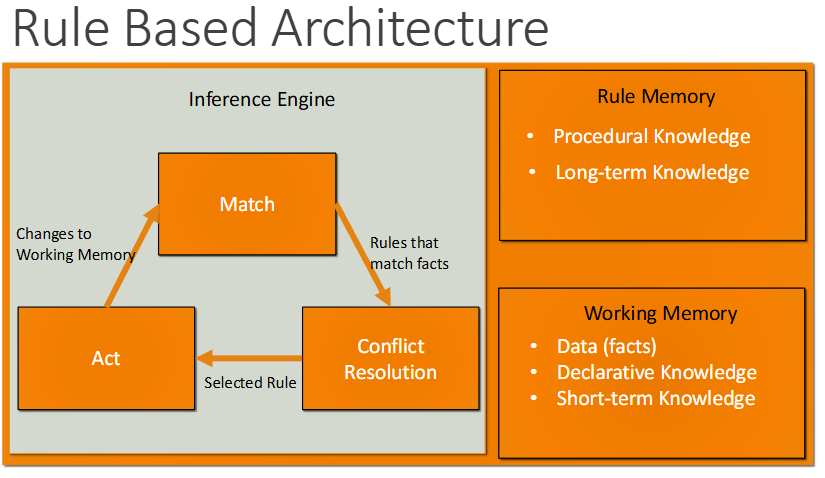
**NOTES**

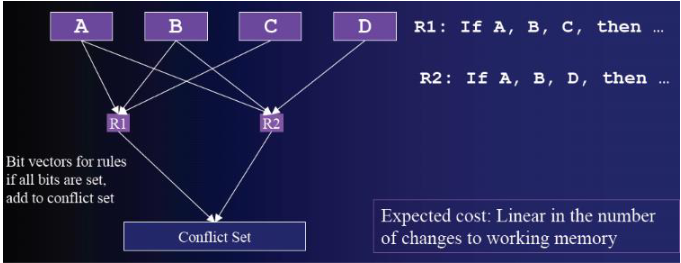
**Requirements for a rule based architecture**

* rule-base
* Stores information about subject domain
* Rules are the main way of expressing this knowledge
* If then format, IF <condition> THEN <conclusion>
* working memory
  + Requires Facts to function, encoded within WM
  + Represents current set of assignments of facts to variables
* inference engine
  + Deployment of evidence to arrive at new conclusions
  + Consists of search and reasoning procedures enabling solutions to a problem to be found
  + Typically follow one of two top level strats -> forward or backward chaining
* forward chaining
  + Attempts to match known facts to available rules
* Backward chaining
  + Determines the facts forms a condition

**Rule based architecture**

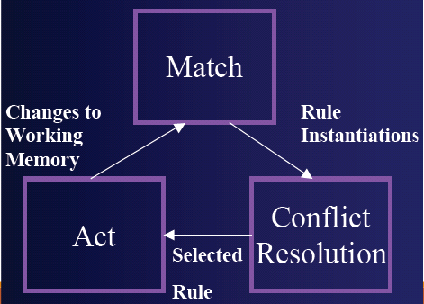


* Picking the next rule to fire
  + If only simple tests in conditions, compile rules into match net



**Conflict resolution**

* Which match rule should fire
* Which instantiation of a rule should fire?
  + Separate instantiation for every match of variables in rules



* Conflict resolution filters
  + Refractory Inhibition
    - Don’t fire same instantiation that has already fired
  + Data Recency:
    - Select instantiations that match most recent data
  + Specificity:
    - Select instantiations that match more working memory elements
  + Random
    - Select randomly between the remaining instantiations
* Other conflict res strats
  + Rule order
    - Pick first rule that matches
      * Makes order of loading important – bad for big systems
  + Rule importance
    - Pick rule with highest priority
      * Right 80% of time, forces total order on rules

**RBS Pros and Cons ->**

* Advantages
  + Corresponds to way people think of knowledge
  + Expressive
  + Modular knowledge
  + Easy to write and debug compared to decision tress
  + More concise than FSM
* Disadvantages
  + Memory intensive at times
  + Computationally intensive at times
  + Difficult to debug at times

**Rules**

* Consists of two parts
  + - If part also known as antecedent
    - Then part called the consequent
    - IF <antecedent> THEN <consequent>

-----------------------------------------

    📑 Notes for the Report

-----------------------------------------

-- -- Structure (suggested headings) -- --

- Introduction

    ->    Brief description of your problem domain (the tree you were assigned). - DONE

    ->    Purpose of expert systems and your chosen approach. - DONE

- System Description

    ->    How you converted the decision tree into rules.

    ->    How your knowledge base and inference engine work.

    ->    Example of rules (but don’t just paste the code — explain).

- Consultation Examples

    ->    Screenshots or transcripts of runs.

    ->    Show different outcomes.

    ->    Comment on how well it worked.

- Evaluation of System

    ->    Strengths (e.g. accurate outputs, tailored questions).

    ->    Weaknesses (e.g. limited scope, possible ambiguities).

    ->    Suggestions for improvements (e.g. adding more knowledge, handling uncertainty).

- AI Context

    ->    Rule-based systems in wider use (medical diagnosis, ticket pricing, etc.).

    ->    Ethical/commercial/social issues in your domain (e.g. misclassification consequences, user trust).

    ->    Human factors: explainability, usability, user interface.

- Knowledge Acquisition & Learning

    ->    Difficulties of acquiring rules from experts.

    ->    Inductive learning (e.g. decision tree learning from data) vs. manual rule creation.

    ->    Why uncertainty/complexity is challenging for rule-based systems.

- Conclusion

    ->    Summary of achievements.

    ->    Future work.

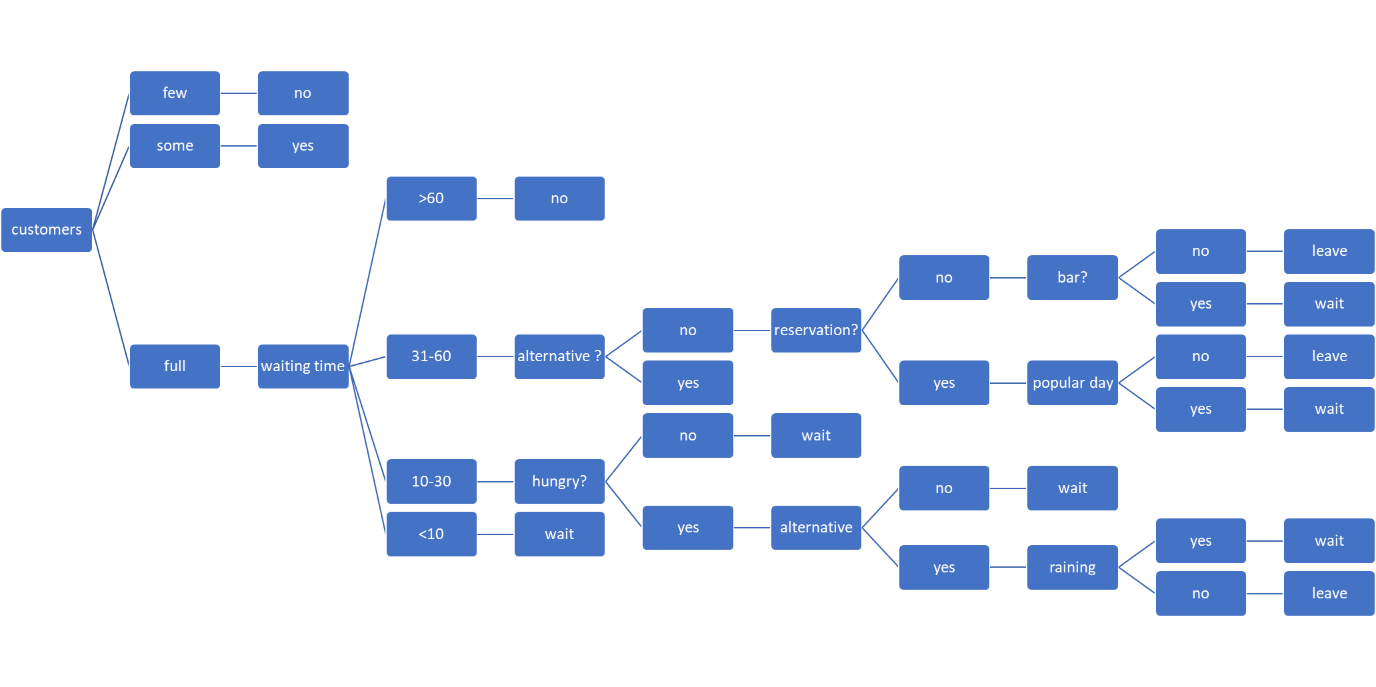
    ->    References

    ->    Academic and technical sources (properly formatted).

- Appendices

    ->    Full code (as text).

    ->    Link to Panopto demo.



    Whether to wait for a table can be a subjective exercise.  If this tree were to be adapted to help businesses determine whether punters are liable to wait, and to adjust elements of their business practice to optimise the number of customers, how would this be done?