

Knowledge Based Handling LINUX Utility: Help System

Jayakumar¹ Mohan kumar¹

PG Department of computer application, Sacred Heart College, Tirupattur, Vellore (dist), Tamilnadu¹

Abstract

Linux it's a wide availability and open architecture which provides a suitable and convenient for command user interface and towards research. We believe that the existing are in a formal usage, yet practical, there is need of knowledge based scheme for LINUX utilities towards building a higher-level online help and active help system. Hear with us propose a hierarchy of knowledge level to represent LINUX utilities: Static object, command syntax, command synopsis, command semantics and LINUX semantics. The ultimate goal of this paper is to realize a LINUX command user interface named "LINUX Knowledge based Utility(LKU)" that is knowledgeable enough to give intelligent advice to the active user.

Keywords: static object, command syntax, command synopsis, command semantics, Linux semantics

1. Introduction

LINUX provides a suitable and convenient within the human-computer interaction researchers can work on improving command line user interface. A user interacts with LINUX by typing commands through shell program known as the command line utility. The main function of the shell is to line feeding, text processing, dispatch or run, any of the hundreds of utility programs available in LINUX. The Linux-based system is modular Unix like operating system. Therefore the modern shell is add-on features. The drawback however, is that it is dreadfully difficult for the normal user to learn. The burden is made heavier by cryptic command names and other accidental undesirable features. Many design deficiencies point out years ago have been repaired. Understandably there are some attempts to cultivate the LINUX system for beginners and even some more ambitious attempts to provide online assistance for experienced users. These efforts take the following approaches.

2. GNU Bash

Bash is the shell, or command language interpreter, that will appear in the GNU operating system. It offers functional improvements over sh for both programming and interactive use[1]. While the GNU operating system provides other shells, including a version of csh, bash is the default shell. Like other GNU software, Bash is quite portable[2].

3. LKU Shell

LKU is the shell, or command language interpreter, for the Unix-like operating system. It is largely compatible with sh and incorporates useful features. It is simply a knowledge processor that executes commands. The term knowledge processor means functionality where text and symbols are expanded to create larger expression [3].

4. Online help system

These systems assist users through a question and answer format and make a suggestion & new ideas for future go on. The simplest of all is "man", a standard LINUX/UNIX command that displays the usage and description of any LINUX command. In this case, the question is always "how is command x used?". The answer printed to the question "man x or command name --help" is the manual pages for the command x and --help. It should be noted that the manual pages are intended to be read by experienced users and are not easily committed by novice user [4].

5. Active help system

Online help systems are of crucial importance for the computer systems of future. Passive help system (which require that the user requests help explicitly from the system) are of little use if the user does not know the existence of a system features. Active help system should guide and advice an user similar to a knowledgeable

assistant [5]. Most computer users feel that computer systems are unfriendly, not co-operative and that it take too much time and too much effort to get something done. They feel that they are dependent on specialists [6].

This paper investigates the apparent validity of consistent knowledge base if-then-rules for LINUX utilities Such as if-then-rules must satisfy the following goals.

- it is manageable in the present Linux/Unix environment.
- it does not limit itself to passive help system
- whenever possible, there is an easy conversation process from documentation to the knowledge base

6. Knowledge level

A knowledge base is a special kind of database for knowledge management. A knowledge base is an information repository that provides a means for information to be collected, organized, shared, searched, and utilized. It can be machine-readable or intended for human use [7]. Linux/Unix is written by programmers for programmers, the interface is so well structured that it is natural to describe it in terms of such hierarchical knowledge levels.

We have identified five levels of abstraction of Linux/Unix.

Level 1: Static Objects

At this level, operating system objects are declared and organized in a subsumption hierarchy[8]. Operating system objects include identifiers, files, filenames, user name and commands. They are arranged in a hierarchy according to their relationships in LINUX.

Level 2: Command Syntax

At this level, the command is always the first word. The option modify the function the command performs its function. The arguments identity the data upon which the command performs its function. Three additional areas you should know about using LINUX command are

- Abbreviations.
- Upper and lower case letter.
- Distinguishes between upper and lower case letter.

Each LINUX command is briefly described. This description includes the command name, valid option, i/o file types, and certain intrinsic properties, such as whether the command is destructive or not.

Eg.

Command [option]...args..

It shows the command currently entered by user. The knowledge base to find where action is needed and shows what need to change and give correct solution to the give problem by the help of rule based prolog language.

Level 3: Command synopsis

At this level, each LINUX command is defined in greater details than for level 2. The “synopsis” or outline, describing how the command is supposed to be used.

Eg.

Command [option]..[file]..

This knowledge level contributes directly toward this goal. LKU give the standard description of the command.

Level 4: Command semantics

At this level, The semantics of each LINUX command is defined. The command is described in terms of simple programming construct, such as loops and condition, and operation primitives, such as file operations(read, write, delete...) we intend that the knowledge engineer will be able to map existing source codes into this formalism.

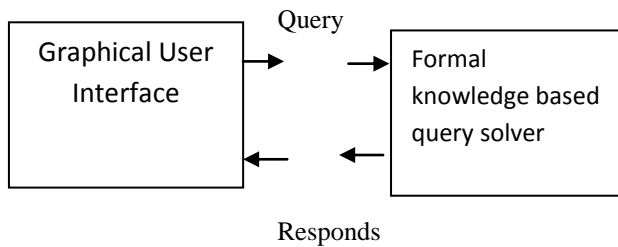
Level 5: .LINUX semantics

The semantics such as I/O redirection and the pipe mechanism are introduced. Until now, LINUX commands have been treated as standalone objects only.

7. Knowledge Base

A knowledge base so constructed repositories and displays an incremental nature of expertise which is defined as:

Fig 1:knowledge base



First, a thorough understand of static objects; second, a syntactic understanding of the utilities; Third, a semantic understanding of those utilities; and finally, the knowledge to put them together.

7.1. Taxonomic Classification

Since our knowledge base scheme is based on rules-based technology. Taxonomy varies slightly from source to source, but the core of the discipline remains: the conception, naming, and classification of organism groups [9].

7.2. Designing Knowledge object

Our model illustrates the central role of the command and its following options recognition process. We need the support of a programming concepts library and set of analysers with associated matching and decision rules. Among the component of the programming concepts library are a set of knowledge objects (KO) which consist of representation of often-used command concepts in command language independent from[10].

7.3. Control Dependence in the KO

Control dependence is related to conditional or alternative primes such as if-then-else rules. We describe control dependence in our KO as follows:

Control-dep (K,A,B,C)

Where

K: is a knowledge object

A and B sub commands in S

C: is a branch (True or False) of commands

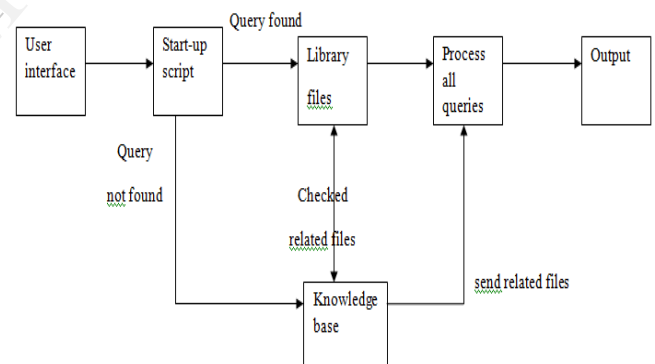
8. Natural Language Description for the Command Query

As command-Query proceeds through command line parameter recognition and fault localization tasks. It requires the use of command task and problem domain descriptions to communicate its finding to the user. These are embedded in each knowledge object as natural language descriptors that can be converted into dialogue for display by a user interface facility[11].

9. Using the Knowledge

We have introduced a knowledge base and representation for LINUX utilities in this section. There are many ways to make use of this knowledge in the context of the user support. This section presents an ideal LINUX shell that how the concepts can be applied to aid users.

The front-end shell is an ordinary LINUX shell, such as "tcsh, bash". The back-end is knowledge base which involves all query commands from shell program [12].



10. Conclusion

This paper began as an effort to create a knowledgeable mixed-initiative Linux command shell that is different from question answering consultant program or template-driven help system. We focus on how instructions activate the knowledge relevant to the performance of the task specified. We found that the very foundation of such a command shell demands a formal knowledge base and representation scheme for LINUX utilities. As for much knowledge base, we found it useful to arrange knowledge of Linux in a hierarchy of knowledge level.

11. References

- [1].<http://www.gnu.org/software/bash/>
- [2].http://www.gnu.org/software/bash/manual/bash-ref.html#What-is-Bash_003f
- [3].http://www.gnu.org/software/bash/manual/bash-ref.html#What-is-Bash_003f
- [4].Mathew Elison (independent consultant and trainer specialising in online help design and technologies).
- [5].Fischer, Gerhard, Lemke, Andreas, and Schwab, Thomas, "Knowledge-based Help Systems," Proceedings of ACM SIGCHI'85, ACM, 1985.
- [6].Gerhard fischer, andreas lemke and Thomas schwab "research group on Knowledge-based system and human-Computer Communication" web site:
link.springer.com/content/pdf/10.1007%2f3-540-13394-1_10.pdf#page-1
- [7].en.wikipedia.org/wiki/knowledge_base
- [8]. Woods, William A., "understanding subsumption and Taxnomoy: A Framework for Progress," in principles of semantic networks: Exploration in the Representation of Knowledge, 1991.
- [9].en.wikipedia.org/wiki/taxonomy
- [10]. F.detienne, "export programming knowledge: A Schema-based approach", Psychology of programming, Eds. J.Hoc, T.Greene, R. Samurcay, D. Gilmore, Academic Press, 1990, pp.205-222.
- [11]. Abdallah Tubaishat "A Knowledge Base for Program Debugging" IEEE, 2001.
- [12]. "Knowledge Representation and Semantics in a Complex Domain: The UNIX Natural Language Help System GOETHE" Herhard Heyer, Ralf Kese, Frank Oemig, Friedrich Dudda, pp.1990