

EXPERT SYSTEM DESIGN IN REXX

MARC VINCENT IRVIN
NORDEN SYSTEMS

EXPERT SYSTEM DESIGN IN REXX

By Marc Vincent Irvin

Expert System Design In REXX

Introduction

98

Because REXX has user friendly syntax, is lightly typed, and handles symbols well it was an ideal medium for a Knowledge Engineering systems experiment named REXRULES.

- WHAT ARE EXPERT SYSTEMS
 - history
 - players
 - structure
 - paradigms
 - justifications
 - applications
- REXRULES: ES INFERENCE IN REXX
 - assets
 - facts
 - rules
 - chainings
 - attributes
 - pros/cons
- ES/REXX PROMISE

(C) MVI Software 05/91

Expert System Design In REXX

WHAT ARE EXPERT SYSTEMS?

- HISTORY
from games to chemistry to medicine to value seen
 - PLAYERS
hosts, tools, shells, experts, engineers, & users
 - STRUCTURE
knowledge, inference engines, and heuristics
 - PARADIGMS
Emycin, Prolog, OPS5, TK-Solver, and Expert Choice
 - QUALITIES
high ROIs by saving, promoting, and enforcing work
 - APPLICATIONS
DASD, Software, JOB, Network, and Help Desk management
- (C)Copyright 05/91 By Marc Vincent Irvin

What Are Expert Svstems?

DEFINITIVE DEFINITION

- An EXPERT SYSTEM is one that assists facillitates, or replaces expert(s).
- ASSISTS - When the expert calls on the system to improve his/her performance.
- FACILLITATES - When the expert's skills improve the performance of non-experts.
- REPLACES - When expert(s) develop expertise exceeding human capabilities.

What Are Expert Systems?

HISTORY

- PARLOR GAMES
Turing asked, can it pass a line up?
Parlor game provided measuring rod.
Eliza was therapist, PARRY was patient.
- NASA's ROBOT Chemist
DENDRAL uncoded molecules.
DENDRAL out did the experts.
- Bacterial Infection Diagnosis
MYCIN gave RXs and explained reasoning.
MYCIN led to EMPTY MYCIN or EMYCIN.
- NEED SEEN: Future Shock's Answer
MIT, computers didn't up productivity.
10-15 yrs to get expert in something.
Experts decode & sift \$K into solutions.
ES are industrial complexity pills.
- BEST AI SOLUTION AMONG MANY
NL, Robotics, Neural Nets, & Fuzzy Logic.

(C)Copyright 05/91 by Marc Vincent Irvin

What Are Expert Systems?

PLAYERS

- Construction Materials
computers, languages, shells, and boxes.
- Architect (Cogpsych/CS/AI)
Knowledge Engineers put it together.
Trained in extraction & compaction.
- Domain Expert
Provides subjective functional analysis.
KA blues, busy silent vague and distant.
- ES Users
They get advised, directed, or corrected.
Potential next generation of experts.
Complex work with little or no training.
- ES Support
Keep facts and rules up to date.
Done by any of the above, or others.

(C)Copyright 05/91 by Marc Vincent Irvin

What Are Expert Systems

STRUCTURE

- **KNOWLEDGE**
 - Facts
 - Rules
 - Frames
 - Attributes
- **INFERENCE ENGINE**
 - Interpreter
 - Scheduler
 - Reporter
- **HEURISTICS**
 - Expert observation
 - Metabase experience

(C)Copyright 05/91 By Marc Vincent Irvin

What Are Expert Systems

STRUCTURE: Knowledge

- **FACTS** ie. Can is bent, top is not off.
data element or multivalued variables
monotonic, nonmonotonic, or uncertain
discrete or inheritable via attributes
 - **RULES** ie. If can is bent then can is b
has rulename for infer, fuzz, & reporting
name: LHS/antecedent RHS/consequen
can use old facts to make new facts
 - **FRAMES** (objects attributes values)
CONCEPT: part
SLOT1: name
SLOT2: condition
-
- INSTANCE: part
NAME: can
CONDITION: bent
PROCEDURE: part_fixer

(C)Copyright 05/91 By Marc Vincent Irvin

What Are Expert Systems

STRUCTURE: Inference Engine

- Inference Engines contain strategies and controls that KE use to manipulate facts and rules. Its 3 main functions are to interpret, schedule, and explain facts and rules to and for its users.
- INTERPRETER
Maps attributes against facts and rules.
Sets and stores processing options.
Does var/memory inits and runs profiles.
Does syntax checking and writes errors.
- SCHEDULER
Seeks goals, fires rules, finds unknowns.
Sets firing priorities and tracks steps.
Interfaces procedurally to environment.
Does message sending and retrieval.
- EXPLAINER
Tells in english how answers were made.

(C)Copyright 05/91 By Marc Vincent Irvin

What Are Expert Systems

STRUCTURE: Attributes

- ATTRIBUTES - Used for query, process controls, and english status reporting.
- USER QUERY
Usually, if knowbase has no answer it will ask user for answer using A) a KE made text or B) an invented text.
- PROCESS CONTROLS
Things like value and range checking, logic tracking, fuzzy or confidence factors, and defaults are often found.
- ENGLISH STATUS REPORTING
Sometimes, IE will provide english like responses about outstanding RULES and set values. They may be KE supplied, but better IEs will give current reasoning and value settings.

(C)Copyright 05/91 By Marc Vincent Irvin

What Are Expert Systems

STRUCTURE: Chainings

- Demo Ruleset for chaining
GOAL = END -OR- RETE = ON
R1: IF A = J & C = 2 THEN END = 1
R2: IF D = M & C = 1 THEN C = 2
R3: IF C = 2 THEN A = J
R4: IF D = M THEN C = 1
R5: IF C = 1 THEN D = M

103

- forward chaining FIRE sequences
R1 R2 R3 R4 (R5)
R1 R2 R3 (R4) R5
R1 (R2) R3 -R4- -R5-
R1 R2 (R3) -R4- -R5-
(R1) R2 -R3- -R4- -R5-
- backward chaining FIRE sequence
R1 R3 R2 (R5) R2 (R4) (R2) (R3)
(R1)END

(C)Copyright 05/91 By Marc Vincent Irvin

What Are Expert Systems

PARADIGMS

- M.1
Mycin backward chained
certainty factors
- PROLOG
declare/proc Logic base
clause driven
- OPS5
RETE forward chained
data driven popular in MIS
- TK-SOLVER
ESS w/o cells
- EXPERT CHOICE
DSS w/o PHD

(C)Copyright 05/91 By Marc Vincent Irvin

What Are Expert Systems

QUALITIES

- 1000 pct return on investment
Northrop, ESP - 14 to 4.3 hrs
DEC nets \$40M/yr on XCON
- VALUABLE SKILLS RETAINED
retirees knowledge coded
career changes painless
- USEFUL SKILLS PROMOTED
Easy, 24 hr, error free access
Automatic history of reasoning
- STANDARDS ENFORCEMENT
critical tasks
routine tasks
- COMPLEXITY MANAGEMENT
information overload
overboard technology

(C)Copyright 05/91 By Marc Vincent Irvin

What Are Expert Systems

MIS APPLICATIONS

- DASD DEFINITIONS
performance/security enforcing
parameter optimizing
- SOFTWARE MAINTENANCE
advice and error recovery
- JOB MANAGEMENT
scheduling and prioritizing
- NETWORK SUPPORT
EP/VTAM/NETVIEW line servers
DASD and SPOOL monitoring
- HELP DESK
expert directory & basic fixups
problem logs and tracks
- AUTOMATED OPERATIONS
Operators/Tech-support replaced
Light's Out has become common

(C)Copyright 05/91 By Marc Vincent Irvin

Expert System Design In REXX

REXRULES: Expert Systems In REXX

- **ASSETS**
untyped, pseudo code, symbolic, and portable
 - **FACTS**
literals, variables, stored, filed, & iterative
 - **RULES**
value, clause, data, formula, clock, & self driven
 - **CHAININGS**
backward, forward, mixed, and custom(depth/breadth)
 - **ATTRIBUTES**
basic, truth, dynamic, imprecise, and fuzzy values
 - **PROS/CONS**
learning, I/O, math, connects, and procedural
- (C)Copyright 05/91 By Marc Vincent Irvin

Expert Systems In REXX

REXX ASSETS

- **UNTYPED MEANS:**
no time defining fields
unlimited mixing
SAY 'Value of' X 'is' N*3.'
- **PSEUDO CODE MEANS:**
coders need little training
code can be self documenting
IF A = B THEN SAY 'A=B'
- **SYMBOLIC MEANS:**
words & phrases paramount
unknown symbols can be found
IF SUNNY & WARM THEN 'SWIM'
- **PORTABLE MEANS:**
PC code can run on VM & MVS
one day REXX IBM on REXX DEC
- **CONNECTED MEANS:**
excellent host interfaces?
(C)Copyright 05/91 By Marc Vincent Irvin

Expert Systems In REXX

REXRULES FACTS

- LITERAL
IF IDEA = 'GOOD' THEN JOKE = 'FUNNY'
INITGOAL: IDEA = GOOD; TIME = LATE
- ASSIGNED (FALSE = 0)
IF DAY OR SUNSET THEN NIGHT = FALSE
rules/tasks can set true/false values
popular in diagnostic systems
- SYMBOLIC
PUT(Tim sees Ann); PUT(Ted sees Sue not)
UNKNOWN = 'WHO'; GET(Tim sees who)
multiple UNKNOWN solutions sep'd by space
advanced pattern match, GET(RANGE'>')
- FILE BASED
X = READ(filename,seq,[key]) for basic files
RC = WRITE(filename,seq,data) to write recs
OPS(MAKE,"','UID1' DAY TIME)
IF OPS("','UID1 = "UID1") THEN 'IDFND'
- ITERATIVE - experimental='Tim Ted Sue Ann'
One pass done for each name.

(C)Copyright 05/91 By Marc Vincent Irvin

Expert Systems In REXX

REXRULES RULES

- VALUE DRIVEN
unset values Q'd for OOS resolution
FORM(s) => RULE(S) => ASK.X
Examples...
RULE_8: IF SPEED = LOW AND,
PRT_QUALITY = HIGH THEN DO
PRT_SET='MODL-1'; RUN(SAY_MODL); END
RULE_2: IF ABBREV('YES',NEEDFAST,1) AND,
PRTPAGES > 5 THEN SPEED = HIGH
- CLAUSE DRIVEN
unfound patterns/GET() Q'd for solution
RULE(s) => ASK.
Examples... unknowns = 'sport'
put(hockey has contact soccer has contact,3)
put(hockey is fast hockey played_with puck,3)
likes_sport: if nop(name) & get(sport has action),
& get(sport played_with puck),
then put(name likes_sport sport)
has_action: if get(sport has contact) &
get(sport is fast),
then put(sport has action)

(C)Copyright 05/91 By Marc Vincent Irvin

Expert Systems In REXX

REXRULES RULES

- DATA DRIVEN
matched objects/OPS() direct reasoning
Examples...
OPS('FIL NAM A',QUEST,'NAME TEXT')
OPS(MAKE,QUEST,'QUES1 WHAT IS TEMP?')
OPS_RULE:
IF OPS(QUEST,'NAME = "QUES1"') THEN,
 OPS(SET,QUEST); SAY NAME TEXT
 OPS(REMOVE,QUEST)
END
- FORMULA DRIVEN
Unset values in formulas/EQU. get solved
Examples...
MAINGOAL ▪ 'SHOTS'
EQU.BACBOOZ = (150/WGHT)*(PCT/50)*SHOTS,
 * .025
EQU.PCT ▪ PROOF/ 2
ASK.WGHT ▪ 'WHAT DO YOU WEIGH' NAME?'
* Implied via rules of algebra...
EQU.PROOF ▪ PCT* 2
EQU.SHOTS = BACBOOZ/(150/WGHT*PCT/50*.025)

Expert Systems In REXX

REXRULES RULES

- CLOCK DRIVEN
rules/forms/tasks on dateltimeelapsed
Examples...
/* in 30 minutes run CK_LINES once */
CLK.CK_LINES = 'M30'
/* on 3/3 run every 2min from 12 to 6*/
CLK.SUBMIT = '91/03/03 12:00 0.M2*18:00'
CK_LINES: 'EXEC LINESCAN'
ACCTRULE: IF CLK.SUBMIT THEN 'SUBMIT X'
- SELF DRIVEN
recursion done on rules/tasks/patts/masks
recursion via FIREIRUNIGETIOPS commands
Example...
HANOI: RUN('MOVE 6 LEFT MIDDLE RIGHT')
MOVE: PARSE VAR RUNSTR ?N ?A ?B ?C
 IF ?N ^= 1 THEN DO
 ?M = ?N - 1
 RUN('MOVE ?M ?A ?C ?B')
 SAY 'MOVE DISK' ?N 'TO' ?C
 RUN('MOVE ?M ?B ?A ?C')
 END

Expert Systems In REXX

REXRULES CHAININGS

- **BACKWARD** using literals & assigned facts
maingoal = 'animal'
rule1: if mammal & carnivore & striped,
 then animal = tiger
rule2: if haired then mammal = true
rule3: if meateater then carnivore = true
ask.striped = 'is animal striped? Y/N'
equ.haired = 1*1/1+0-false /* ans = 1 */
- **FORWARD** using file based facts
FIRE('PICK HOLD DROP STOP',FOR 100)
PICK: IF OPS(GOAL,'TASK="ADD"') AND,
 OPS(BRICK,'PLACE="HEAP"'),
 THEN OPS(MODIFY,BRICK,'A 10 HAND')
HOLD: IF OPS(GOAL,'TASK="ADD"') AND,
 OPS(BRICK,'PLACE="HAND"'),
 THEN OPS(MODIFY,GOAL,'DROP')
DROP: IF OPS(GOAL,'TASK="DROP"') AND,
 ETC...
- **MIXED**, unset value in fired rule starts GOAL.
Once GOAL set fired rule resumes. Fires in
GOALs takeover & when done GOAL resumes.

(C)Copyright 05/91 By Marc Vincent Irvin

Expert Systems In REXX

REXRULES ATTRIBUTES

- **STATIC**
Sets query, process, & english info controls.
ASK.VIEW = "Select VIEW from:@" CHK.VIEW
DFT.VIEW = "FOR" /* default reply */
CHK.VIEW = "FOR AGAINST"
FMT.VIEW = 1 7 ALPHABETIC 3
WHY.VIEW = "Need to know side you're on,"
 "OK!" /* continuation example */
DOC.VIEW = "Only reply is FOR or AGAINST."
IDK.VIEW = "AGAINST" /* ie. I don't know */
- **TRUTH**
Uses boolean true/false logic for REXX vars.
YES = 1; NO = 0; DONTKNOW = '@'
ASK.EATSMEAT = 'Does animal eat meat? Y/N'
CHK.EATSMEAT = 'CHK_YESNO:' /*set 1 1 0 */
IF EATSMEAT = NO THEN HERBIVORE = TRUE
IF EATSMEAT=DONTKNOW THEN EATSMEAT=1
IF EATSMEAT THEN DO
 HERBIVORE = NO; CARNIVORE = YES; END
ELSE CARNIVORE = FALSE

(C)Copyright 05/91 By Marc Vincent Irvin

Expert Systems In REXX

REXRULES ATTRIBUTES

■ DYNAMIC

Most attributes can execute tasks or rules.

ASK.NAME ▪ "ASK_NAME:" /* use name task */

DFT.NAME ▪ "DFT_NAME:" /* default task */

CHK.NAME ▪ "CHK_NAME:" /*check name lgc*/

PNL.NAME ▪ "PNL_NAME: name addr phone"

ASK_NAME: /* msges below shown to user */

SAY "It's" TIME(), please enter name."

SAY "Thank you."

DFT_NAME: /*assume William's using system*/

if am then rspns ▪ BOB; else rspns ▪ WILL

CHK_NAME:

IF AM THEN,

IF FIND('BOB TIM JAY',RSPNS) > 0,

THEN RUNCC ▪ 1; ELSE RUNCC ▪ 0

PNL_NAME:

! --- FULLSCREEN INFO ENTRY ---

ENTER NAME: %NAME !

ADDRESS: %ADDRS !

PHONE: %PHONE !

EOP:

For @ panel var invalid dialog asks for input.

100

(C)Copyright 05/91 By Marc Vincent Irvin

Expert Systems In REXX

REXRULES ATTRIBUTES

■ CHRONOLOGIC

Rules can be run after N hrs, mins, or secs.

Rules can be run based on "date" or "time".

Rules can be run every N hrs, mins, or secs.

Variables used to signal if time has come.

CLK.PAYRULE ▪ "M30" /*Do 30 mins from now*/

CLK.CHKDASD ▪ "H1*21:00" /*@ hr, stop 9pm*/

* On May 30th tell operator's to go home.

CLK.MSGOPER ▪ "91/05/30 09:00 0.M10*12"

* Issue clock request for 10 PM shutdown.

CLK.STOPRUN ▪ DATE(O) '22:00 0'

PAYRULE: "MSG ALL IT'S PETTY CASH TIME."

* test of CLK. val needed to ctl miss fires.

CHKDASD:

IF CLK.CHKDASD THEN FIRE(DASDRULES)

SPECIAL_RULE_CHECKS_CLOCK_STUFF:

SELECT

WHEN CLK.MSGOPER THEN RUN(TELLOP)

WHEN CLK.STOPRUN THEN EXIT 000

OTHERWISE NOP

END

If no rule then only CLK. switch is set.

(C)Copyright 05/91 By Marc Vincent Irvin

Expert Systems In REXX

REXRULES ATTRIBUTES

- IMPRECISE
IF RUN(BIG: A) OR RUN(MED: A),
 AND ^RUN(LOW: A) THEN A_OK = TRUE
BIG: PARSE VAR RUNSTR X
IF X >= 9 THEN RUNCC = 1; ELSE RUNCC = 0
MED: PARSE VAR RUNSTR X; RUNCC = 0
IF X < 10 AND X > 5 THEN RUNCC = 1
- FUZZY (FUZ. is certainty factor if sw on)
CHK.QUALITY = " HIGH GOOD FAIR POOR "
FUZ_QUALITY = "0 1 .75 .5 .25 "
Output example:
 MEDIA_TO_CONSIDER = 80% FOILS
 MEDIA_TO_CONSIDER = 92% SLIDES
- EPISODIC
Assume FUZ. for quality/ ability/ quantity set
CHK.RATING = " QUALITY ABILITY QUANTITY "
DSS_RATING = "0 .345 .243 .161 "
Output example:
 RATING = .376 HAYAT, F
 RATING = .234 NISS, T

(C)Copyright 05/91 By Marc Vincent Irvin

Expert Systems In REXX

REXRULES PROS/CONS

- PROS
Needs no compile, but compiler possible.
Can mimic most PROLOG & OPS5 code well.
Perfect medium for teaching AI skills.
Grade schoolers could learn this in hours.
Intensely flexible parse ability kept.
Ideal for LADDER like NL implementations.
Relatively little cost to buy & maintain.
IBM's SAA stand is less support overhead.
Embodies six plus popular IE paradigms.
Highly flexible calculator ability.
Infers using both dialogs and panels.
- CONS
Not very fast CPU wise, nor IO wise.
Lacks high math capabilities.
Bleeding edge...
Few people use or have heard of REXX.
MVS version lacks EXECIO's index feature.
VM REXX Compiler can't do INTERPRET cmds.
Does not do LISP like list processing.
Has few syntax checking features yet.
Has little to no documentation yet.

(C)Copyright 05/91 By Marc Vincent Irvin

Expert Systems In REXX Towers of Hanoi in LISP

```
(defun tower-of-hanoi (disks from to spare)
  (unless (endp disks)
    (tower-of-hanoi (rest disks) from spare to)
    (format t "~%Move ~a from ~a." (first disks) from to)
    (tower-of-hanoi (rest disks) spare to from)))
```

III

Expert Systems In REXX Towers of Hanoi in Prolog

```
loc = right; middle; left
predicates
  hanoi(integer)
  move(integer, loc, loc, loc)
  inform(integer, loc, loc)
clauses
  hanoi(N) if
    move(N, left, middle, right).
  move(1, A, _, C) if
    inform(1, A, C), !.
  move(N, A, B, C) if
    M = N - 1,
    move(M, A, C, B),
    inform(N, A, C),
    move(M, B, A, C).
  inform(Disk, Loc1, Loc2) if
    write("\nMove disk ", Disk, " from ",
          Loc1, " to ", Loc2, ". ").
```

EXPERT SYSTEMS IN REXX

Expert Systems in REXX Towers of Hanoi in REXX

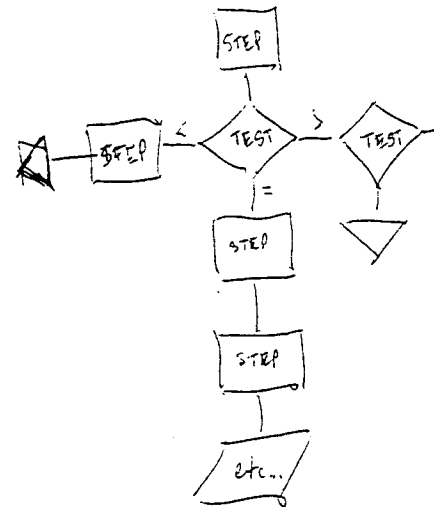
```

HANOI: RUN('MOVE' SIZE 'LEFT MIDDLE RIGHT')
MOVE: PARSE VAR RUNSTR ?N ?A ?B ?C .
  IF ?N = 1,
  THEN SAY 'MOVE DISK' ?N 'TO' ?C
  ELSE DO
    ?M = ?N - 1
    RUN('MOVE ?M ?A ?C ?B ?N')
    SAY 'MOVE DISK' ?N 'TO' ?C
    RUN('MOVE ?M ?B ?A ?C')
  END
  
```

112

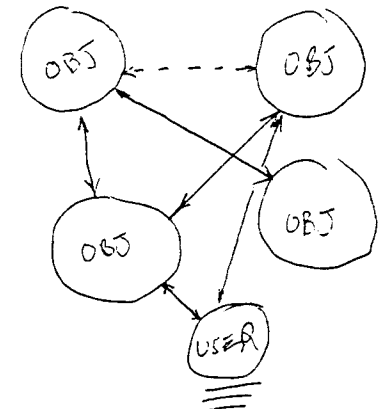
(C)Copyright 05/91 by Marc Vincent Irvin

PROCEDURAL



Take out any step
or test and logic
collapses.

DECLARATIVE OR OBJECT ES



Goes to whatever
object will provide
needed information
For example if
any of the above were
~~was~~ removed it
would just try to
get ANSWER FROM
the user.