Automated Knowledge Engineering

Maanda Raudzingana

Supervisor: Dr. Karen Bradshaw

Overview

- Recap
- Design & Implementation
- Evaluation
- Conclusion
- Questions

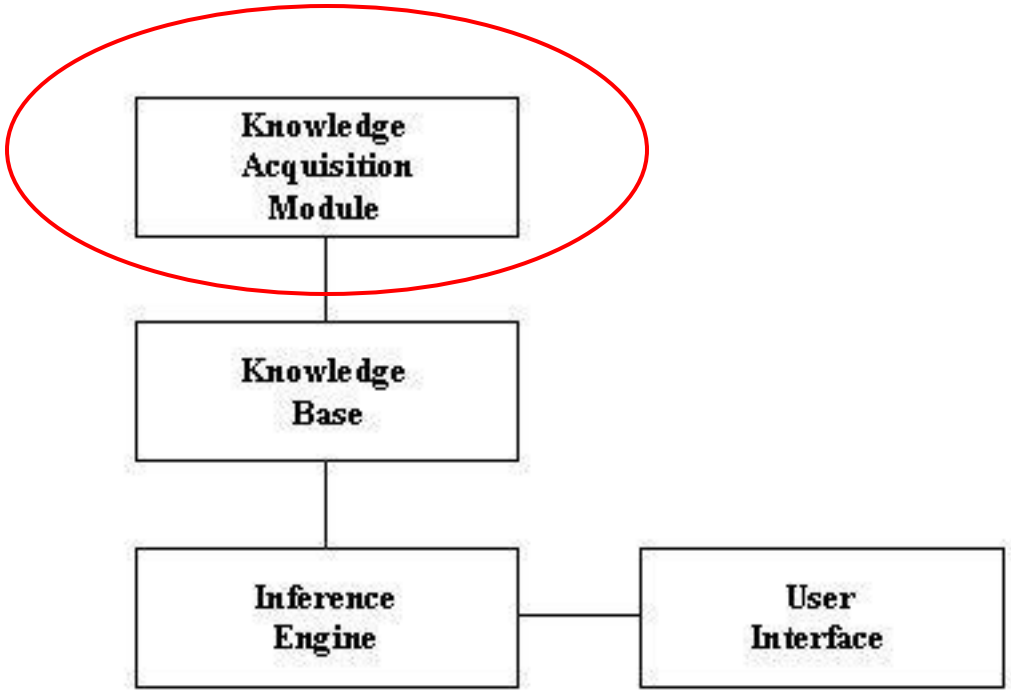
Expert systems

- **Knowledge based systems** that **mimic decision-making** ability of **human expert** in a **specific domain** of knowledge
- Rely heavily on complete and reliable knowledge
- Knowledge engineering bottleneck
- Problems
  - Time constraints
  - Communication
  - Constant reconfiguration complicated
  - Costs

Research objectives

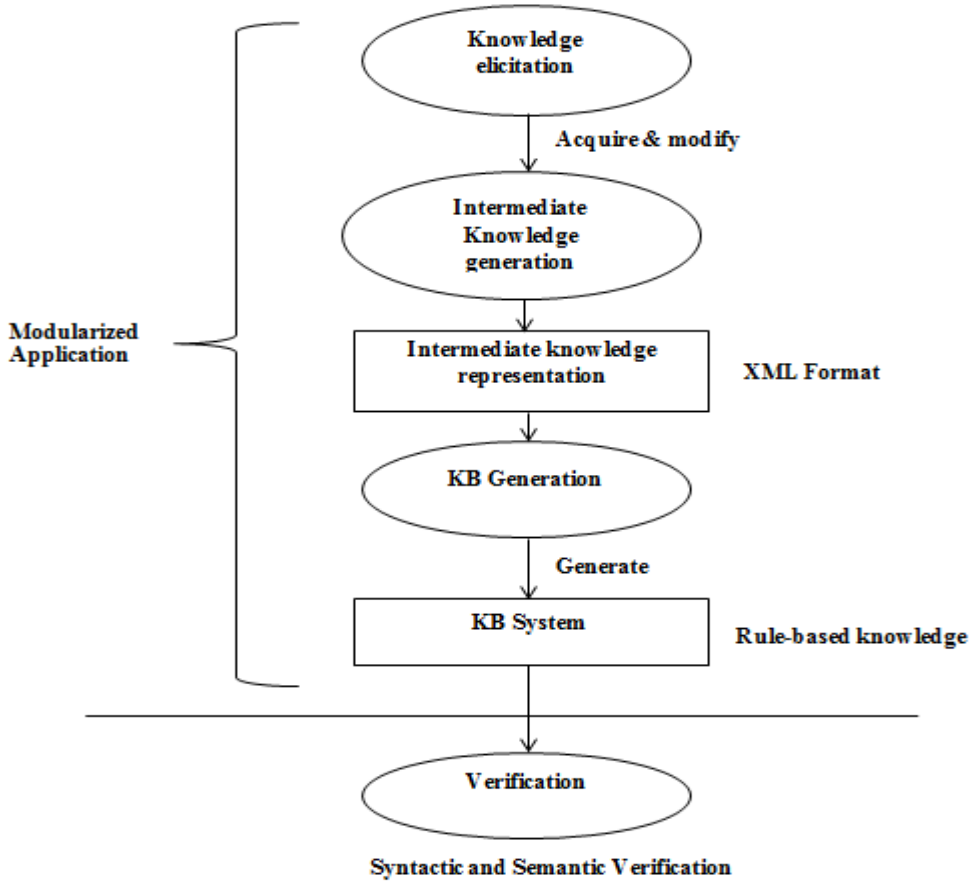
- Investigate use of knowledge base construction tools
- Automation of knowledge base construction?
- Develop a tool to show proof of concept
  - Allow conversion of facts into knowledge
  - Allow generation of executable knowledge base
  - Allow executing program to reach valid conclusions

Architecture of an expert system



a) Architecture of an expert system

Design & Implementation

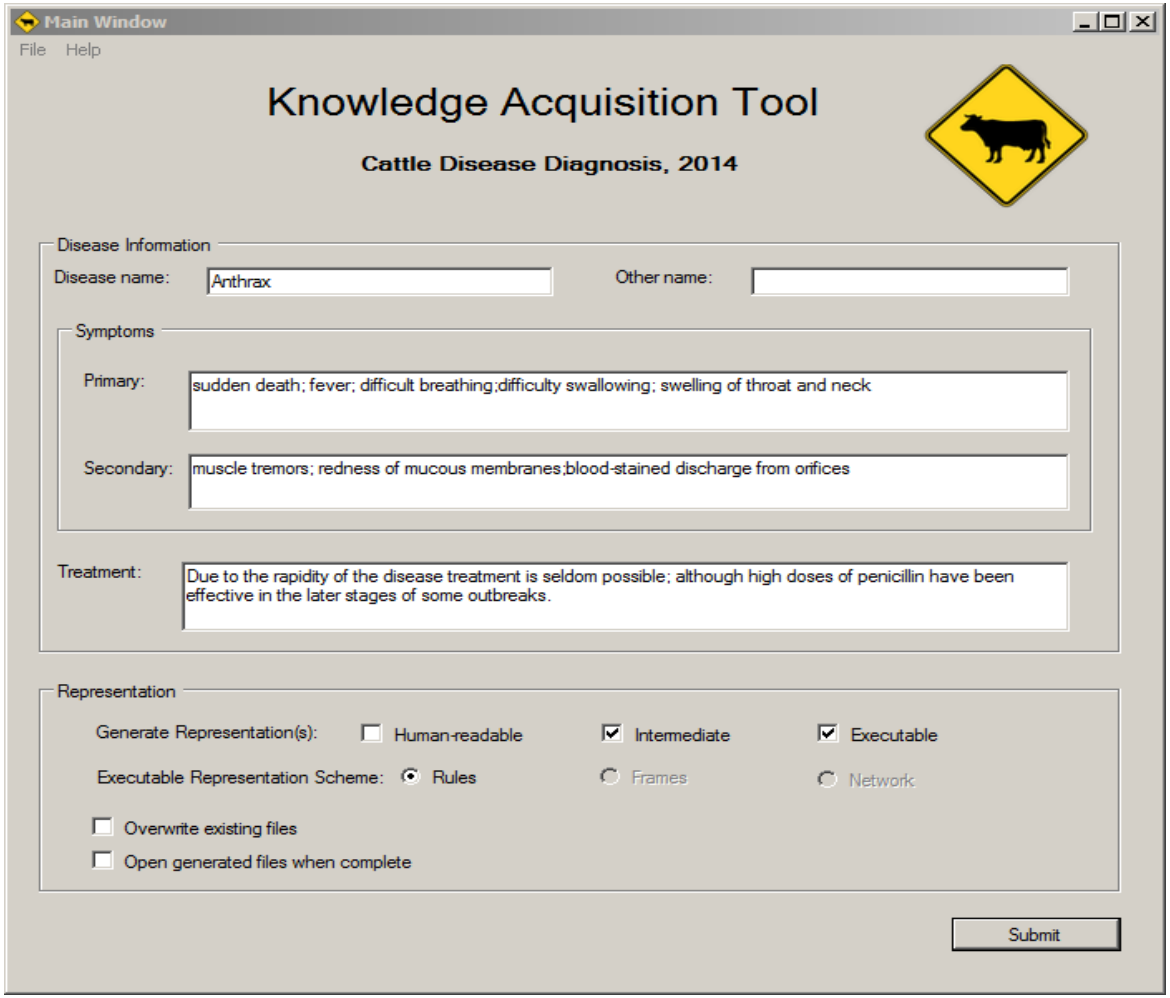


b) Architecture of proposed tool

1. Knowledge elicitation

- Acquiring the knowledge from the expert
- Cattle disease diagnosis
  - Identification of symptoms and advice on action
- **Graphical user interface** to allow input of facts
  - Windows Forms
- Design facilitates what information is stored
  - Data elements of interest
- Elicited data stored for review purpose

1. Knowledge elicitation



2. Intermediate knowledge representation

- Prepossessing data into intermediate representation
  - Integration of modules or other system
  - Storage
  - Monitoring
- XML representation
  - Excellent for data storage and transportation
  - Several tools available to process XML
- **Intermediate knowledge generator** written in C#
- Invoked on request for knowledge base generation
- Input to rule generator

2. Intermediate knowledge representation

```
<DiseaseList>
    <disease>
        <name>Anthrax</name>
        <othername>Anthrax disease</othername>
        <symptoms>
            <mainSymptoms>
                <symptom>Fever</symptom>
                <symptom>Sudden death</symptom>
                <symptom>Difficult breathing</symptom>
            </mainSymptoms>
            <secondarySymptoms>
                <symptom>swelling of the throat and neck</symptom>
            </secondarySymptoms>
        </symptoms>
        <treatment>Vaccinate livestock annually against anthrax.</treatment>
    </disease>
<DiseaseList>
```

3. Knowledge base generation

- Rule-based representation
- Adheres to specific syntax
- Rule algorithms/types govern rule construction
  - Example rule: Conclude diagnosis if all main symptoms present
  - Example rule: Conclude diagnosis if probable and at least one main
- **Rule generator** written in C#
  - Intermediate representation as input
  - Recompile existing knowledge if specified
- Rules as final system output

3. Knowledge base generation

```
Rule 10
    IF diarrhoea = Yes
    OR loss_of_mass = Yes
    OR loss_of_condition = Yes
    THEN ParatuberculosisPrimaryPresent = PRESENT
    Paratuberculosis = probable
    NumProbable = (NumProbable + 1)
    ParatuberculosisChecked = TRUE
    B = (B+2)
    ELSE ParatuberculosisPrimaryPresent = FALSE;

Rule 11
    IF Fever = Yes
    OR Bilsters = Yes
    OR Lameness = Yes
    OR salivation = Yes
    OR smacking_of_the_lips = Yes
    OR grinding_of_the_teeth = Yes
    OR nasal_discharge = Yes
    THEN Foot-and-mouth-diseasePrimaryPresent = PRESENT
    Foot-and-mouth-disease = probable
    NumProbable = (NumProbable + 1)
    Foot-and-mouth-diseaseChecked = TRUE
    C = (C+2)
    ELSE Foot-and-mouth-diseasePrimaryPresent = FALSE;
```

Evaluation/Verification

I. Knowledge represented as rules
II. Correct conclusions/diagnoses
III. Physical testing – executable?

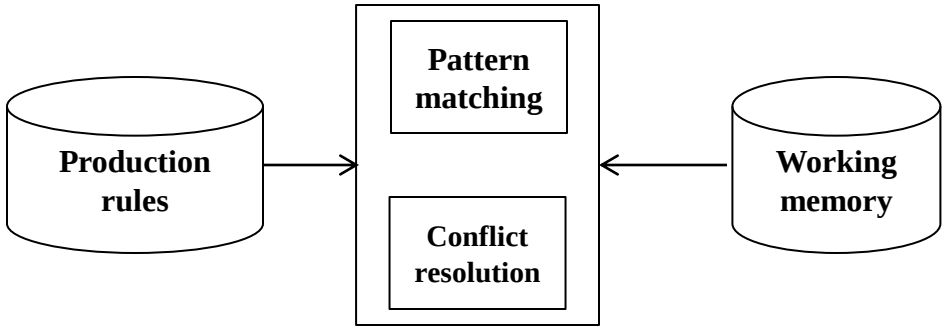
I. Knowledge as rules

- Rule logic: conclude diagnosis if the disease is **probable**, and there's at least one primary symptom, and other diseases have been checked (no conflicts)

```
84  Rule 12
85      IF Botulism = probable
86      AND  BotulismPrimaryPresent = PRESENT
87      AND  AnthraxChecked = TRUE
88      AND  BrucellosisChecked = TRUE
89      THEN  Disease = Botulism;
```

II. Correct conclusions

- Proposed hypothetical inference engine



- Keep a score for each disease during runtime
  - Symptoms reported: (a, b, c, d, e)
  - Diseases in system: (D1, D2, D3)
  - Membership of symptoms D1 (a, d, g, h, j) D2(a, b, c, e, f) D3(a, i, j)
  - **Diagnosis:** D2

III. Executable

- Tailor to specific expert system shell/inference engine
- VP-Expert 1.2 (student version)
- Minor changes to rule base
  - Restructure rules
  - Change syntax
  - Add control & query statements
- Result:
  - Generated knowledge base recognized and loads with no errors
  - Generated knowledge base is executable
  - Accurate diagnoses, sometimes
- Limitation from VP-Expert inference engine
  - Temptation to make up for shortcomings in knowledge base

Testing in VP-Expert shell

- Diseases in system:
  - Anthrax
  - Brucellosis
  - Botulism

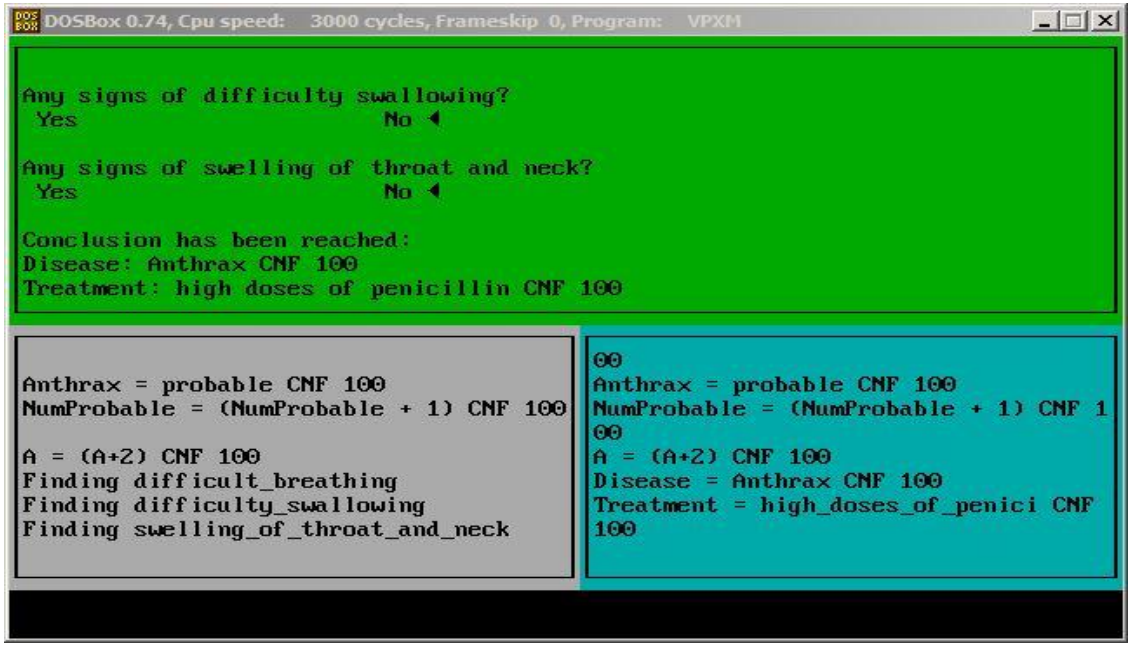
Reported symptoms:

Anthrax (2 primary, 1 secondary)

Brucellosis (1 primary)

Botulism (1 primary)

Diagnosis = Anthrax



Testing (2)

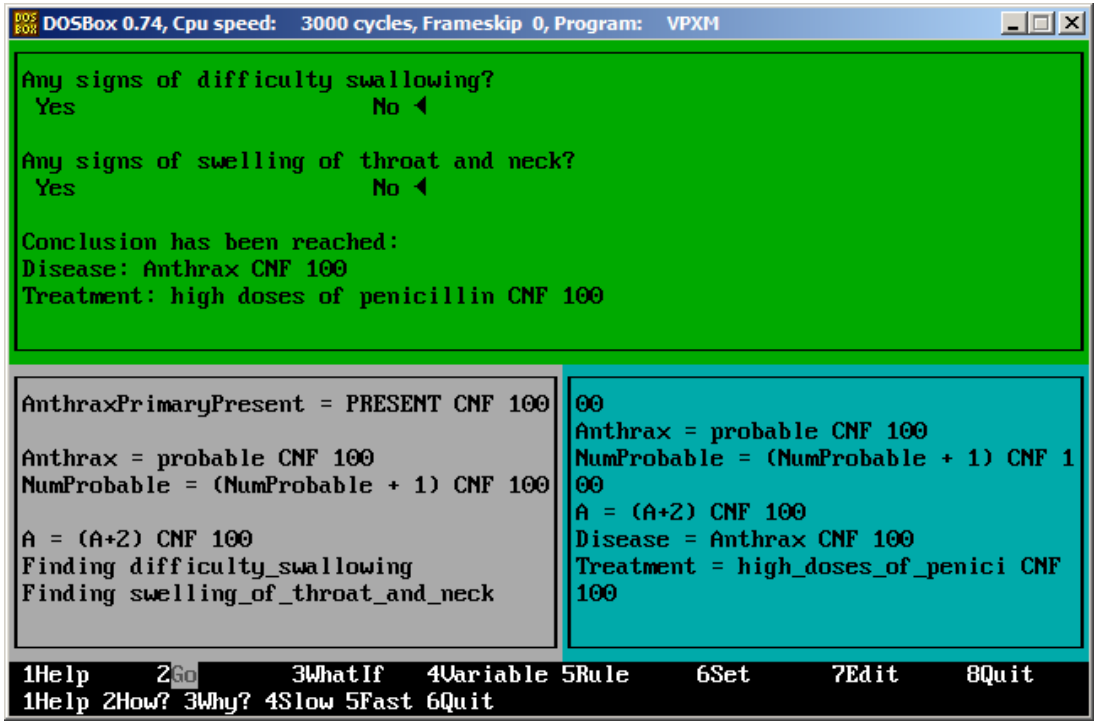
- Diseases in system:
  - Anthrax
  - Brucellosis
  - Botulism

  Reported symptoms:

  Anthrax (2 primary)

  Botulism (2 primary)

  Diagnosis = Anthrax



Testing (3)

- Diseases in system:
  - Anthrax
  - Brucellosis
  - Botulism

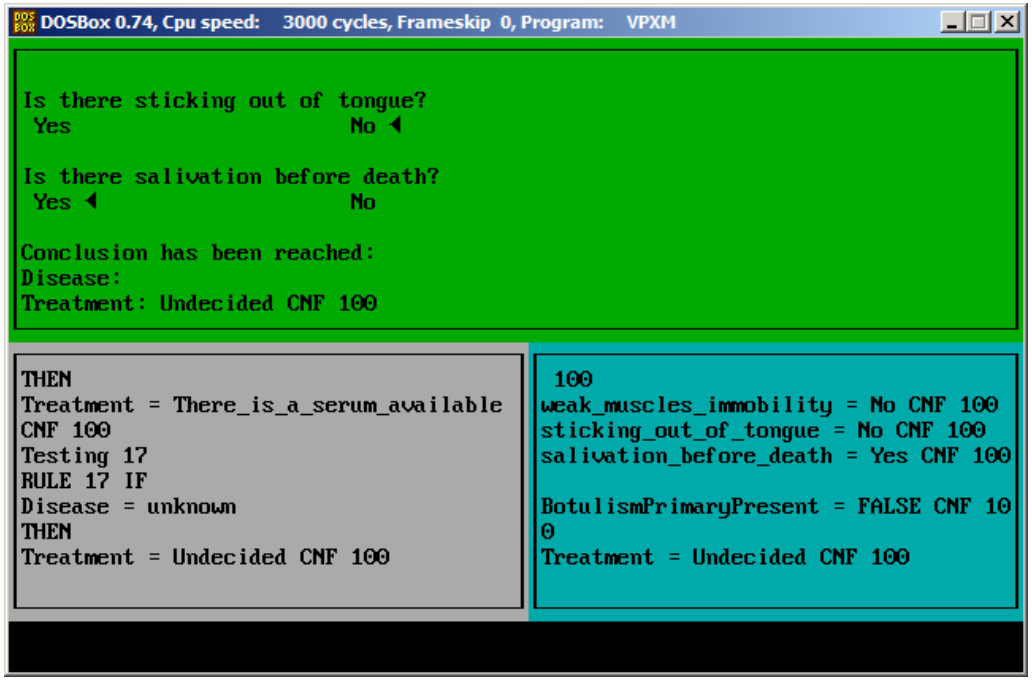
  Reported symptoms:

  Anthrax (1 primary)

  Brucellosis (1 primary)

  Botulism (1 primary)

  Diagnosis = Undecided



Conclusion

- Possible to automate knowledge base construction
- Successful addition of new knowledge
- However, challenging to separate knowledge base from inference engine
- Minor changes needed to tailor knowledge bases to different inference engine
- Future work:
  - To overcome limitations from external programs, custom inference engine
  - Dynamic acquisition
  - Hybrid representation scheme

Questions

?