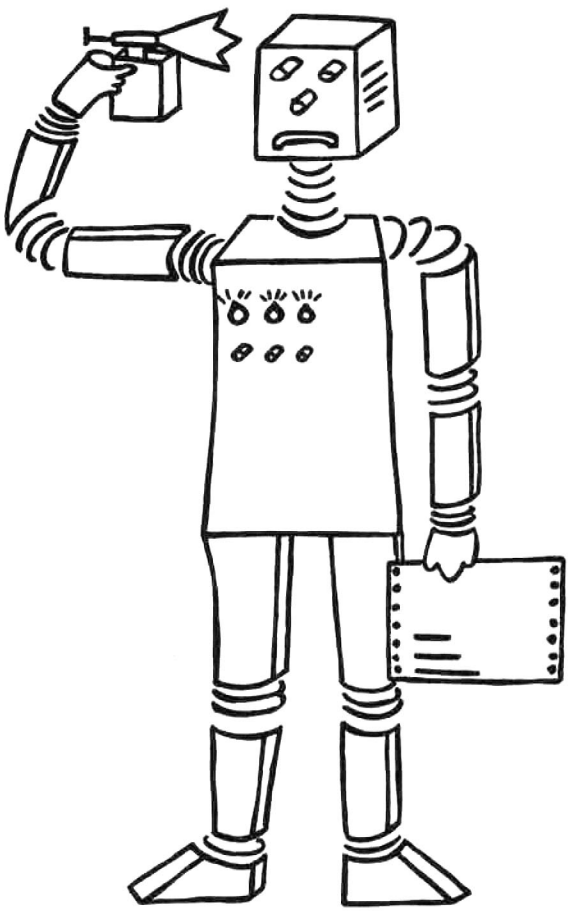
Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern 1, W. Germany

Artificial
Intelligence
Laboratories

SEKI-REPORT



μ-UNIXPERT: Diagnosis of Printer Problems

Michael Lessel
Harold Boley

SEKI-Report SR-87-09 August 1987

μ-UNIXPERT: DIAGNOSIS OF PRINTER PROBLEMS

Michael Lessel
Harold Boley

University of Kaiserslautern
W.Germany

Abstract

The μ-UNIXPERT systems perform knowledge-based diagnosis in the domain of printers as part of the computer periphery. They are implemented in LISPLOG, a functional/logical language that provided the required programming spectrum ranging from operating-system calls to the explanation component.

Acknowledgements

We would like to thank the members of the LISPLOG Group, the μ-UNIXPERT.2 Laboratory Course, and especially Jürgen Herr, Ansgar Bernardi and Franz Kammermeier.
Gebhard Przyrembel volunteered as our UNIX™ expert.

Introduction

μ-UNIXPERT (Micro UNIX-PERiphery Tester) [Lessel 86] is a knowledge-based diagnosis system for the treatment of printer hardware faults and of other problems which can appear in connection with printing (user errors and software faults).

The system both generates diagnoses and gives proposals for clearing. To reach this goal, μ-UNIXPERT first chooses a diagnosis

domain, and then establishes and tries to prove a concrete hypothesis. If the proof fails, the system tries to generate another hypothesis. The main difficulty occurring in this diagnostic task is to reconstruct the situation when the printing order was given.

The entire knowledge of μ-UNIXPERT consists of a set of PROLOG-like facts and rules. The system is implemented in LISPLOG [Boley 86], a LISP/PROLOG integration running in FRANZ LISP under the UNIX 4.2 BSD operating system on a Vax 11/750.

After a short introduction into the problem domain, we will present two versions of the system, our basic μ-UNIXPERT.1, and our refined μ-UNIXPERT.2, employing automatized knowledge acquisition.

The Problem Domain

There is a wide range of reasons for problems appearing in connection with printer jobs, covering hardware and software faults as well as faults falling into the user's responsibility. Most users lack an overview of the whole (operating plus computer) system and therefore run into difficulties if they have to analyze the reasons for erroneous behavior, and to induce or carry through appropriate countermeasures.

An explicit modelling of processes running on the computer is exceedingly complex, even if it is done on a high level of abstraction, especially because we have no suitable software specifications. There is hardly any possibility at all to formally understand the behavior of processes on a lower level of abstraction, e.g. the level of the hardware. Therefore we have deliberately used informal problem-solving strategies of an expert who has a good overview of the computer and the processes running on it.

Now let us consider a short selection of problems that may appear when a UNIX printing command is issued.

- User faults
  - Often there are several printers connected with the computer. Thus it may happen that an output will arrive at a printer different from the one the user wanted to choose, because there is an unexpected default printer or a wrong preadjustment in the .login-file.
  - The user may confuse similar print options. This can for example

cause the formatting of a file, while the user wanted to send it to a certain printer.
- If the user omits the file name, the text will be expected from the teletype.

- Software faults
  If the SPOOLER DAEMON is not present, it is impossible to print. Furthermore there are no countermeasures possible for an ordinary user. A higher-priority system (run by the superuser) would be able to solve this problem in some cases.

- Hardware faults
  We will summarize problems like faulty printer connections, disordered multiplexers, internal hardware faults, or faulty printer adjustments as hardware faults.

This is only a small collection taken from the variety of possible faults. Still, we regard them as sufficiently representative to demonstrate the practicability of our μ-UNIXPERT approach.

μ-UNIXPERT.1

In the present section we want to give a short overview of the basic version's mode of operation. We found it to be advantageous to perform a prediagnosis, in order to constrain the domain of possible diagnoses and thus reduce the complexity of the entire diagnostic task. For that purpose we can partition the set of all diagnoses into several domains. Starting with the acquired data, we will cast suspicion on a certain domain of diagnosis. Thereupon the generated hypotheses will be checked successively by backward reasoning, until a plausible diagnosis is found.

There is a high degree of uncertainty on the level of prediagnosis (because we want to generate a suspicion as fast as possible). Hence there must exist a possibility to do a new prediagnosis if the suspicion first generated later proved to be wrong. If it is possible to find a diagnosis, the program will inform the user and, if appropriate, suggest remedial measures.

First of all the program will ask some general questions. The

knowledge derived from the answers is used as foundation of the preselection of a domain of diagnosis mentioned above. Now the program needs knowledge to check the validity of a suspicion, which will be gained mainly by using operating-system facilities. In some cases the printer waiting queue must be examined (Is the print job waiting in the queue? Was the print job put into the queue? et cetera); in other cases a file must be searched for certain details (for example, look into the .login file if there is a printer preadjustment).

Normally a long period elapsed from the issuing of the printing command until the time of diagnosis. Therefore the system must try to reconstruct the state of the computer at the time when the printing command was given. In the basic version, the acquisition of the required knowledge is done primarily by interaction with the user.

One goal, then, is to minimize the number of questions the user will be asked. This can be achieved, because the program can do most tests and analyses automatically (μ-UNIXPERT.2).

μ-UNIXPERT.2

The development of the advanced version μ-UNIXPERT.2 was based upon the knowledge base of the basic version. Here, we only introduce the numerous extensions to μ-UNIXPERT.1, because of the similarity of the mode of operation in both versions. Above all, these extensions contribute to the user acceptance of the system.

While the first version asked the user a lot of questions, the second version does all calls to functions of the operating-system and analyses of their results by itself. The basic version statically questions the user about the information which is required for the prediagnosis (menu technique). The second version allows the user to describe his problem. The system derives the required knowledge from this description. Of course, μ-UNIXPERT.2 is not a natural-language comprehending system in the usual sense. Nevertheless the system is capable to understand, interpret, and transform into an internal representation most user inputs concerning the problem domain "printer diagnosis".

If the system lacks information to do the prediagnosis, this will

either be deduced or, should this be impossible, the user will be asked. If there is uncertain or incomplete information, the generated suspicion will be uncertain, too. For simplicity's sake we tried to have the knack of uncertainty by means of certainty factors [Frost 86], though the weakness of this method is obvious.

In the second phase of diagnosis the suggested suspicion will be checked. Here is another source of uncertainty, if the user does not know the answer to a question. Therefore we have implemented a logic with three truth values (true, false, unknown).

While μ-UNIXPERT.1 has only a restricted component for the generation of explanations (capability to list the information the system has either deduced or asked for from the user, and to trace the program by means of the debugger), this component is essentially more comfortable in μ-UNIXPERT.2. It is possible to call this explanation component whenever the system asks a question, after the prediagnosis, and after the ultimate diagnosis. When it has done the prediagnosis, the system explains how a certain suspicion was generated. Apart from the classical HOW and WHY questions, it is possible to ask which diagnoses were excluded up to now, and how this exclusion was derived.

Why LISPLOG

It is important for the implementation of a system to choose a method of description adequate to the problem's level of abstraction. Therefore we must exact some requirements from a programming language which shall be appropriate for the implementation of a knowledge-based system.

The programming language should enable a natural representation of the available knowledge and the separation of knowledge and processing method. Furthermore it is important that the language contain a built-in deduction mechanism to draw inferences from the knowledge base. μ-UNIXPERT depends on the facility to call operating-system functions and analyze their results. The programming language should support this, too, in a simple manner. LISPLOG fulfills all these requirements and proved to be well suited as an implementation language in our domain.

Another interesting point is the integration of logic and functional programming. While the PROLOG part of LISPLOG predominated in

μ-UNIXPERT.1 and LISP was used for I/O and calls to operating-system functions only, we have achieved a higher degree of integration of functional and logic programming in μ-UNIXPERT.2.

Sample Dialogues

In the following we will compare two sample dialogues dealing with the same problem to show the superior comfort of the second version. It is evident in the second dialogue that the system takes very much work upon itself (e.g. examination of the printer waiting queue and acquisition of the required information).

1st Dialogue (μ-UNIXPERT.1)

In this example the user was trying to print out the file "passau", but the printer is in mode OFFLINE, so that printing is impossible. In the following, English comments are enclosed in (* *).

```
% Unixpert     (* Calling micro-Unixpert.1 *)

Lisplog Lader Version 1.00 12.12.86

LISPLOG System geladen.
micro-UNIXPERT geladen


*(unixpert)

       ******************************
          micro-UNIXPERT gestartet
       ******************************

Welches Problem haben Sie? (0/1)
(0) Der Druck ist nicht gekommen.
(1) Sie sind mit dem Druck nicht zufrieden.
> 0

Wo sollte der Druck hin? (B48/B14)
> B14
(* B14 is the correct building *)

Geben Sie bitte den Wortlaut des Befehls ein.
> lpr -Plpb14 passau
```

```
(* Printing command with option -Plpb14, i.e. *)
(* target printer is in building 14.          *)
(* If this information is omitted, then the   *)
(* preadjustment of the printer variable in   *)
(* the .login file is checked.                *)

Ist das Prompt-Zeichen erschienen nachdem Sie die
Return-Taste gedrueckt  haben ? (ja/nein)
> ja

(* The preselection is finished now.     *)
(* The system has generated a suspicion, *)
(* which will influence its future       *)
(* questioning behaviour.                *)



---------------------------------------------
lpb14 is ready and printing
Rank   Owner      Job  Files        Total Size
active lessel     900  passau       20110 bytes
---------------------------------------------

(* The program causes the output of the  *)
(* printer waiting queue. In an extended *)
(* version, the program itself shall     *)
(* analyze the waiting queue.            *)

Bitte sehen Sie in der Warteschlange nach, ob es
ein Warning gab  (ja/nein).

> nein

Ist Ihr Auftrag in der Warteschlange?  (ja/nein)
> ja

Befindet sich Ihr Druckauftrag an der obersten
Stelle der Warteschlange ?  (ja/nein)

> ja


---------------------------------------------
lpb14 is ready and printing
Rank   Owner      Job  Files        Total Size
active lessel     900  passau       20110 bytes
---------------------------------------------
Ist Ihr Job mittlerweile aus der Warteschlange
verschwunden?  (ja/nein)
> nein

Ueberpruefen Sie bitte, ob der Drucker auf ONLINE steht.

Soll eine neue Diagnose versucht werden? (ja/nein)
> nein

      micro-UNIXPERT normal beendet
```

2nd Dialogue (μ-UNIXPERT.2)

The problem is the same as in the first dialogue. The length of the dialogue results from the extensive use of the explanation facility.

```
2: lpr -Plp48211 passau
3: Unixpert

Sie hatten Probleme beim Absetzen
eines UNIX-Druckauftrages?

Bitte beschreiben Sie Ihr Problem:

> druck nicht gekommen

(* The system has comprehended the pidgin- *)
(* German phrase and paraphrases it.       *)

Der Druck ist nicht angekommen

Ist das ok ?  > ja

Nun beginnt die Vordiagnose.

Auf welchem Drucker sollte der Druck laufen ?

> Bau48211

Die Vordiagnose ist beendet ! Noch Fragen ?

>  help

Die Vordiagnose ist  nun  beendet: Jeder der folgenden
vier Diagnosebereiche wurde mit einem Certainty-Factor
zwischen 1 und -1 belegt:

funktionsstoerung-drucker
fehler-file
fehler-druckbefehl
falscher-drucker

Sie koennen Sie sich alle,
nur die wahrscheinlichen (positiven)
nur die unwahrscheinlichen (negativen)
oder eine bestimmte Anzahl von Diagnosebereichen (1 - 4)
ausgeben lassen

>  pos

Die positiven Bereiche sind:

funktionsstoerung-drucker

keine weiteren positiven Bereiche.

>  funktionsstoerung-drucker

Diagnosebereich wurde gewaehlt, weil folgende Praemissen
gelten:

richtiger-drucker    1
fehler-druckbefehl    -1
```

```
Es gibt keine weiteren Praemissen

>  richtiger-drucker

Regel Nr. 1 hat gefeuert
Es gelten folgende Praemissen:

(wohindruck B48211)   1
(option B48211)   1

keine weiteren Praemissen

Aufgrund dieser Praemissen wurden folgende Hypothesen
erschlossen:

richtiger-drucker   1

keine weiteren Hypothesen


>  weiter

  Beginn der Hauptdiagnose.
Aufruf Systemfunktion : (targetprinter)
Aufruf Systemfunktion : (pq-change)


Steht der Drucker B48211 auf ONLINE ?  > help

Sie koennen hier folgendes eingeben:
ja
nein
weiss nicht
warum   (dann erklaere ich Ihnen,warum
         ich diese Frage stelle)

Steht der Drucker B48211 auf ONLINE ?   > warum

   Mit Strategie 2 will ich beweisen:

      auf-OFFLINE(_drucker)

Ich habe dafuer schon gezeigt:

   (1)   _drucker is #(targetprinter)
   (2)   ask(druckgekommen)
   (3)   false(druckgekommen)

Ich muss noch zeigen:
   (1)   ask(Online)
   (2)   false(Online)


 Tippen Sie :   0 fuer 'Warum ?'
                n fuer 'Wie wurde n-te Praemisse bewiesen ?'
                <Praed> fuer 'Wie wurde <Praed> bewiesen ?'
                z fuer 'zurueck zur letzten Frage'
                q fuer quit

              > 3

***** Vordiagnose *****
```

```
 Tippen Sie :   0 fuer 'warum ?' (nur sinnvoll waehrend Diagnose !)
                n fuer 'Wie wurde n-te Praemisse bewiesen ?'
                <Praed> fuer 'Wie wurde <Praed> bewiesen ?'
                z fuer 'zurueck zur letzten Frage'
                q fuer quit

                > q

Steht der Drucker B48211 auf ONLINE ?   > ausgeschlossen

    Von den folgenden Fehlern weiss ich bereits,
    dass sie nicht gelten:

    (1)   papier-alle(B48211)
    (2)   papierstau(B48211)
    (3)   ausgeschaltet(B48211)
    (4)   papier-falsch-eingelegt(B48211)
    (5)   farbband-leer(B48211)
    (6)   traktor-defekt(B48211)
    (7)   stecker-draussen(B48211)

                Nummer (0 = zurueck)  ----> 3

  Mit Strategie 1  habe ich bewiesen:

        false(ausgeschaltet(B48211))

         Praemissen :

   (1)   B48211 is #(targetprinter)
   (2)   #(file-in-queue)

(* The system will look through the printer waiting queue. *)
(* Thus the user need not care about that. *)

Tippen Sie : 0 fuer 'warum ?' (nur sinnvoll waehrend Diagnose !)
             n fuer 'Wie wurde n-te Praemisse bewiesen ?'
             <Praed> fuer 'Wie wurde <Praed> bewiesen ?'
             z fuer 'zurueck zur letzten Frage'
             q fuer quit

               > q

Steht der Drucker B48211 auf ONLINE ?   > nein
 ================================================

      Schalten Sie den Drucker auf ONLINE !
      Wenn der Druck dann nicht kommen sollte

      liegt es am Spooler Daemon.
 ------------------------------------------------

                  Ist der Fehler behoben ?
          ----> j

 ================================================
     Ich freue mich, dass ich Ihnen helfen konnte.
 ================================================

               Moechten Sie eine weitere Anfrage an
               UNIXPERT2 stellen?
                   --->  nein

 Auf Wiedersehen!
```

References

[Boley 86] : H.Boley (Editor): A bird's-eye view of LISPLOG. The LISP/PROLOG integration with initial-cut tools, Universität Kaiserslautern, SWP-86-08

[Lessel 86] : M.Lessel : micro-UNIXPERT: Ein wissensbasiertes System zur Behandlung von Problemen bei UNIX-Druckauftraegen. Universität Kaiserlautern, SWP-86-04

[Frost 86] : R.A.Frost : Introduction to Knowledge Base Systems Collins, 1986.