UNIVERSITY OF OSLO
Department of Informatics

Unifying IT Support with Knowledge Management

Bishwa Shrestha

Network and System Administration

Oslo University College

May 23, 2011

Unifying IT Support with Knowledge Management

Bishwa Shrestha

Network and System Administration
Oslo University College

May 23, 2011

Abstract

The effectiveness of IT support is determined by the ability to quickly diagnose and solve a problem. The diagnostics process can be aided to a great extent by gathering and presenting the knowledge scattered across relevant domain in real time to the IT Support. With effective knowledge management we can even automate the diagnostics process to a great extent thus leveraging the IT Support from repetition of tasks. In this research work we discuss the challenges and values of incorporating IT Support related knowledge into a Knowledge Management System. We discuss techniques to extract topics from support portals such as Internet Forums and ticketing systems in a form understandable by Cfengine and use the Cfengine Topic Map Model to organize and represent the knowledge.

Acknowledgments

I would like to express my sincere appreciation towards my supervisor Alexander Yngling for his advice and motivation. My gratitude to Professor Mark Burgess for his encouragement, support and valuable insights throughout the project. Thanks to Hårek Haugerud and Kyrre M. Begnum for their guidance throughout the course.

My friends and family have always been there for me. Thank you for your support.

Bishwa Shrestha

Contents

List of Figures

Chapter 1

Introduction

The IT support person must be leveraged from the tedious task of having to go through pieces of scattered knowledge within an organization. The person in the IT Support is one of the most bugged and pressurized person in an organization. Users are often interrupted by the troubles they face with their systems. The work efficiency of people within an organization depends upon how fast the IT support can fix their problems. Generally the IT support person has to have a very good interpersonal skill. However, that skill is backed up by the thorough knowledge of system and problems that a person is facing. The faster the problem is solved, the happier a person is. There are often cases where more than one person faces the same kind of problem. The IT support has to recall and repeat the same solution over and over again in such a case. Can we leave such a repetitive task to machines (softwares) so that the IT Support can focus on other important tasks? To what extent?

It is common for a company to have a forum, blog or wiki or some other kind of system where authorized participants can discuss and share their knowledge. People can search for certain keywords and they will be presented with a list of documents in the discussion that are related to the keywords supplied. The users then have to go through the document by themselves and find out the answers. Sometimes they have to go through multiple documents in order to find the solution to their problem. If they cannot find a solution, they can ask a new question and expect to get a reply quickly, or look somewhere else. It is always easier if someone else has already solved the problem. In such a case, we can find the knowledge "'scattered"' within the system. All one has to do is look for them.

Proper knowledge management can speed up the rate of work. There are many cases when the IT personnel have to solve the problem himself, such as: in cases where the user requires installing software (root access). As IT personnel, we are expected to have solutions to all of user's problems. We cannot direct them to a page on the forum, or somewhere else and leave them to find the solution by themselves. If we don't know, or cannot remember the solution to the problem, it is us who needs to go to the forums, consult others or

forward the request to someone else so that the user gets the necessary help in due time or before he/she gets frustrated. It is desirable for an organization to have its knowledge accessible to all which still remains there when a personnel leaves the company. An IT personnel can access such a knowledge base that contains information from internal documents, forums and other experts in real time so that he can deal with the problems at hand. If new problems are encountered then there must be a way for the IT Support personnel to add it to the Knowledge base. That way the and organization's knowledge stays within itself and the process of troubleshooting, fixing a problem or learning by new personnel can be speeded.

1.1 Motivation

Most organizations have their own knowledge base and knowledge management systems, however primitive they can be. They can be in the form of document based systems, blogs/forums, databases etc. It is a tedious job for an IT support personnel to go through these documents and gather knowledge about the product/system. If a support ticket arrives, the support personnel needs to search through the huge amount of information in order to find out all the tickets/forum topics related to this and try to find out a solution easily. A better option could be to start solving the problem from scratch which is time consuming and leads to duplicating information. Moreover, keeping track of the problems and solutions gets harder as the amount of information gets bigger.

The motivation towards unifying IT support with Knowledge management came through the study and usage of Cfengine's Knowledge management system. Cfengine is a configuration management tool ... . it provides a configuration language for managing a knowledge base that can be compiled into a topic map. [1]

The concept of treating knowledge as a configuration of ideas [2] makes it simpler to model a Knowledge management system using the Cfengine topic map model. The knowledge map already contains documentation, guides and all the information about the promises applied organization wide. It would be of great value to add the support information into the Knowledge map as well.

1.2 Problem Statement

The knowledge in various IT Support portals can be accumulated and linked together and used by the support personnel to provide a better support. In this thesis we

> *Suggest a Unified model of IT support aided by knowledge management.*

For this, the following needed to be done first:

1. *Extraction of topics from forums and support tickets automatically.*
2. *Use Cfengine Topic map model to add the automatically generated topics and knowledge added by human experts into the Knowledge map.*

1.3 Thesis Outline

This paper shall be outlined in the following manner:

- Chapter 1 gives a brief overview of the topic and makes the problem statement.
- Chapter 2 presents the background and theory behind the topic and related subjects and terminologies used in this paper.
- Chapter 3 contains the methodology, design and implementation of concept.
- Chapter 4 is the results section and contains the results obtained from the implementation setup.
- Chapter 5 the results presented in Chapter 4 are discussed.
- Chapter 6 is the concluding chapter which talks about the findings of this thesis in relation to questions raised and the motivation for this topic. Possible future work and development of this concept is also addressed.

Chapter 2

Background and literature

In this section we discuss the concepts related to IT Support and Knowledge Management.

2.1 IT support

IT support is the place people generally go to when something undesirable happens to their systems. The IT personnel are responsible for the continuity of services. As more and more people and organizations are being introduced to IT services and products, the importance of IT support is ever increasing. However, it still remains a rather unexplored field[3]. A lot of time is spent daily on small housekeeping tasks such as: cleaning up disk spaces, keeping the internet connection alive, locating files, monitoring changes, etc. Although applications like Cfengine are making life easier for the System Administrators by performing these tasks in a convergent manner, the major areas of the IT Support are still being left out. A considerable amount of time is being spent daily battling with IT related problems. *End-user time spent on non-job-related PC activities accounts for more than 40% of a PCs total cost.*[4].

With a large group of people being familiarized with IT, the number of problems are also high. Moreover, it is interesting to note that people are facing "similar" kinds of problems. The Internet has been an excellent way to communicate, interact and solve problems. There are thousands of blog posts, forums, support portals, commercial/non-commercial websites with how tos, walkthroughs, troubleshooting guides that can be accessed easily using popular search engines. Although these search engines make life easy by finding solutions, the main drawback is that one is presented with too much information! Browsing through the list of solutions becomes a time consuming task in itself.

2.1.1 IT Support Portals

There are a lot of ways for organizations and businesses to provide support to their customers. Some of the popular ways being used today are discussed here.

1. **Internet forum**

   Also known as message board, discussion group or bulletin board, an Internet forum allows users to post messages and interact with each other. An Internet forum is a collection of posts: messages wrapped up in threads: discussions. It has proved to be a very useful tool for providing support to the users of a technology. Careful study of discussions in a forum can reveal deep insight into what kind of problems are commonly faced by users and how they are solving it. The administrators and moderators of a forum are generally the IT support personnel who can provide support to their customers. Maintaining such portal saves a lot of time and effort of a support personnel as users can browse through the forum topics themselves, get involved in a discussion to find the solution to their problem. Moreover, the users themselves are proactively taking part in helping each other find solutions. It is somewhat a self organizing community of online presence.
   However, the problem arises when the number of discussion start growing. Finding a solution then could take much time and effort.

2. **Wiki**

   With the success of Wikipedia(the online encyclopedia), organizations are starting to keep their own wikis which let the users write and modify articles collaborately.

3. **Corporate Blogs**

   Corporate blogs can be internal and external. Internal blog is used for discussion and sharing of knowledge by the employees whereas external corporate blogs can be used for informing the customers about new products, plans for upcoming releases etc. The users might or might not be allowed to express their views in these blogs.

   A corporate blog might be considered informal, however they hold a plethora of knowledge that can prove valuable it used properly. Each blog post usually goes through a careful editing process before being

posted which means that the content needs less cleanup before being analysed.

4. **IT Support Personnel Group**

An IT Support personnel is one who is employed to make sure that customers' problems are solved in time and that they know how to use the features of the product correctly. According to Noel Burton [5] *User support is a specialist function which retains, on behalf of the companys user population, technical knowledge about IT and the way the company uses it, in order to deliver that knowledge in a focused form to solve specific technical and business problems on both a reactive and proactive basis, such that user productivity is maintained and enhanced, thereby further enabling the user to contribute to the companys business goals.* The IT Support Personnel Group is described in the ITIL v3 as Service Desk. According to [6] the service desk is a *Single Point of Contact between the Service Provider and the Users. A typical Service Desk manages Incidents and Service Requests, and also handles communication with the Users.*

The IT support has different responsibilities is different organization. However one of the most common responsibilities of the IT support is to diagnose and solve a problem. It is the primary task of an IT support to respond to customer requests which might arrive through various mediums such as emails, ticketing systems and telephone calls, as soon as possible. Normally customers seek support when something has stopped working or something is hampering normal operation. The IT support must find a way to restore normal operation with minimal business impact.

Ideally a support personnel is required to know everything about the company and the ptoduct he/she is assigned to support. But knowing everything is almost impossible, so the support person may need to consult with other people or read documents and go through old archives in the process of finding a solution to the problem. A good knowledge management system can create such an environment for the people as the support desk. Among others, the knowledge base can hold all the previously known errors and their solutions or the discussions related to them. This saves a lot of time and effort for the IT Support.

5. **Ticketing System**

A ticketing system is a web-based portal where customers can directly post incidents and feature requests. A ticket contains a message with

the description of the problem/feature and the details of the user reporting it. The IT support can also create a ticket and delegate task to other responsible personnel in the organization. This case arises when the IT support cannot solve the porblem by itself. The ticketing system contains a history of all the incidents and features requested over time.

6. **Documentation**

   The online documentation and reference manuals are the officially created by the company. It contains the technical details about cfengine and is well structured. The information in the documentation is organized into the company's knowledge map and is available in real-time through the Cfengine Nova.

2.1.2 Problems faced by IT support

IT support need to analyse and diagnose the problems quickly in order to help the user. However, the solution might not always be found as fast as it was expected. The support personnel cannot remember everything and cannot know everything; as such, there are chances of solving a problem that has already been solved, or taking longer period of time consulting with others about the problem.

2.2 Knowledge: An Intellectual capital

2.2.1 Knowledge

Knowledge is a vague concept. There is no single definition of knowledge that has been agreed upon by everyone. Plato's definition of knowledge as: "'justified true belief"' though accurate is not enough in the IT industry which always demands clarity. Professor Mark Burgess of Oslo University College argues:

> *Knowledge has much in common with configuration: what after all is knowledge but a configuration of ideas in our minds, or on some representation medium (paper, silicon etc).* [1][2]

Such a definition makes it easier for programmers, system designers and those who are involved in the field of knowledge management in Information Technology. It gives us the confidence that Knowledge can be retrieved and stored in any form. Computer systems are getting very good at storage as well

as knowledge retrieval algorithms. The ideas we have, the things one knows, experience about how to solve problems, trouble shooting hardware and software, vague ideas about what we think might have caused a specific problem etc can be carefully put into documents. With next generetion technologies such as Cfengine and its cf-know agent, these knowledge can be arranged just as normal configuration files.

One can always argue about the inaccuracy in representing absolute knowledge. As does Marvin Minsky in "'The Society of Mind"':

> *The things we deal with in practical life are usually too complicated to be represented by neat, compact expressions. Especially when it comes to understanding minds, we still know so little that we cannot be sure our ideas about psychology are even aimed in the right directions. In any case, one must not mistake defining things for knowing what they are. Marvin Minsky, The Society of Mind, 1985.*

However, there is a need to change in the way we think and incorporate new ideas to solve the problems being faced. As we keep adding values to the new ideas we start understanding more, the value of a new approach or idea. Documentation has been the primary means (developed) by humans to transfer knowledge between generations. Civilizations that didn't learn how to write or document things are unknown to the world, so is the immense knowledge gained by their experience. Businesses and economy are one of the driving forces of modern world and IT is being the driving force behind them.

Organizational Knowledge has immense value and potential to businesses, specially in the current age where the frequency of new employees joining a company and old ones leaving is high. Leibold, Probst and Gibbert state the importance of knowledge as follows:

> *The new source of wealth is knowledge, not labor, land or financial capital. It is the intangible, intellectual assets that must be managed.* [7]

Knowledge is context-specific. What might be considered as highly valued in one domain might be completely meaningless in another domain. Hence it is necessary to define the context of knowledge.

2.2.2 Knowledge Management

European Committee for Standardization (CEN) defines knowledge management as "the management of activities and processes for leveraging knowledge to enhance competitiveness through better use and creation of individual and collective knowledge resources". According to CEN there are five core knowledge activities in the knowledge management framework of CEN: identify, create, store, share and use [8].

Hong and Ståhle have identified four perspectives of knowledge management [9]:

1. The philosophical and psychological perspective (what is knowledge, where does it come from),
2. The organisational and sociological perspective (how to create and master knowledge)
3. The economic and business perspective (how to extract value from knowledge) and
4. the technological perspective (how to provide efficient and effective tools for managing knowledge)

We are concerned with the the organisational and sociological perspective and the technological perspective.

Varintorn and Nazrul [10] have divided the knowledge management architecture into 4 elements:

1. Knowledge components
2. knowledge management process
3. information technology (IT)
4. organizational/corporate culture.

In their study they also identified four main sub-processes of Knowledge Management:

1. Knowledge Creation and acquisition
2. knowledge organization and retention
3. knowledge dissemination
4. knowledge utilization

"For effective and efficient knowledge management, it is necessary to incorporate various types of technology [11]. With this in mind, the new technology for knowledge management should be developed in a way that it is able to perform various functions within one system with respect to other technologies. It should not work only by itself as in the past. However, it is up to the company (with its resources and capabilities) to select the sort of technology that can facilitate or support knowledge management in the organization more effectively."

Gandhi argues that KM is not merely information collection, organization, presentation, storage and retrieval. KM is also not a linear, static process. It is

dynamic and cyclical process that requires users to continuously engage with information, acquire new knowledge, apply it to improve decisions, create new information and knowledge in the process, apply that new knowledge to new situations, and so on[12].

The field of Knowledge Management is still undergoing research and has not reached its academic maturity yet. However, businesses and organizations are recognizing the value of Knowledge Management. With plethora of Knowledge Management softwares available in the market, both open source and commercial, we are entering an age where knowledge sharing and retention is valued. IT and Knowledge Management move together in symbiosis. First IT was used to manage knowledge, now various areas of IT needs knowledge management. IT support is one of them.

2.2.3 Topic Maps

Topic Map is a standard for the representation and interchange of knowledge, with an emphasis on the find ability of information[**?**]. The ISO standard is formally known as ISO/IEC 13250:2003. A topic map is an index of key concepts extracted from document(s) interconnected by their relationships with each other. A topic map consists of Topics, Associations and Occurrences (TAO).

The Topic Map Model

The topic map model separates the classification from the topic itself. By doing this there is a freedom to put things anywhere and into any category and still be able to extract meaningful knowledge.

The classical way of classification that creates hierarchies separates entities completely from each other. It puts the entities into separate boxes or shelves that that are labeled by categories. This method of categorization soon drains out its usability when the number of categories grow. David Weinberger discusses how knowledge is being un-boxed and being converted into "'Smart Leaves"' in [13].

There are number of inherent drawbacks of the hierarchical classification system:

- Too many categories
- Non-uniform distribution of things into categories
- If a thing belongs to more than one category, it has to be present in both places. We might of course create shortcuts, but that again might grow in number and become unmanageable.

- Addition of a new category needs review of the entire existing content.

The Topic Map Model in instead provides a way to link concepts to concepts and information. There is no hierarchy, anything can belong to any type.

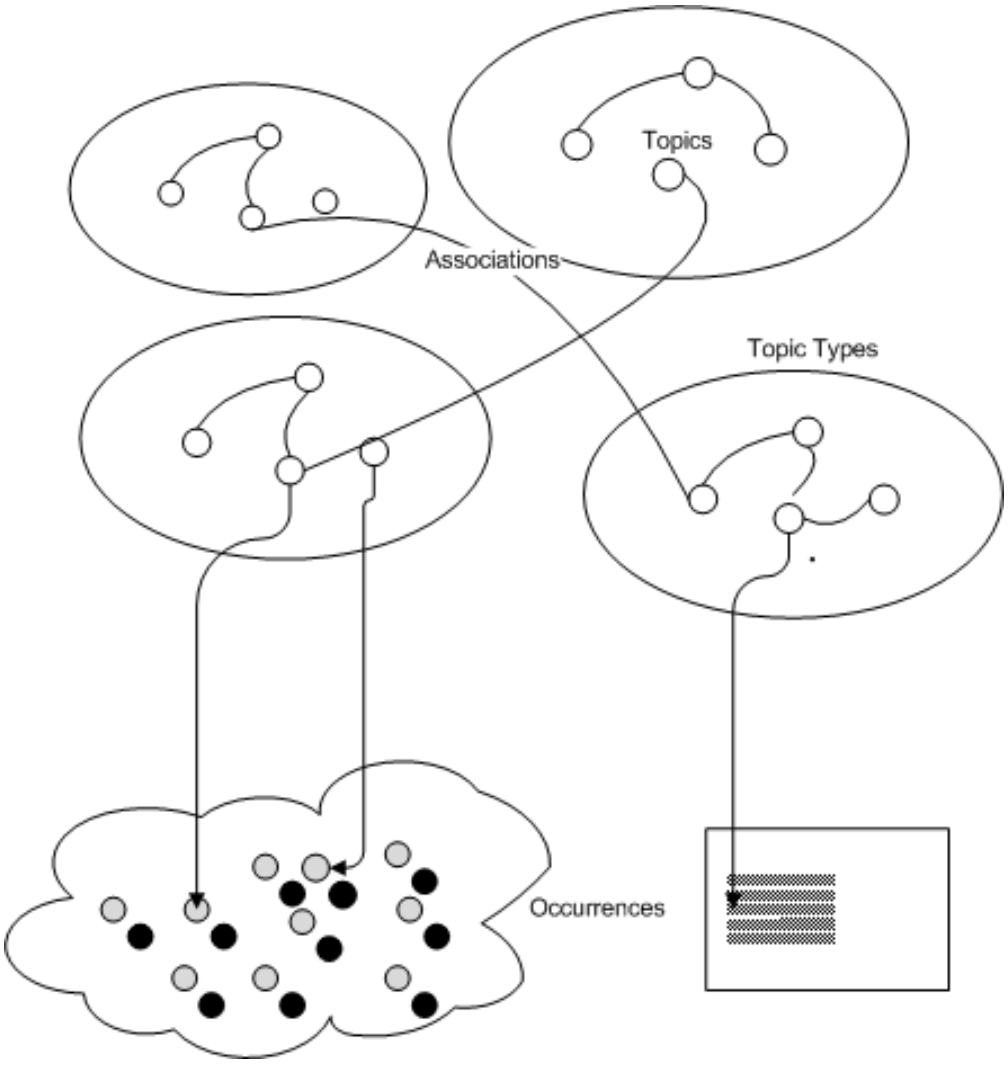

Figure 2.1: Topic Maps can form a network of related entities

As seen from the above diagram, a topic can belong to types, it can have links to multiple topics and occurrences. Steve Pepper explains that with Topic Maps there is a separation of two layers of Topics and Occurrences: the knowledge layer and the information layer[14].

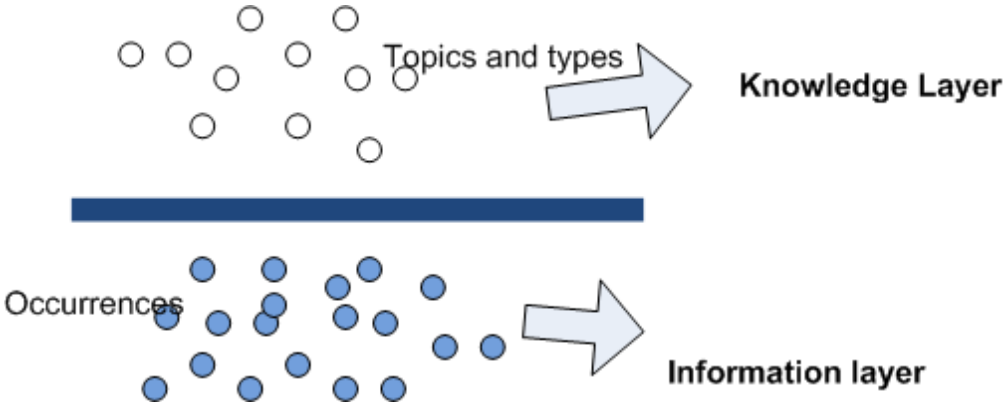

Figure 2.2: Topic Maps create two layers: Knowledge layer and Information Layer

2.3 Promise Theory and CFEngine

2.3.1 Promise Theory

At the heart of Promise Theory is the concept of autonomous agents behaving at their will. This fairly new theory was proposed by professor Mark Burgess at Oslo University College for modeling Cfengine's autonomous behaviour. With increasing number of networked systems, determining the behaviour of a particular system is almost impossible. A system the that behaves in a certain manner in a networked system might not behave in the same way in a complex networked environment. There are a lot of factors that affect it as a result of which unexpected behaviour might shown by the system. Promise Theory paves way for understanding the behaviour of agents in complex networked environment which would not have been possible in normal ways.

"'Voluntary Cooperation"' and "'Convergence"' are the key concepts in Promise theory. The idea the the behaviour of any agent cannot be influenced by external factors and that they do just what they "'want"' to do makes it immune to attacks and malicious behaviour. It is a diversion from conventional way of thinking that a system must do what it was programmed to do. It is an obligation which is open external influences. Moreover, when an agent is "'forced"' to do something that it might not be able to do, then unexpected things can happen. Hence, the behaviour of such systems are unpredictable. The idea of agents operating at their will suggests that its behaviour is more predictable. We can say things like "'This agent can never do that!"', and an agent not behaving at its will means that it cannot do things (or cannot be made to do things) that are out of its scope. This gives a way to design robust systems.

Convergence is synonymous to self-healing behaviour. With defined and predictable behaviours of agents in a system it can have a desired state. The agents work together to get into this desired state no matter what changes might occur in the system. In a complex environment, changes are bound to happen. The concept of convergence gives us a way to define the desired state of the system and be ensured that it always tries to be in that state.

Promise Theory is based upon: (1) Intentions, (2) Documentation and (3) Predictability. A system's intentions must be known before hand and well documented. Organizations spend lots of resources to make their system behave the way it should. Predictability of a system's behaviour avoids unnecessary troubles and makes the system easily adaptable to any kind of environment.

A promise is a directed link that consists of two autonomous agents and a promise body, expressed

$a1 \xrightarrow{b} a2.$

Agent a1 (promiser) offers a behavior, while agent a2 (promisee) utilizes it. The promise body b describes the nature and constrain of the promise. A promise is expected to be kept and verified.

2.3.2 Cfengine

Cfengine is an agent-based, policy controlled configuration management system developed at Oslo University College by Mark Burgess. It primarily provides automatic configuration management, monitoring and control of computers, data centers and even embedded systems. The flexible framework provided by Cfengine makes it a useful tool to research various areas of configuration management. It also provides a robust framework to study knowledge management. The Cfengine Topic map model allows users to write policies about organizational knowledge and store them in a well structured manner in the company's knowledge base. Cfengine has the capability to project complex relationship from the various topics available.

Cfengine has the following characteristics, quoted from[15]:

- *Centralized policy-based specification, using an operating system independent language.*
- *Distributed agent-based action; each host agent is responsible for its own maintenance.*
- *Convergent semantics encourage every transaction to bring the system closer to an ideal average state, like a ball rolling into a potential well.*
- *Once the system has converged, action by the agent desists.*

- Document every action.

2.3.3 Components of Cfengine

Cfengine consists of a number components that function different purposes. However they can work together to achieve a common goal. These components are described briefly below.

- **cf-agent**

  The cf-agent is the heart of Cfengine. It is an autonomous agent that reads the desired state of the system from a policy file and works to take the system to that desired state. The policy files contain bundles with promises that need to be kept for that system.

  cf-agent's primary function is to keep those promises. It runs at predefined interval (reconfigurable) in the background looking for changes in the system and trying to keep the system in a desired state.

- **cf-execd**

  The cf-execd runs cf-agent at regular interval.

- **cf-know**

  The cf-know collects knowledge about system and promises(automatically generated), documents(semi-automatic) and other organizational knowledge(manually entered) and stores them in a topic map model that is unique to Cfengine. The knowledge is assumed to be static and can be regenerated entirely from policy files should they get lost in any way.

- **cf-monitord**

  It is monitoring agent that runs in the background collecting various data about the host that are desirable for analysis.

- **cf-promises**

  The cf-promises validates syntax of the promises defined in a policy file. It is also used to generate the configuration for cf-know about the enterprise knowledge in the commercial edition.

- **cf-runagent**

  The cf-runagent is used to run cf-agent remotely.

- **cf-serverd**

  The cf-serverd acts as a policy distribution point for all other agent. Every machine has the capability of becoming a server, however a machine

into which the other clients are bootstrapped functions as the policy server.

- **cf-report**

  It generates reports to the user from databases.

- **cf-key**

  It generates private/public key pairs for secure communication.

- **cf-hub**

  This component is only available in the enterprise editions in which it function as a tool to collect data from connected hosts. This data is collected locally by cf-monitord.

Cfengine has evolved and grown into a robust framework over the years. It has been used as a primary tool my many students and researchers in the field of system automation, configuration management, system monitoring, system modeling and in recent years, due to the development of the Cfengine topic map language along with the complete rewrite of the Cfengine, it has grabbed the attention of knowledge management researchers.

2.3.4 Cfengine Topic Map Model

The basic idea behind the ISO Topic Model and the Cfengine Topic Map model is the same, ie. TAO. The Cfengine Topic map model however simplifies the process of encoding topic maps. Knowledge can be represented in the Cfengine Topic Map model as follows:

```
topics:

topic_type_context::

"Topic name"

comment => "Use this for a longer description",

association => a("forward assoc to","Other topic","backward assoc");

"Other topic";

occurrences:

Topic_name::
```

```
"http://www.example.org/forumid1234"

represents => { "Forum Post"};
```

2.3.5 Representation of knowledge using topic maps

A topic can be a word or a phrase representing a problem or an inquiry. An occurrence can be a body of text in any of the support portals (forums, bug trackers, documentation etc). An association is the relationship that ties topics with occurrences and topics with topics. Hence, a problem (*problem1*) can have the following relationship with a web page (*forumpage10*) in the forum:

$$problem1 \xrightarrow{\text{is about}} forumpage10$$

$$problem1 \xrightarrow{\text{is discussed in}} forumpage10$$

$$problem1 \xrightarrow{\text{is solved by}} forumpage10$$

The relationship between *problem1* and another problem (*problem2*) can be expressed as:

$$problem1 \xrightarrow{\text{is related}} problem2$$

In this way the topics and discussions in various support portals can be combined together in a unified view. In such a Knowledge Management system, if a support ticket is submitted by the customer, the following scenarios may arise:

1. The Support Personnel knows the solution to the problem and responds to the customer immediately. In such a case the support personnel should be encouraged and motivated to enter the problem with the solution into the knowledge base. If the entry already exists, then at least the recurrence of the problem must be documented. A variable can be incremented every time to indicate that the problem was encountered so that it can be used for future statistical analysis.

2. The Support Personnel does not know the answer to the problem. He/she queries the knowledge map and is directed to a solution in one of the support portals. The problem might have already been encountered by another customer in which case, the answer is found in the ticketing system itself. The problem might also be discusses in the company's forum which may or may not present a solution. There might be a document for the specific problem - a user guide, troubleshooting guide which might.

Either way, all the topics related to the current problem can be presented before the IT Support so that he/she can make correct and immediate decision to help the costumer.

3. No topic related to the new inquiry is found. In such a case, the IT Support might add an entry to the Knowledge base with the status unsolved. Such entries can be updated later when suitable solutions are found.

Knowledge Management is a continuous process. With sudden, rapid changes taking place in the IT industry, today's knowledge might be outdated tomorrow. However, Cfengine believes in the convergent philosophy. After a while some elements stop changing and still retail their value. Steps must be taken such that such knowledge is not lost and readily available when in need.

Professor Mark Burgess of Oslo University College and Professor Alva L. Couch of Tufts University have been working on ways to infer causal relationships between entities and topics. They even present a way of automatically generating human understandable stories about problems and related solutions. Using the techniques discussed in [16] [17] for troubleshooting systems and browsing through related documents, a directed graph from the initiator(problem) to the terminal(solution) can be generated which contain all the intermediate variables that affect them.

Cfengine has the ability to create a Knowledge Map from the internal Configuration Management Database (CFDB) that can be compiled to generate a Knowledge Map using the Cfengine Topic Map Model. It also links individual topics to sections in the documentations, unit tests and examples.

The commercial version Cfengine Nova (currently at version 2) offers all these abilities and opens doors to great possibilities. Configuration Management and Monitoring tools are usually designed for System Administrators to aid them in managing resources and keep the IT services up and running. Cfengine seems to be a special case. It has evolved through the years, with academic research being the foundation for the design and operation, into a tool that can be used also as a Knowledge Management System. It has, in its design, the ability to document almost every kind of knowledge that can be formulated into a promise. Cfengine can be tailored to formulate promises from support ticket or an error description that can in turn be documented in the company's Knowledge Base.

2.3.6 Suggestions for using Knowledge Management for IT Support

IT Support have been using knowledge management extensively. The person handling the customer requests usually maintains notes about problems that occurred in the past, their solutions, and other related problems and their diagnosis. These notes may be in a physical notebook, a database, excel sheet or

simple piece of paper. This is an indication that people need help to remember. As the company grows or the number of services grow, so do the number of incoming requests from the customer. With lots of requests at the disposal, it becomes tough to prioritize them and reply to them in due time. IT support has been criticized in many ways. It is however not due to the lack of expertize or lack of responsibility in the part of the employees, it is due to the sheer volume of support requests, boring induced due to repetitive tasks and lack of a good knowledge base.

Many research and standards have suggested that the IT support should maintain a good knowledge base. Wang et. al. suggest the use of a separate Knowledge Management Database and a Configuration management Database instead of just the latter and point out some key features of Knowledge Management Database[18].

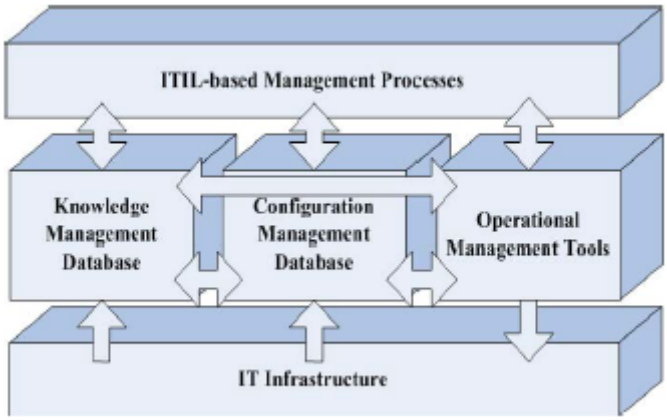

Figure 2.3: Enhanced ITSM Architecture with KMDB [18]

In small organizations the role of IT support and System Administration is indistinguishable. Hence, IT support and System Administration very much related to each other.

Consortium For Service Innovation have defined some best practices for Knowledge Centered Support(KCS) in their KCS Practices Guide. They have also designed an ideal template for the User Interface that can be presented to the Support personnel that combines the information from Incident Management system as well as Knowledge Management System. CSI suggest the use of articles [19], which are templates for each piece of knowledge which can evolve over time due to the interaction of users and support personnel.

[20] suggests two ways for effective IT support aided by Knowledge Management:

- Maintaining a knowledge base where IT staff can create and retrieve knowledge
- Give access to end-users to the knowledge base via a web-based self-service.

Tools that make knowledge management easy are not just important but necessary. However, the change in the mindset of everyone involved in the knowledge management process, even end users is necessary for knowledge management to succeed. Users that participate in Internet forums and other support portals should be made aware that whatever they discuss can play an important part in enhancing the support they receive. Efficient organization of content in various support portals must be encouraged and IT support personnel should be motivated to add whatever they know into the company's knowledge base.

2.3.7 User Interaction

The principle reason for the slow development in the field of Knowledge management is the limitations imposed by the system to the Users that are using them. Although the design or modeling of a user interface to a knowledge management system is beyond the scope of the topic, it is a topic worth considering. The ways the user of a technology might or should interact with a system should be carefully considered from the start of any project.

Many researches have proposed various ways of representing knowledge systems. Various ways of representing knowledge have emerged such as Resource Description Format(RDF), Topic Maps, DARPA Agent markup Language(DAML), Web Ontology Language(OWL), CFEngine Topic Map Language etc. Users of such Knowledge systems are required to achieve certain level of understanding to interact with such systems. These representations are for applications, not necessarily for humans. Furthermore, these representation systems are required to be maintained by experts. Such a requirement has its own set of benefits and drawbacks.

IT support personnel can be trained to use and maintain a portion of the Knowledge Management System. However, people working in IT support are constantly in communicating with layman users and often have a hard time understanding the requests and problems.

2.4 Knowledge Mining

2.4.1 Text Mining

Text mining is an essential part of knowledge management as it is used for gathering knowledge from a block of text[cite Text Mining Handbook]. Peter

Norvig presents a method of extracting natural language words and phrases from unstructured document in[21]. Google has provides a list of most frequently used words and common spelling errors which can be useful while gathering data which can be used to map the words and phrases to.

2.4.2 Information Extraction

We also need to extract topical information from a single forum post, a document or a support ticket such that they can be linked with the existing knowledge map. The big challenge is figuring out what an article or discussion is about. One method is to count the number of tokens in the article and create a histogram. The topic with the highest occurrence can be considered as the topic being discussed.

Researches have found many methods for extracting text from documents. Hsu and Yih discuss a template based information extraction technique in [22]. Several researches have identified probabilistic ways to extract topics from documents [23].

Chapter 3

Model and Methodology

3.1 Tools and Algorithms

The following opensource libraries, algorithm and proprietary software were used for the completion of this research work.

- **libcurl**[1] - for downloading web pages
- **libxml2**[2] - for parsing HTML documents into plain text
- Word segmentation technique described in [24] - for extracting words and phrases.
- **Cfengine Nova v2.0**[3] - for knowledge map generation

3.1.1 libcurl

*libcurl is a free and easy-to-use client-side URL transfer library.*It is highly portable and supports many protocols (21 specified in the website) including HTTP, HTTPS and FTP. It has an easy-to-use C library interface which was used to download web pages.

3.1.2 libxml2

Libxml2 is a C XML parser available under the MIT license. It is also highly portable. libxml provides a module called HTMLParser that can be used to parse complex HTML documents. It also provides a module for using xpath queries to find tags and text within a html document.

3.1.3 Natural language Corpus Data

The algorithm described can be used to convert unstructured text independent of punctuation into words and phrases.

[1]*http://curl.haxx.se/libcurl/c
[2]*http://xmlsoft.org
[3]*http://cfengine.com/nova

Example:

'itisatruthuniversallyacknowledged'

can be transformed into

'it', 'is', 'a', 'truth', 'universally', 'acknowledged'

3.1.4 Cfengine Nova

Cfengine Nova is a commercially licensed version of the core Cfengine Software(opensource). It offers a knowledge-enhanced framework for configuration management. The cf-know component reads enterprise knowledge from policy files and creates a knowledge map based on the Cfengine topic map model.

A Simplified algorithm

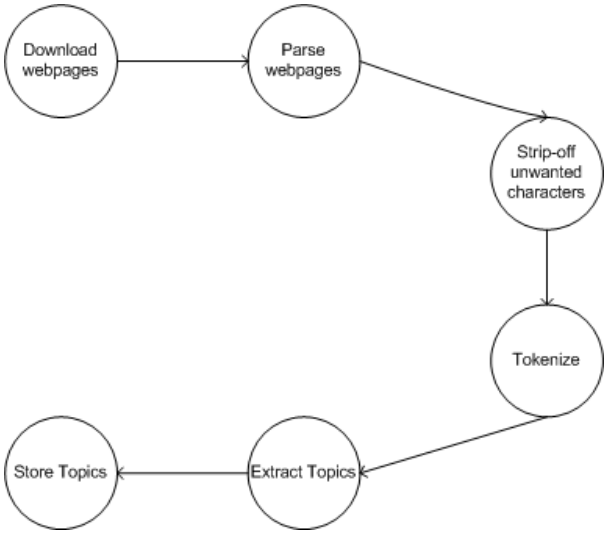

Figure 3.1: Extraction topics from the webpages

3.2 The unified model

Websites have the traditional way of categorizing things hierarchically. Things are classified and kept into subdirectories. Similar is the case for sub domains of a website. If an organization want to create a sub domain, the usual way is to create a directory and give in a public name. In this way, the data are scattered across multiple folders/cabinets.The unified model accepts this hierarchy. However, it encouraged the sharing of knowledge between individual entities. The entities being the different support portals. Each portal must

retain knowledge in such a way that they can be easily connected with the Knowledge Map.

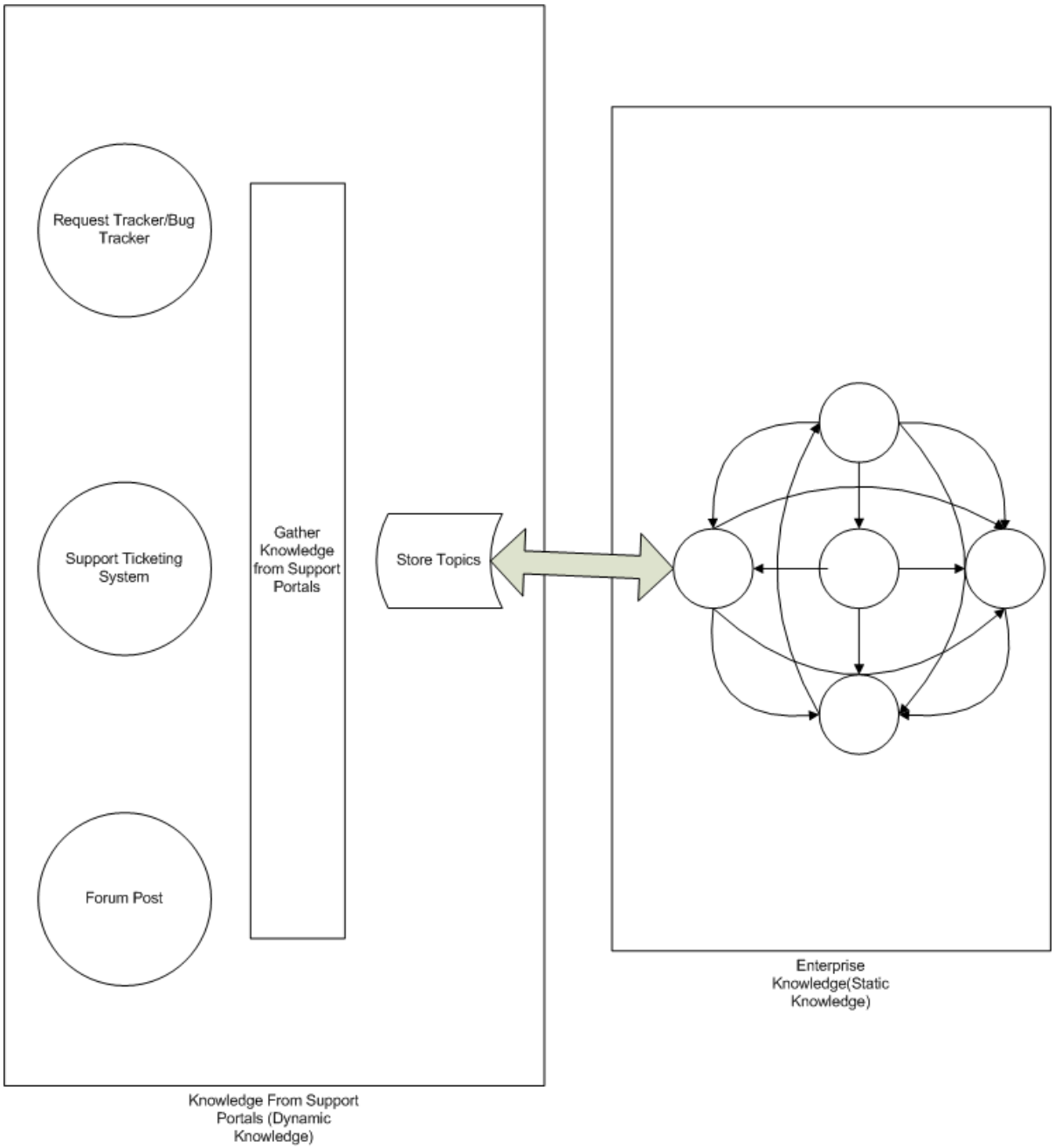

Figure 3.2: Initial Model for Unifying IT Support

3.2.1 The associations

Knowledge between these portals can be connected using the topic map using relationships such as:

- 'is related to'
- 'is mentioned in'
- 'issolved by'
- 'is discussed in'

Since the Cfengine Topic Map model supports both forward and backward associations, these relationships can have the following inverse relationships:

- 'is related to'
- 'mentions'
- 'solves'
- 'discusses'

This relationship is shown by the following figure:

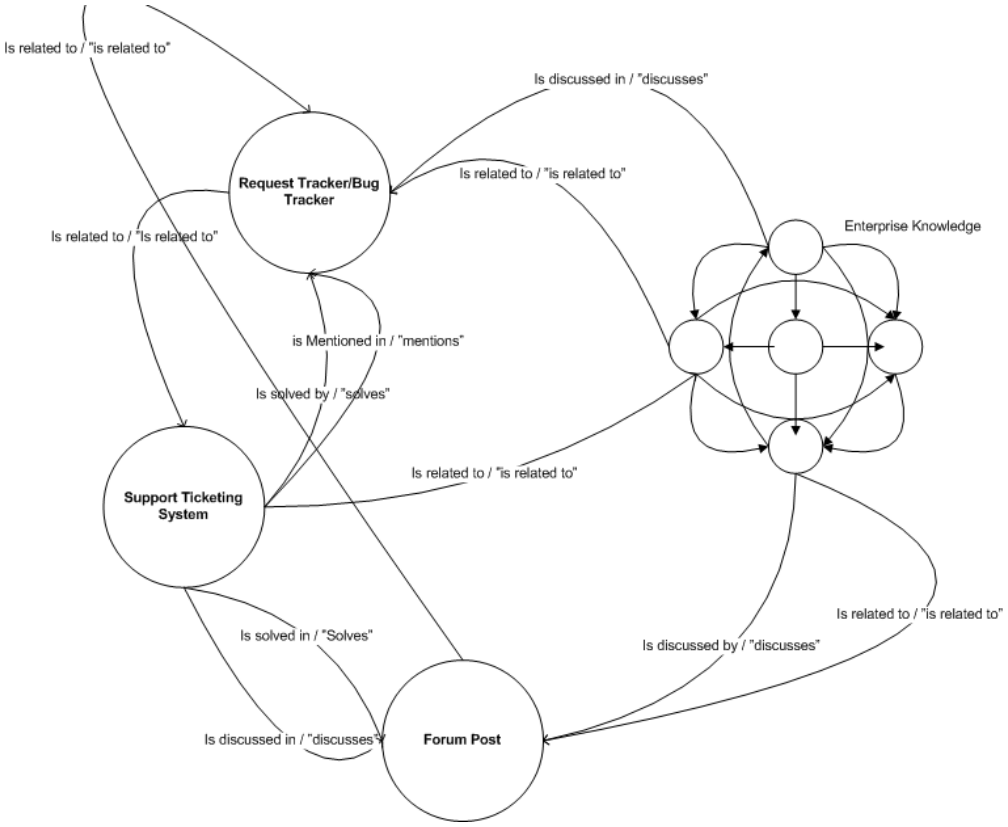

Figure 3.3: Determining and adding associations to topics from different domains

These associations are represented in the Cfengine Topic Map Model as follows:

```
topics:

topic_type::

'topic name'

a('is related to','topic2',''is related to'),

a('is mentioned by','topic3','mentions'),

a('is solved by','topic4','solves'),

a('is discussed by','topic3','discusses');
```

3.3 Gathering Knowledge from Support Portals

Knowledge Gathering

When knowledge is taken as a configuration of ideas, it becomes easy to organize it as we have many tools available for efficient configuration management. We believe that knowledge can be managed efficiently in a manner in which configuration management is done. Tangible knowledge can therefore be managed effectively within an organization.

3.3.1 Manual Entry

The best known way to extract knowledge is to use human experts. They have the best processing system - the brain, which is also a repository of immense knowledge gained through learning and experience. Knowledge can be encoded in the CFEngine Topic Map Model easily. For example, the following piece of knowledge:

> Error number 30012(reported in URL1) with error description "'Unknown built-in function"' while installing Cfengine can be solved by solution1(id: 5157)

```
topics:

cfengine_3_installation::
```

```
"‘ERROR_30012"’

comment=>␣"‘Unknown␣built-in␣function"’,

association␣=>␣a("‘is␣solved␣by"’,"’solution1"’,"’solves"’);

occurrences:

"‘ERROR_30012"’::

"‘http://URL1"’

represents␣=>␣{"‘Error␣Report"’};
```

The manual method is tedious, however it is very accurate and wasily understandable by other people. However, there might be some areas which were not discovered by manual inspection, these might be found by other approaches such as the probabilistic approach.

3.3.2 Internet Forum Data

The manual method though accurate might not be efficient for large volume of text. The forums and ticketing systems contain huge amount of text. going through each and every entry is not feasible. Algorithms are good at extracting text from documents. However, when the text is natural language text and unstructured as one would expect in some kind of documents, the process involved can get much more complex.

In order to analyse the forum entries, first we needed to download the webpages into a local machine. This was done by using the libcurl library. It provides functions in C that can be invoked to download files from HTTP and HTTPS protocols.

An Example is given below:

```
curl = curl_easy_init();

    curl_easy_setopt(curl, CURLOPT_URL, argv[1]);

    curl_easy_setopt(curl, CURLOPT_ERRORBUFFER, curl_errbuf);

    curl_easy_setopt(curl, CURLOPT_NOPROGRESS, 0L);

    curl_easy_setopt(curl, CURLOPT_VERBOSE, 1L);
```

```
curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_cb);

curl_easy_setopt(curl, CURLOPT_WRITEDATA, &docbuf);

err=curl_easy_perform(curl);
```

The discussion in the company's Internet Forum had the following generic form (disregarding the from page where the discussions are listed with the author's name, the number of views, number of replies, etc ...)

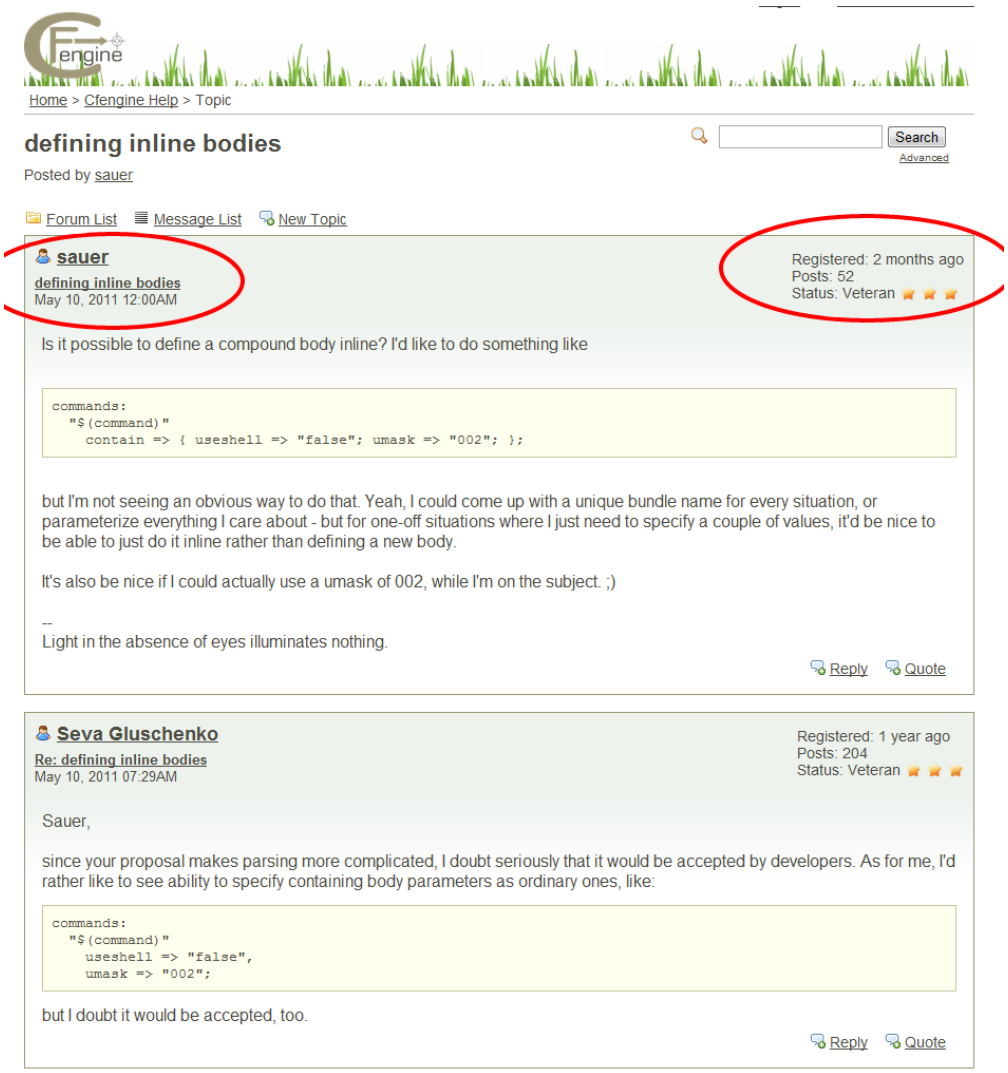

Figure 3.4: A Typical Forum Post

This page had the following internal HTML formatting:

Figure 3.5: HTML source showing user name and user id

Extracting any readable text from this HTML file needed the use of a HTML parser. The libxml2 library provides a robust module called HTMLParser which can parse almost any HTML page. A sample code for initializing the parser is given below:

```
xmlDocPtr getdoc (char *docname)

{ htmlDocPtr doc;

  doc = htmlParseFile(docname, "utf-8");

  if (doc == NULL )
   {
    fprintf(stderr,"Cannot parse document\n");
    return NULL;
   }

  return doc;
}
```

The parser creates a tree of nodes and textual data. This tree must be traversed in order to get meaningful information. The desired data can be retrieved by simply searching for the tags. This can be done by recursively traversing the tree and performing string comparisons and pulling out data into a data structure.

The data structure was defined to store the following information about each discussion(thread):

- Title
- Thread owner
- Date
- Post ID
- Number of replies
- All replies

Each thread has a thread id and each reply has a reply id that can be used to construct URLs that point to a particular thread or even a particular reply to a thread.

The following user information could be extracted from each forum post:

- User name
- User ID
- Registered time
- Number of posts
- Status (rank)

The unregistered did not have the registered time, number of posts and status fields.

Extracting these data from each forum post was also made possible by the xpath[4] module provided by libxml2. XPath[5] is a language developed by World Wide Web Consortium(w3c)[6] to facilitate addressing of parts of an XML document.

The content of a post is under a div with message-body as its property as shown here:

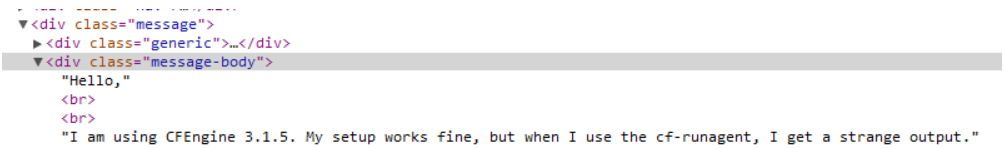

Figure 3.6: Using XPath queries inside a HTML doc such as this is relatively easy

To extract the entire message text, the following XPath query was used:

[4]*http://xmlsoft.org/html/libxml-xpath.html
[5]*http://www.w3.org/TR/xpath/
[6]*http://www.w3.org

```
//div[@class='message']//div[@class='message-body']
```

Thus extracted text from the posts was refined and made free from unwanted characters and hyperlink data.

Definition:
Topics are single words or multiple word phrases.

Extracting topics from the collected text required splitting the text into tokens of single words. The algorithm described in [24] was used to tokenize the words and extract natural language words from it.

The process is as follows:

1. Define a Probabilistic Model (Bayesian Model)
2. Enumerate the candidates
3. Select the most probable candidate

The idea is to assign numbers to each token that gives the probability of it being a valid word. The token with the highest value is a valid word.

3.3.3 Determining the topic of discussion

Many probabilistic models are being used for information extraction from a collection of text. Tf-Idf (Term Frequency Inverse document frequency) is one of such methods. It is normally represented as (adaptation from [25]):

$$\mathrm{tf_{i,j}} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

Figure 3.7: Term Frequency

$$\mathrm{idf_i} = \log \frac{|D|}{|\{j : t_i \in d_j\}|}$$

Figure 3.8: Inverse Document Frequency

$$(\mathrm{tf\text{-}idf})_{\mathrm{i,j}} = \mathrm{tf_{i,j}} \times \mathrm{idf_i}$$

Figure 3.9: Calculation of tfidf

This method is however very expensive as it has to search for the term in the entire document set for calculating the inverse document frequency.

One method, which might not be anywhere near accurate to the tf-idf method is to just calculate the frequency of the term in that particular post and leave out the inverse document frequency. When the most common words like

"'the"', "'a"', "'an"' etc are left out, the term having the highest frequency can be guessed as the topic of the post. This is not a very smart method, however, if we can deduce something from the topics already present in the Knowledge map and the term found in the Cfengine posts, this would save a lot of computations.

There is however need for more information in order to categorize the extracted topics into suitable topic types. For example: there is very little information for a topic to be related to cf-agent's errors or cfengine installation problems etc. The use of metadata and the ability to use it for analysis is shaping up the miscellaneous nature ot the Internet[13]. An organization's data is ordered to a great extent compared to the Internet's. Furthermore, processes can be imposed to structure the data in a desirable form.

The metadata currently available in the forums are not enough for careful analysis. Users must be allowed to create tags and there must be ways to flag posts as important, informative or unrelated. Under such circumstances, the following information can be extracted which show relevance of each entry and thus can be put into suitable types:

1. Popularity of a post - eg. total number of views, total replies
2. Importance of a post - eg. What kind of users(ranks) are participating in the discussion

3.3.4 Support Ticketing System data

The Ticketing System is a more formal way of providing IT support to the customers. There are several commercial and opensource ticketing systems available.

Customers send various types of support requests. Examining the support portal, common types of requests were:

- Request for license file.
- Request for pilot
- Request for documentation
- A policy doesn't behave as expected
- Errors shown by Cfengine
- New feature request

All of these support tickets have a certain severity and the type such as Cfengine Community Edition, Cfengine Nova etc. The OTRS stores the data in its own database and while browsing the through the topics, one has no idea about other related requests. The Ticketing system can be merged into Cfengine's Knowledge Management System. I believe the Promise Theory allows sufficient ways to model the support tickets into a policy file. The convergent property of a support need more discussion.

Cfengine can then store the support knowledge into its knowledge map directly. The existing reports can then be modified to show the statistics about the support tickets. eg. The support promises not kept, promise compliance and moreover, the business value report can also add the support quality information.

3.3.5 Bug Tracker data

A Bug tracker or a request tracker is a web application that is used to report bugs in a software or to request for new features. When a request is posted, it is analysed and assigned to a developer of a support personnel. The person then solves it and the request is closed if the fix is verified. Notes can be added to the issue later and it can again be reopened if the issue arises again.

The user interface of a typical bug tracker looks as below:

ID	Category	Severity	Reproducibility	Date Submitted	Last Update
0000545	[Cfengine 3 Community Edition] Other	major	always	2011-03-31 13:08	2011-04-08 12:17
Reporter	bas	**View Status**	public		
Assigned To	mark				
Priority	normal	**Resolution**	no change required		
Status	resolved			**Product Version**	
Summary	0000545: processes are always restarted in svn version 1949				

Description	This type of bundle always fails it used to work with versions prior or equal to 1917:
	{{{
	bundle agent ntp_daemons_check()
	{
	processes:
	debian::
	"ntpd"
	comment => "Check if ntpd is running",
	process_count => check_range("ntp", "1", "1"),
	process_select => sara_select_parent_process("1");
	methods:
	ntp_out_of_range::
	"any" usebundle => ntp_daemons_restart();				

Figure 3.10: A bugtracker showing details about the bug request [7]

As seen here, a bug request typically contains the following information:

- ID
- Category (Scope of the request)
- Severity of the issue
- Reporter
- Priority
- Status
- Summary and
- Description

These details can be chosen to be displayed or hidden according to the viewer's preferences.

Topics inherent to a particular context

Topics must generally be accompanied by a context. In the topic map language, a context is called a topic type. It allows us to understand the theme of a discussion or a story. Often times, the description and the following discussions contain code snippets as shown in the figure. These code snippets can be helpful in determining the topic of the discussion. In our case, if we parse the text in the description correctly, the classes and bundles inherent to Cfengine can be found and linked to the knowledge map.

The IT support personnel then can easily determine whether a problem reported by the customer is actually bug or not. Whether it has been solved or not and if it is solved, in which version is it solved? Software Engineers should make the practice of specifying in which revision number the bug was fixed so that it is easy to create release notes. Moreover, it is also easy for the IT support to provide patches to the customers related to particular bug fixes. They can also inform the customer when the bug was foxed or in which release it will be fixed.

3.3.6 Online Documentation data

Organizations generally maintain an online documentation of their products. These might include one or more of the following:

- Whitepapers
- Reference Manual
- Frequently Asked Questions
- Quick start guides and tutorials
- Troubleshooting guides
- Support tutorials and how to's

All of these documents can have some kind of pre-specified format. These documents can be organized in such a way that the topics can be easily added to the knowledge base. It involves more discipline and practice. These documents can be easily scanned and topics can be extracted from them.

In fact, CFEngine has already done this tasks. Topics in the Cfengine Reference manual[8] can be linked from the CFEngine Nova Knowledge map.

[8]http://www.cfengine.org/manuals/cf3-reference.html

3.4 A trouble free support desk

The support desk is always under pressure. Too much requests, too many complaints, too many customers and less resources. Knowledge Management leverage a lot of pressure off the support personnel.

With efficient knowledge management all data from the support portals can be collected and a unified view can be presented to the support personnel. Instead of wasting hours in finding a solution to a problem or solving the same problem again and again, the support personnel can use the knowledge map and find things easily. This can make the customer support very effective and efficient. After all customer satisfaction is the key to the growth of a business.

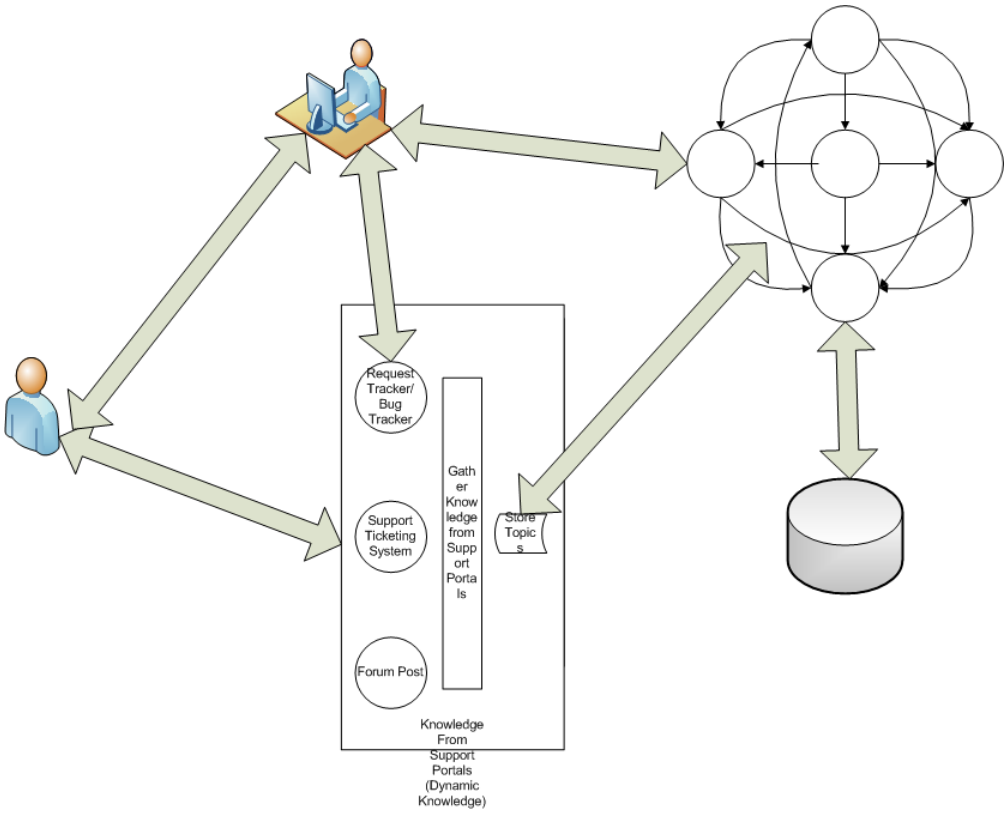

Figure 3.11: IT Support Interaction and Knowledge Sharing

Chapter 4

Results

4.1 Manual Knowledge Entry

The Cfengine Language provides an easy and intuitive way to gather and organize knowledge. Any type of knowledge could be represented by using Cfengine's Topic Map Model. The GUI provided by the commercial version Cfengine Nova was quite good to navigate through the Knowledge Map as shown below:

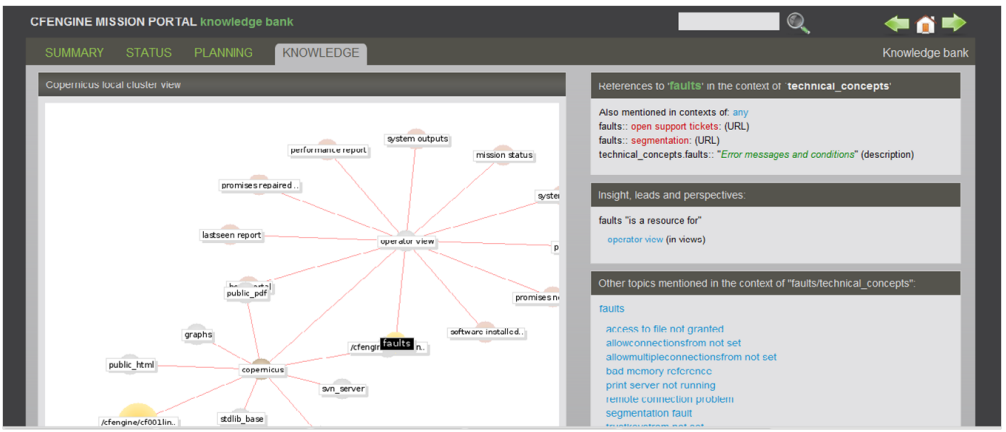

Figure 4.1: Cfengine Nova Knowledge Map showing faults as topics with links to the Ticketing system

4.2 Extraction of topics from the Cfengine Forum

Text extracted from the webpages contained some unwanted characters and punctuations. These needed to be stripped-off. The sequence of these chages are shown in the consecutive figures(from fig.. to fig..)

4.3 Writing Policies

The topics found in the webpages were fed to a program that created CFEngine policy files automatically. The result is as follows:

Inputs:

- Topic1 name(eg. forum$_corrupt)Topic2name(eg.cf-agenterrors)$

- Occurrences (URL) This URL was constructed from the forum id and the post id in case of the forum data.

eg. If forum id is 3,22050,

The URL is:

```
https://cfengine.com/forum/read.php?3,22050
```

Output:

The above inputs is translated to:

```
topics:

forum_topic::

'forum_corrupt'

('discusses','cf-agent errors','is discussed in'),

occurrences:

forum_corrupt::

"https://cfengine.com/forum/read.php?3,19172,19173#msg-19173"

represents => { "‘Discussion"'};
```

Looking for a little more info...

When there are updates available on your repo, and you've set up your /etc/yum.repos/*.repo files to point to the appropriate locations, couldn't you just run the command "/usr/bin/yum -y update" in a promise? - Any updates that are available will be downloaded and installed.

Or, am I missing something about what you need?

Kind Regards,
deb

There have been many, many tests, but no electrons were harmed.

Reply Quote

chadpatt
Re: Redhat Patching (update) using yum, rpm or yum via cfEngine.
May 06, 2011 06:11PM

Registered: 13 days ago
Posts: 6
Status: Contributor

No, I could do that... I have done that. I was just wondering if there is a way to use the internals of cfengine instead of using an external command... I guess its all the same?

Reply Quote

debheller
Re: Redhat Patching (update) using yum, rpm or yum via cfEngine.
May 06, 2011 07:14PM

Registered: 1 year ago
Posts: 29
Status: Regular

Yeah, understood. There is some package handling available, but not in the way you're looking for - which seems to be Linux-centric. Have you seen this?

[www.cfengine.org]

Kind Regards,
deb

There have been many, many tests, but no electrons were harmed.

Reply Quote

usernotfound
Re: Redhat Patching (update) using yum, rpm or yum via cfEngine.
May 07, 2011 12:26AM

Registered: 13 days ago
Posts: 3
Status: Newbie

Using the suggested "/bin/yum -y update" is actually executing LESS yum commands that the "cfengine internals", which I believe you mean:

packages: "$(slist)", package_policy => "add", package_method => yum;

That I believe will call at least two commands from yum and/or possibly rpm, and then have to make a comparison. Yum handles all this very well internally with one command :D

Figure 4.2: Forum post from which topics need to be extracted

After parsing the HTML and looking for relevant tags, the following information was extracted:

```
heading: Redhat Patching (update) using yum_rpm or yum via cfEngine.
Post Owner: chadpatt:3,318
user:
name=chadpatt
id=3,318
Registered: Yesterday
Posts: 5
Status: Contributor
*****************************************
date:  May 06, 2011 05:40PM
url:  https://cfengine.com/forum/read.php?3,21796,21796#msg-21796
Post Id:  3,21796
Message Id:  21796
text:  I
 have various installs with different packages installed and a dedicated
 repo for patching. I know how to setup a promise to update specific
packages... but how do you write a package to wildcard all installed
packages for patching?Sees like it would he something likepackages slist => { ".*" };
 but that doesn't seem to work... I have also triedpackages slist => { "a.*", "b.*" ..
  but no go with that as well...Any help would be appreciated.
=========================================
heading: Redhat Patching (update) using yum_rpm or yum via cfEngine.
Post Owner: chadpatt:3,318
user:
name=debheller
id=3,17
Registered: 1 year ago
Posts: 27
Status: Regular
*****************************************
date:  May 06, 2011 06:07PM
url:  https://cfengine.com/forum/read.php?3,21796,21801#msg-21801
Post Id:  3,21796
Message Id:  21801
text:  Looking for a little more info...When
 there are updates available on your repo, and you've set up your
/etc/yum.repos/*.repo files to point to the appropriate locations,
couldn't you just run the command "/usr/bin/yum -y update" in a promise?
 - Any updates that are available will be downloaded and installed.
 Or, am I missing something about what you need?Kind Regards,deb
=========================================
heading: Redhat Patching (update) using yum_rpm or yum via cfEngine.
Post Owner: chadpatt:3,318
user:
```

```
name=chadpatt
id=3,318
Registered: Yesterday
Posts: 5
Status: Contributor
*****************************************
date:  May 06, 2011 06:11PM
url:  https://cfengine.com/forum/read.php?3,21796,21802#msg-21802
Post Id:  3,21796
Message Id:  21802
text:  No,
 I could do that... I have done that. I was just wondering if there is a
 way to use the internals of cfengine instead of using an external
command... I guess its all the same?
```

After some clean up, the following topics were extracted:

```
update patching yum_rpm
cfengine yum wildcard slist ...
```

Chapter 5

Discussion

The approach presented would create a decent pool of topics that could be fitted into the knowledge map. This was an attempt to do what people normally think in abstract terms. Managing IT knowledge and unifying support portals is still a daunting process. It requires a lot of time, patience and discipline. This thesis was initiated with an expectation that it would arise interest of System Administrators and researchers in this field.

The model presented could generate topic maps that represented data from the support portals. However, the data extracted from the web portals were not sufficiently good enough to completely represent the IT support portals. This was due to limitations in the topic extraction algorithms. This algorithm needs to be more refined.

The task was, however made easy by the Cfengine Topic Map model. It presented a framework in which the high level modeling could be done. Attempting to program knowledge maps is a challenging task and modifying them to fit our needs is even harder.

5.0.1 Suggestions

1. More Metadata in the Internet forums Tim Berner Lee - the creator of the world wide web and currently working on his semantic web project - suggests the users to provide their raw data[1]. This might not be encouraging to commercial organizations. However, organizations that maintain discussion forums such as Cfengine's need to support the ability to provide a framework where users can add more metadata. Stackoverflow[2] is such a site which has come up with a very innovative idea. It maintains user reputations, and allows them to create tags after they have reached a certain level. This (generally)makes users more respon-

[1] TED Conference 2009 - www.ted.com
[2] www.stackoverflow.com

sible and nonsense can be avoided. The additions of the following facilities in the Internet forum might encourage more user participation in the forums and extraction of topics would be easier:

- Ability to create tags or topics saying what topics the discussion covers. The tags should be monitored by the IT personnel periodically so that they make good representations.
- Ability to flag posts as "'solved"' or anything similar or contrasting.

Chapter 6

Conclusion

IT Support in organizations are under a lot of pressure. Companies that start small and grow face a bottleneck in their it support because there is always a person who knows it all and make things work for the customers. The knowledge of individuals stay in their minds which if well documented can be a valuable asset to any organization.

Web-based support portals provide an efficient way of communicating with the customers. However, the knowledge available in these places are not available at one place. Our model presents a way to organize knowledge within an organization.

Cfengine provides a high level language that can be used to effectively represent knowledge. Being primarily used as a configuration management tool, it should not take up all the memory and bandwidth and CPU resources of the machine it is running on. Probabilistic approaches to knowledge mining from textual data are highly resource intensive. They require a lot of analysis and a unified model of knowledge management incorporating the IT support should not be resource intensive.

Chapter 7

Future Work

This research work was started with an expectation to initiate interest among system administrators and researchers to a section of an organization that plays a vital part for the proper functioning of the organization. There are still a lot of tasks that can be done or improved such as more intelligent algorithms for topic mining. Complex algorithms are left unexplored due to time limitations. Also, defining a formal process for Knowledge Management for IT support would greatly improve the services provided by any type of organizations.

Bibliography

[1] Mark Burgess. Knowledge management and promises. In *Proceedings of the 3rd International Conference on Autonomous Infrastructure, Management and Security: Scalability of Networks and Services*, AIMS '09, pages 95–107, Berlin, Heidelberg, 2009. Springer-Verlag.

[2] Mark Burgess and Alva L. Couch. Unifying configuration management and knowledge management (cfengine on topical islands). Technical report, Oslo University College, 2010.

[3] Anne Marie Kanstrup and Pernille Bertelsen. Participatory it-support. In *Proceedings of the ninth conference on Participatory design: Expanding boundaries in design - Volume 1*, PDC '06, pages 87–94, New York, NY, USA, 2006. ACM.

[4] Julie Smith David, David Schuff, and Robert St. Louis. Managing your total it cost of ownership. *Commun. ACM*, 45:101–106, January 2002.

[5] Noel Bruton. *How to Manage the IT Help Desk*. Oxford, Butterworth Heinimann, 2002.

[6] Office of Government Commerce. *ITIL v3 Service Operation*. Stationery Office Books (TSO), 2007.

[7] Gilbert. & Gibbert Michael. Leibold, Marius. Probst. *Strategic Management in the Knowledge Economy.* Publicis Corporate Publishing, 2 edition, 2005.

[8] CENWorkshop Agreement CWA 14924-1. European guide to good practice in knowledge management, part 1. Technical report, European Committee for Standardization, 2004.

[9] J. Hong and P. StÃ¥hle. The coevolution of knowledge and competence management. *International Journal Management Concepts and Philosophy*, pages 129–145, 2005.

[10] Nazrul Islam Varintorn Supyuenyong. Knowledge management architecture: Building blocks and their relationships. *School of Management, Asian Institute of Technology, Pathumthani 12120, Thailand*, 2006.

[11] Peter Tyndale. A taxonomy of knowledge management software tools: origins and applications. *Evaluation and Program Planning*, 25:183–190, 2002.

[12] Smiti Gandhi. Knowledge management and reference services. *Journal of Academic Librarianship*, 2004.

[13] David Weinberger. *Everything is Miscellaneous*. Times Books, 2007.

[14] Steve Pepper. The tao of topic maps. *www.ontopia.net*.

[15] Mark Burgess. A tiny overview of cfengine: Convergent maintenance agent. Technical report, Oslo University College.

[16] Mark Burgess Alva L. Couch. Human-understandable inference of causal relationships.

[17] Alva L. Couch and Mark Burgess. Troubleshooting with human-readable automated reasoning. In *Proceedings of the 24th international conference on Large installation system administration*, LISA'10, pages 1–, Berkeley, CA, USA, 2010. USENIX Association.

[18] Hao Wang, Bo Yang, Liang Liu, Qian Ma, Ke Wei Sun, and Ying Chen. Knowledge enhanced it service management. In *e-Business Engineering, 2007. ICEBE 2007. IEEE International Conference on*, pages 173–180, oct. 2007.

[19] Consortium For Service Innovation(CSI). Kcs practices guide v5, 2010.

[20] Roger Morse Peter Gilbert and Monica Lee. Enhancing it support with knowledge management. Technical report, CA Technologies, December 2007.

[21] Jeff Hammerbacher Toby Segaran. *Beautiful Data*. O'Reilly Media, 2009.

[22] Jane Yung-jen Hsu and Wen-tau Yih. Template-based information mining from html documents. In *Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence*, pages 256–262, 1997.

[23] T. Steyvers, M. & Griffiths. *Probabilistic Topic Models*. Latent Semantic Analysis: A Road to Meaning.

[24] Peter Norvig. *Natural Language Corpus Data*, chapter 14, pages 219–242. O'Reilly Media, 2009.

[25] Wikipedia. tf-idf.

Appendix A

HTML Parser and Tokenizer

```
int main(int argc, char **argv)

{ char *docname;
  htmlDocPtr doc;
  int count = 0;

if (argc <= 1)
{
  printf("Usage: %s docname\n", argv[0]);
  return(0);
}

docname = argv[1];
doc = getdoc(docname);

count = getPosts(doc,(xmlChar*)XPATH_MESSAGE);

dump_cooked_data(count);

xmlFreeDoc(doc);
xmlCleanupParser();
return (1);
}

/*************************************************/
xmlDocPtr getdoc (char *docname)
{
  htmlDocPtr doc;
  doc = htmlParseFile(docname, "utf-8");

  if (doc == NULL ) {
    fprintf(stderr,"Document not parsed successfully. \n");
```

```
    return NULL;
  }

  return doc;
}

/*************************************************/

xmlXPathObjectPtr getnodeset (xmlDocPtr doc, xmlChar *xpath)

{ xmlXPathContextPtr context;
  xmlXPathObjectPtr result;

context = xmlXPathNewContext(doc);
if (context == NULL)
{
  printf("Error in xmlXPathNewContext\n");
  return NULL;
}
result = xmlXPathEvalExpression(xpath, context);
xmlXPathFreeContext(context);
if (result == NULL)
{
  printf("Error in xmlXPathEvalExpression\n");
  return NULL;
}
if(xmlXPathNodeSetIsEmpty(result->nodesetval))
{
  xmlXPathFreeObject(result);
  printf("No result\n");
  return NULL;
 }
return result;
}

/*************************************************/

void print_nodeset_data(htmlDocPtr doc, xmlChar *xpath)
{
  xmlNodeSetPtr nodeset;
  xmlXPathObjectPtr result, userinfo;
  int i;
  xmlChar *keyword;
  xmlNodePtr cur;

printf("xpath: %s\n",xpath);
result = getnodeset (doc, xpath);
```

```
userinfo = getnodeset (doc, XPATH_USER_INFO);

if (result)
{
  nodeset = result->nodesetval;
   for (i=0; i < nodeset->nodeNr; i++)
    {
{
  keyword = xmlNodeListGetString(doc, nodeset->nodeTab[i]->xmlChildrenNode, 1);
  //printf("Message(%d)\n%s, name = %s\n",i, keyword,nodeset->nodeTab[i]->name);
  xmlFree(keyword);
}
    }


  xmlXPathFreeObject (result);
}
}
/***********************************************/

int getPosts(htmlDocPtr doc, xmlChar *xpath)

{ xmlNodeSetPtr nodeset;
 xmlXPathObjectPtr result, userinfo,messageUrl;
 xmlXPathObjectPtr messageDate,messageHeading,messageOwner;
 int i;
 xmlChar *keyword;
 xmlNodePtr cur;
 int count = 0;

THIS_AGENT_TYPE = cf_agent;

result = getnodeset (doc, xpath);
userinfo = getnodeset (doc, XPATH_USER_INFO);
messageUrl = getnodeset (doc, XPATH_MESSAGE_URL);
messageDate = getnodeset (doc, XPATH_MESSAGE_DATE);
messageHeading = getnodeset(doc,XPATH_MESSAGE_HEADING);

if(!(messageOwner = getnodeset(doc,XPATH_OWNER_WITH_DETAILS)))
  {
    messageOwner = getnodeset(doc,XPATH_OWNER_NO_DETAILS);
  }

if (result)
{
  nodeset = result->nodesetval;
  count = nodeset->nodeNr;
```

```
  for (i=0; i < count; i++)
    {
      getPost(doc,nodeset->nodeTab[i]->xmlChildrenNode,i);
      add_user_info(doc,userinfo->nodesetval->nodeTab[i]->xmlChildrenNode,i);
      add_message_date(doc,messageDate->nodesetval->nodeTab[i]->xmlChildrenNode,i);
      add_message_url(doc,messageUrl->nodesetval->nodeTab[i],i);
      add_owner(doc,messageOwner->nodesetval->nodeTab[0],i);
      add_heading(doc,messageHeading->nodesetval->nodeTab[0],i);

      // tokenize
      getTokens(i);
    }
  xmlXPathFreeObject(result);
  xmlXPathFreeObject(userinfo);
  xmlXPathFreeObject(messageUrl);
  xmlXPathFreeObject(messageDate);
  xmlXPathFreeObject(messageHeading);
  xmlXPathFreeObject(messageOwner);
 }
 return count;
}

/*************************************************/
void getTokens(int index)

{ char text[CF_BUFSIZE] = {0};
 char word[CF_MAXVARSIZE] = {0};
 int i=0,len=0,j=0;

 snprintf(text,sizeof(text),"%s",posts[index].text);
 len=strlen(text);
 for(i=0;i<len;i++)
   {
     switch(text[i])
       {
       case ' ':
       case '\n':
       case '\t':
       case ',':
       case ':':
       case ';':
       case '.':
 word[j]='\0';
         j=0;
   IdempPrependRScalar(&(posts[index].words), word, CF_SCALAR);
       break;
```

```
       //case '\0':

       default:
 word[j++]=text[i];
 break;

       }
   }

 word[j]='\0';
 IdempAppendRScalar(&(posts[index].words), word, CF_SCALAR);
}
/*************************************************/
void dump_cooked_data(int count)

{ int i;
  struct Rlist *rp;

printf("### count = %d ###\n", count);

for (i=0; i < count; i++)
  {
    printf("heading: %s\n", posts[i].post_heading);
    printf("Post Owner: %s:%s\n", posts[i].post_owner.name,posts[i].post_owner.id);
    printf("user: \n\tname=%s\n", posts[i].author.name);
    printf("\tid=%s\n", posts[i].author.id);
    printf("\tRegistered:%s\n",posts[i].author.registered);
    printf("\tPosts:%s\n",posts[i].author.posts);
    printf("\tStatus:%s\n",posts[i].author.status);
    printf("*****************************************\n");
    printf("date: \t%s\n", posts[i].date);
    printf("url: \t%s\n", posts[i].url);
    printf("Post Id: \t%s\n", posts[i].post_id);
    printf("Message Id: \t%s\n", posts[i].message_id);
    printf("text: \t%s\n", posts[i].text);
    printf("=========================================\n");
  }
}

/*************************************************/
static void add_user_info(xmlDoc * doc,xmlNode * a_node, int node_nr)

{ xmlNode *cur_node = NULL;
 xmlNode *child =NULL, *parent=NULL;
 xmlChar *text;
 char id[CF_SMALLBUF];
 char node_content[CF_MAXVARSIZE]={0};
```

```
 int type=0;
 char rtype,*retval;
 char value[CF_MAXVARSIZE]={0};

 for (cur_node = a_node; cur_node; cur_node = cur_node->next)
   {
     switch(cur_node->type)
       {

       case XML_TEXT_NODE:
 {
   if( FullTextMatch("^[ \t]*(.+):(.+)$",cur_node->content))
      {
if (GetVariable("match","2",(void *)&retval,&rtype) != cf_notype)
  {
    snprintf(value,CF_MAXVARSIZE,"%s",retval);
  }

if (GetVariable("match","1",(void *)&retval,&rtype) != cf_notype)
  {
    if(strcmp(retval,TEXT_USER_REGISTERED) == 0)
      {
snprintf(posts[node_nr].author.registered,CF_MAXVARSIZE,"%s",value);
      }
    else if(strcmp(retval,TEXT_USER_STATUS) == 0)
      {
snprintf(posts[node_nr].author.status,CF_MAXVARSIZE,"%s",value);
      }
    else if(strcmp(retval,TEXT_USER_POSTS) == 0)
      {
snprintf(posts[node_nr].author.posts,CF_MAXVARSIZE,"%s",value);
      }
  }
      }
 }
  break;
       }
   }
}

/*************************************************/

static void add_message_date(xmlDoc * doc,xmlNode * a_node, int node_nr)
{
  xmlNode *cur_node = NULL;
  char value[CF_MAXVARSIZE]={0};
```

```
  for (cur_node = a_node; cur_node; cur_node = cur_node->next)
    {
      switch(cur_node->type)
{
case XML_TEXT_NODE:
  snprintf(value, sizeof(value),"%s",cur_node->content);
  Chop(value);
  if( strlen(value) > 0)
      {
trimleft(value,sizeof(value));
snprintf(posts[node_nr].date,CF_MAXVARSIZE,"%s",value);
      }

  break;
}
    }
}
/*************************************************/
static void add_message_url(xmlDoc * doc,xmlNode * a_node, int node_nr)

{ xmlNode *child, *parent, *cur_node;
 xmlChar *text;
 char value[CF_MAXVARSIZE]={0};
 char *retval,*rtype;

 THIS_AGENT_TYPE = cf_agent;

 cur_node=a_node;
 child = cur_node->children;

 while(!(text = (char*)xmlNodeListGetString(doc,child, 1)))
   {
     child = child->xmlChildrenNode;
   }

 parent = child->parent;
 snprintf(posts[node_nr].url,CF_MAXVARSIZE,"%s",parent->properties->children->content)

 snprintf(value, sizeof(value),"%s",parent->properties->children->content);

 if( FullTextMatch("^(.+)\?(.+,.+),[0-9]+#msg-(.+)$",value))
   {
     if (GetVariable("match","3",(void *)&retval,&rtype) != cf_notype)
       {
 snprintf(posts[node_nr].message_id,CF_MAXVARSIZE,"%s",retval);
      }
```

```
     if (GetVariable("match","2",(void *)&retval,&rtype) != cf_notype)
       {
 snprintf(posts[node_nr].post_id,CF_SMALLBUF,"%s",retval);
       }
   }
 else
   {
     snprintf(posts[node_nr].post_id,CF_SMALLBUF,"-1");
     snprintf(posts[node_nr].message_id,CF_SMALLBUF,"-1");
   }
}

/*************************************************/
int ScheduleAgentOperations(struct Bundle *bp)
{}

/*************************************************/
static void getPost(xmlDoc * doc,xmlNode * a_node, int node_nr)
{
  xmlNode *cur_node = NULL;
  xmlNode *child =NULL, *parent=NULL;
  xmlChar *text;
  char id[CF_SMALLBUF];
  char node_content[CF_MAXVARSIZE]={0};

  for (cur_node = a_node; cur_node; cur_node = cur_node->next)
    {
      switch(cur_node->type)
{
case XML_TEXT_NODE:
  break;

case XML_ELEMENT_NODE:
  if(strcmp(cur_node->name,"div")==0)
    {
      if(xmlHasProp(cur_node,"class"))
{
  snprintf(node_content,sizeof(node_content),"%s",cur_node->properties->children->cont

  if(strcmp(node_content, TEXT_MESSAGE_BODY) == 0)
    {
      text = (char*) xmlNodeListGetString(doc,cur_node->xmlChildrenNode, 1);
      snprintf(posts[node_nr].text,CF_BUFSIZE,"%s",text);
    }
  else if(strcmp(node_content, TEXT_GENERIC) == 0)
  {
    getUserInfo(doc,cur_node,node_nr);
```

```
  }
  else if(strcmp(node_content, TEXT_USER_INFO) == 0)
  {
    // do nothing
  }
}
    }
  break;

}
    }
}
/************************************************/
int getUserInfo(xmlDoc * doc,xmlNode * node, int node_nr)

{ xmlNode *child, *parent, *cur_node;
  xmlChar *text;
  char node_content[CF_MAXVARSIZE]={0};

  cur_node=node;
  {
    child = cur_node->children;

    while(!(text = (char*)xmlNodeListGetString(doc,child, 1)))
      {
parent=child;
child = child->xmlChildrenNode;
      }

    snprintf(posts[node_nr].author.name,CF_MAXVARSIZE,"%s",text);

    snprintf(node_content,sizeof(node_content),"%s",parent->properties->children->cont

    if((strstr(node_content,"?")))
      {
snprintf(posts[node_nr].author.id,CF_SMALLBUF,"%s",(strstr(node_content,"?") + 1));
      }
    else
      {
snprintf(posts[node_nr].author.id,CF_SMALLBUF,"-1");
      }
  }

xmlFree(text);
return 1;
}
```

```
/************************************************/
void trimleft(char *buffer,int bufsize)

{
 char test[1000];
 char *tmp,*tmp1;

 tmp= strdup(buffer);
 tmp1=tmp;

 for(;*tmp!='\0';tmp++)
   {
     if((*tmp != ' ') && (*tmp != '\n'))
       {
 snprintf(buffer,bufsize,"%s",tmp);
 break;
       }
   }
 free(tmp1);
}

/************************************************/
static void add_heading(xmlDoc * doc,xmlNode * a_node, int node_nr)

{ xmlNode *cur_node = NULL, *child=NULL;
  char value[CF_MAXVARSIZE]={0};
  xmlChar *text;

  child=a_node->children;

  while(!(text = (char*)xmlNodeListGetString(doc,child, 1)))
    {
      child = child->xmlChildrenNode;
    }

  snprintf(posts[node_nr].post_heading,CF_MAXVARSIZE,"%s",text);
  xmlFree(text);
}
/************************************************/
static void add_owner(xmlDoc * doc,xmlNode * a_node, int node_nr)

{ xmlNode *cur_node = NULL, *child=NULL;
  char value[CF_MAXVARSIZE]={0};
  xmlChar *text;

  child=a_node->children;
```

```
  while(!(text = (char*)xmlNodeListGetString(doc,child, 1)))
    {
      child = child->xmlChildrenNode;
    }

  snprintf(posts[node_nr].post_owner.name,CF_MAXVARSIZE,"%s",text);

  snprintf(value,sizeof(value),"%s",child->parent->properties->children->content);

  if((strstr(value,"?")))
    {
      snprintf(posts[node_nr].post_owner.id,CF_MAXVARSIZE,"%s",(strstr(value,"?") + 1)
    }
  else
    {
      snprintf(posts[node_nr].post_owner.id,CF_SMALLBUF,"-1");
    }

  xmlFree(text);
}
/*************************************************/
```