2	password

TF-IDF	Term
1.0	export
0.936	server
0.514	import
0.514	para
0.343	5698ism00

TF-IDF	Term
1.0	server
0.555	name
0.413	found
0.369	error
0.369	record

TF-IDF	Term
1.0	server
0.923	application
0.676	type
0.489	assign
0.489	base

TF-IDF	Term
1.0	java
0.727	jet
0.727	qoiv
0.698	server
0.364	init

TF-IDF	Term
1.0	server
0.751	domino
0.367	memory
0.367	restart
0.328	lotus

TF-IDF	Term
1.0	server
0.765	domino
0.332	5724e6200
0.281	p3s3
0.279	notes

TF-IDF	Term
1.0	server
0.926	topology
0.617	nmc
0.514	services
0.498	service

TF-IDF	Term
1.0	server
0.717	domino
0.479	5724e7000
0.479	p2s2
0.472	customer

Search Query strings for similar pages:

- server tsm set command password
- export server import para 5698ism00
- server name found error record
- server application type assign base
- java jet qoiv server init
- server domino memory restart lotus
- server domino 5724e6200 p3s3 notes
- server topology nmc services service
- server domino 5724e7000 p2s2 customer

10. Warehouse

TF-IDF	Term
1.0	infosphere
0.981	warehouse
0.896	environment
0.785	based
0.785	details

TF-IDF	Term
1.0	balanced
0.657	warehouse
0.438	notifications
0.382	software
0.372	infosphere

TF-IDF	Term
1.0	target
0.762	name
0.667	warehouse
0.556	define
0.444	olap

TF-IDF	Term
1.0	agent
0.857	proxy
0.743	database
0.714	warehouse
0.607	user

TF-IDF	Term
1.0	control
0.995	database
0.746	warehouse
0.589	unicode
0.459	version

TF-IDF	Term
1.0	tivoli
0.978	data
0.6	warehouse
0.305	application
0.28	information

TF-IDF	Term
1.0	utility
0.991	control
0.958	warehouse
0.813	database
0.75	manager

TF-IDF	Term
1.0	brick
1.0	red
0.533	gc18
0.483	warehouse
0.462	ibm

TF-IDF	Term
1.0	balanced
0.657	warehouse
0.612	software
0.372	base
0.372	c1000

TF-IDF	Term
1.0	table
0.981	sample
0.785	classicscd
0.687	warehouse
0.667	empty

Search Query strings for similar pages:

- infosphere warehouse environment based details
- balanced warehouse notifications software infosphere
- target name warehouse define olap
- agent proxy database warehouse user
- control database warehouse unicode version
- tivoli data warehouse application information
- utility control warehouse database manager
- brick red gc18 warehouse ibm
- balanced warehouse software base c1000
- table sample classicscd warehouse empty

7.7 Future work and conclusion

The experimental model for evaluation will be based on response from users who evaluate a result as being similar or not.

This application evolves as a recommender system in future that recommends results to the user based on his profile. More specifically, we look forward to tailor this application to present results to the user based on his profile.

The preprocessing stage of the application takes into account stop word removal. In order to enhance the performance of the system we are working to build a custom dictionary that can link up similar words and define similarity precisely.

The results obtained in terms of precision and recall signify that the query terms used based on the term weighting algorithm obtain results better than normal search.

The numbers of clicks for similar page results have had a significant growth since embedding the component in enterprise search engine of the organization.

The design proposals mentioned in the previous chapters have been approved (by concerned authorities) and are in process of evolving that into a knowledge management framework.

References

[1] Drucker, P. The Post-Capitalist Executive, Managing in a Time of Great Change, Penguin, New York, 1995.

[2] Hackbarth, G. "The Impact of Organizational Memory on IT Systems", in Proceedings of the Fourth Americas Conference on Information Systems, E. Hoadley and I Benbasat (eds.), August 1998, pp. 588-590.

[3] Von Krogh, G. "Care in Knowledge Creation," California Management Review (40:3), 1998, pp. 133-153.

[4] Alavi, M. "KPMG Peat Marwick U.S.: One Giant, Brain," Harvard Business School, Case 9-397- 108, 1997.

[5] Cranfield University." The Cranfield/Information Strategy Knowledge Survey: Europe's State of the Art in Knowledge Management," The Economist Group, 1998

[6] Daniel Edmund O'Leary, "Enterprise Resource Planning Systems: Systems, Life Cycle, Electronic Commerce, and Risk", Published by Cambridge University Press, 2000, ISBN 0521791529, 232 pages.

[7] Gazeau, M. 'Le Management de la Connaissance,' Etats de Veille, Juin 1998, pp. 1-8.

[8] Davenport, T. H., and Prusak, L. "Working Knowledge", Harvard Business School Press, Boston, 1998.

[9] Alavi, M. "Managing Organizational Knowledge," in Framing the Domains of IT Management Research: Glimpsing the Future through the Past, R. W. Zmud (ed.), Pinnaflex Educational, Resources, Cincinnati, OH, 2000.

[10] Dorothy E. Leidner, “Strategic Information Management: Challenges and Strategies in Managing Information Systems”, Butterworth-Heinemann, 2003, ISBN 0750656190, 625 pages.

[11] Tom D. Davenport, “Cooperative ERP Life-cycle Knowledge Management”, Proceedings of the Ninth Australasian Conference on Information Systems, 29 September – 2 October 1998, Sydney, Australia, pp.227-240.

[12] Creswell, J. (1998). Qualitative inquiry and research design: Choosing among five traditions. Thousand Oaks, California: Sage Publications.

[13] Creswell, J. (2003). Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. Thousand Oaks, California: Sage Publications.

[14] Denzin, Norman K. & Lincoln, Yvonna S. (Eds.). (2005). The Sage Handbook of Qualitative Research (3rd ed.). Thousand Oaks, CA: Sage. ISBN 0-7619-2757-3.

[15] Loseke, Donileen R. & Cahil, Spencer E. (2007). “Publishing qualitative manuscripts: Lessons learned”. In C. Seale, G. Gobo, J. F. Gubrium, & D. Silverman (Eds.), Qualitative Research Practice: Concise Paperback Edition, pp. 491-506. London: Sage. ISBN 978-1-7619-4776-9

[16] Marshall, Catherine & Rossman, Gretchen B. (1998). Designing Qualitative Research. Thousand Oaks, CA: Sage. ISBN 0-7619-1340-8

[17] Thomas S. Kuhn, The Function of Measurement in Modern Physical Science, Isis, 52(1961): 161-193. (quantitative research)

[18] Malthouse, Edward C; Bobby J Calder (2005). "Relationship Branding and CRM". in Alice Tybout and Tim Calkins. Kellogg on Branding. Wiley. pp. 150–168.

[19] Bligh, Philip; Douglas Turk (2004). CRM unplugged – releasing CRM's strategic value. Hoboken: John Wiley & Sons. ISBN 0-471-48304-4

[20] Defect management tools, article found on the web by: quality consultancy P kantelinen Ltd. At the location: http://www.laatuk.com/tools/defect_track_tools.html, Cited as a proprietary of laatuk.com, site last updated: 11th Aug 2008.

[21] Miles L. Mathieu, Ernest A. Capozzoli (2002). The Paperless Office: Accepting Digitized data. Troy State University.

[22] Finlay, R. (2001, March). Bridging the paper-digital document divide. Canadian Underwriter, 68(3), 70.

[23] deJong, Jennifer (2008-04-15). "Mea culpa, ALM toolmakers say", published in SDTimes, November, 2008

[24] Carey Schwaber, August 2006, The Changing Face of Application Lifecycle Management., Forrester Research, Inc.

[25] Nahm, U. Y., & Mooney, R. J. (2000). “Using information extraction to aid the discovery of pre-diction rules from texts.”, In Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000) Work-shop on Text Mining, pp. 51-58 Boston, MA.

[26] Califf, M. E. (1998). Relational Learning Techniques for Natural Language Information Extraction. Ph.D. thesis, Department of Computer Sciences, University of Texas, Austin, TX.

[27] J. Weizenbaum, ELIZA, a computer program for the study of natural language communication between man and machine. Communications of the ACM, (9), 1966.

[28] Christof Manoz, “From document retrieval to questions answering”, a tutorial at the Natural language processing 07 EM-LCT, 2007

[29] Christof Manoz, “Tequesta: The University of Amsterdam's textual Question answering system”, in the proceedings of TREC 10, 2002

[30] K. Bohm and T. Rakow, “Metadata for multimedia documents, SIGMOD record, special issue on the metadata for digital media, December 1994

[31] Keith Goffin, Colin New, “Customer support and new product development - An exploratory study Article Information”, International Journal of Operations & Production Management, 2001,Vol:2, Issue:3, Page:275 – 301,ISSN:0144-3577, MCB UP Ltd.

[32] The ALICE, Artificial linguistic internet computer entity, AI foundation.
More information about the system can be found at: (Site last updated June, 09)
http://www.alicebot.org

[33] Dana Vragitov, Evolutionary sentence building for chatter bots, IUSB, computer and information sciences, 2002.

[34] Aaron galca, “Open domain surface based Question Answering system”, Department of Computer science and Artificial intelligence, University of Malta, Advances In computers, Vol. 17, 186-195, New York, 2004.

[35] Hearst, M. A. (2003). What is text mining? Article Found at:
http://www.sims.berkeley.edu/~hearst/text-mining.html

[36] Hearst, M. A. (1999). Untangling text data mining. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99), pp. 3{10 College Park, MD.

[37] Han, J., & Kamber, M. (2000). Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco.

[38] Hernnandez, M. A., & Stolfo, S. J. (1995). The merge/purge problem for large databases. In Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data (SIGMOD-95), pp. 127{138 San Jose, CA.

[39] Cohen, W. W., Kautz, H., & McAllester, D. (2000). Hardening soft information sources. In Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000) Boston, MA.

[40] Chakrabarti, S. (2002). Mining the Web: Analysis of Hypertext and Semi Structured Data. Morgan Kaufmann, San Francisco, CA.

[41] Monge, A. E., & Elkan, C. P. (1997). An efficient domain-independent algorithm for detecting approximately duplicate database records. In Proceedings of the SIGMOD 1997 Workshop on Research Issues on Data Mining and Knowledge Discovery, pp. 23{29 Tuscon, AZ.

[42] Witten, I. H., & Frank, E. (1999). Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Francis

[43] Kaa ping yee, Kevin li, “Faceted metadata for image search and browsing”, proceedings of the ACM conference from 5-10th April 2003, at Lt. Lauderdale, Florida, USA.

[44] Bligh, Philip; Douglas Turk (2004). CRM unplugged – releasing CRM's strategic value. Hoboken: John Wiley & Sons. ISBN 0-471-48304-4.

[45] Malthouse, Edward C; Bobby J Calder (2005). "Relationship Branding and CRM". in Alice Tybout and Tim Calkins. Kellogg on Branding. Wiley. pp. 150–168.

[46] deJong, Jennifer (2008-04-15). "Mea culpa, ALM toolmakers say", published in SDTimes, November, 2008

[47] Carey Schwaber, August 2006, The Changing Face of Application Lifecycle Management., Forrester Research, Inc.

[48] Miles L. Mathieu, Ernest A. Capozzoli (2002). The Paperless Office: Accepting Digitized data. Troy State University.

[49] Finlay, R. (2001, March). Bridging the paper-digital document divide. Canadian Underwriter, 68(3), 70.

[50] Defect management tools, article found on the web by: quality consultancy P kantelinen Ltd. At the location: http://www.laatuk.com/tools/defect_track_tools.html, Cited as a proprietary of laatuk.com, site last updated: 11th Aug 2008.

[51] Y. Kane-Esrig, L.Shlkar, "Using multiple sources of Information to search a repository of software lifecycle artifacts, Proceedings of Bellcore conference on Electronic document delivery, NJ, May 1994.

[52] Chirata, wolfgang, "Using ODP metadata to personalize search", Proceedings of the 27th Annual ACM conference of SIGIR, 2005, ACM press, Salvador, Brazil.

[53] IBM IPS Clearquest, A tutorial on the use of Clearquest for defect management, Internal IBM publication.

[54] Gerard Salton, Christopher Buckley. Term Weighting approaches in Automatic text retrieval. Information processing and Management, Pergamon press, Vol 24, No.5 pp 513-523, January 1988.

[55] Marcelo Montemurro, Damian Zanette. Entropic analysis of the role of words in literary texts. http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0109218, 2001.

[56] Tatsunori Mori and Miwa Kikuchi and Kazufumi Yoshida, Term Weighting Method based on Information Gain Ratio for Summarizing Documents retrieved by IR systems, Journal of Natural Language Processing, 9(4):3—32, 1999.

[57] Kazunari sugiyama, Kenji hantano. Adaptive web search based on user profiles without any effort from users, ACM – 1-58113-844, WWW 2004, May 17-22 2004, New York, USA.

[58] David Hawking, Challenges in Enterprise search, Fifteenth Australasian Database Conference (ADC2004), Dunedin, NZ. Conferences in Research and Practice in Information Technology, Vol. 27. Klaus-Dieter Schewe and Hugh Williams, Ed.

[59] Yiming Yang, An evaluation of statistical approaches to text categorization, Information Retrieval 1, 69–90 (1999), Lower Aca-demic Publishers. Manufactured in "The Netherlands".

[60] Warren Greiff, A theory of term weighting based on exploratory data analysis. ACM 1-58113-01558, SIGIR 1998, Melbourne, Australia.

[61] Daniel S Tsach, "Information mining with the IBM intelligent miner family", An IBM software solutions white paper, Feb 1998.

[62] Introduction to Infosphere web warehouse, A tutorial by Labuser at IBM software labs on 17th Dec 2008.