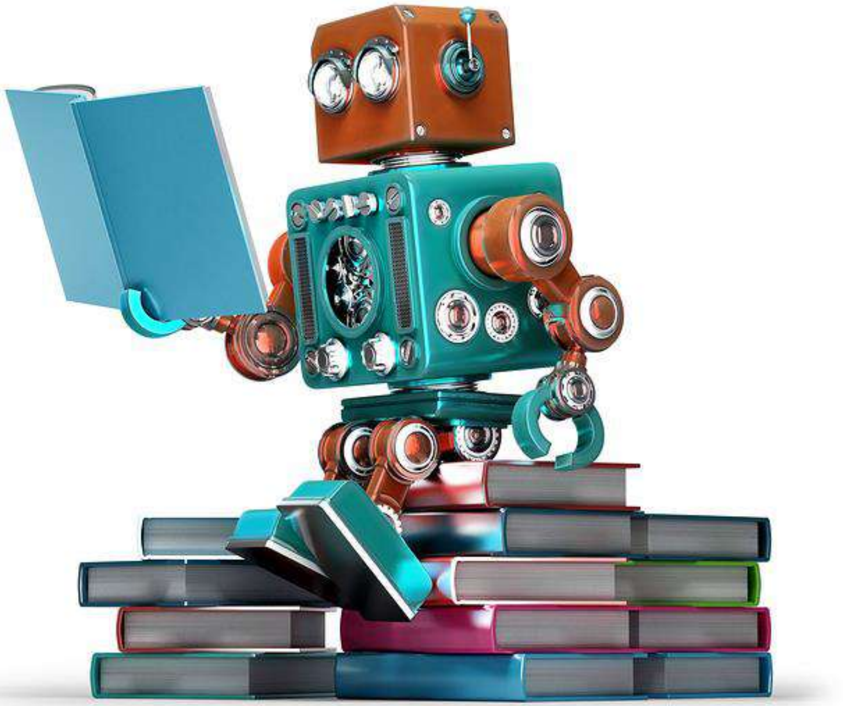
MACHINE REASONING

DAY 2



https://robohub.org/wp-content/uploads/2016/11/bigstock-Retro-Robot-Reading-A-Book-ls-110707406.jpg

DAY 2 AGENDA

2.1 Knowledge Acquisition (Business Rules)

2.2 Knowledge Models (Acquired → Represented)

Format Acquired Knowledge into Representation Templates

2.3 Machine Inference (part 1)

2.4 Knowledge Modelling **Workshop**

DAY 2 TIMETABLE

No	Time	Topic	By Whom	Where
1	9 am	2.1 Knowledge Acquisition 2.2 Knowledge Models	GU Zhan (Sam)	Class
2	10.10 am	Morning Break		
3	10.30 am	2.3 Machine Inference (1/2)	GU Zhan (Sam)	Class
4	12.10 pm	Lunch Break		
5	1.30 pm	2.4 Knowledge Modelling Workshop Tutorial	GU Zhan (Sam) All	Class
6	3.10 pm	Afternoon Break		
7	3.30 pm	2.4 Knowledge Modelling Workshop	All	Class
8	4.50 pm	Summary and Review	All	Class
9	5 pm	End		

2.2
KNOWLEDGE ACQUISITION (BUSINESS RULES)

2.1 KNOWLEDGE ACQUISITION (BUSINESS RULES)

- **Knowledge Acquisition is the transfer and transformation of problem solving knowledge into a form that can be used to build intelligent systems.**
- **Knowledge acquisition is also called:**
  - Knowledge capture
  - Knowledge elicitation
  - Requirements engineering
- **Personnel involved:**
  - Knowledge holder, e.g. subject matter expert (SME); process owner
  - Knowledge engineer, e.g. business analyst; system analyst; consultant

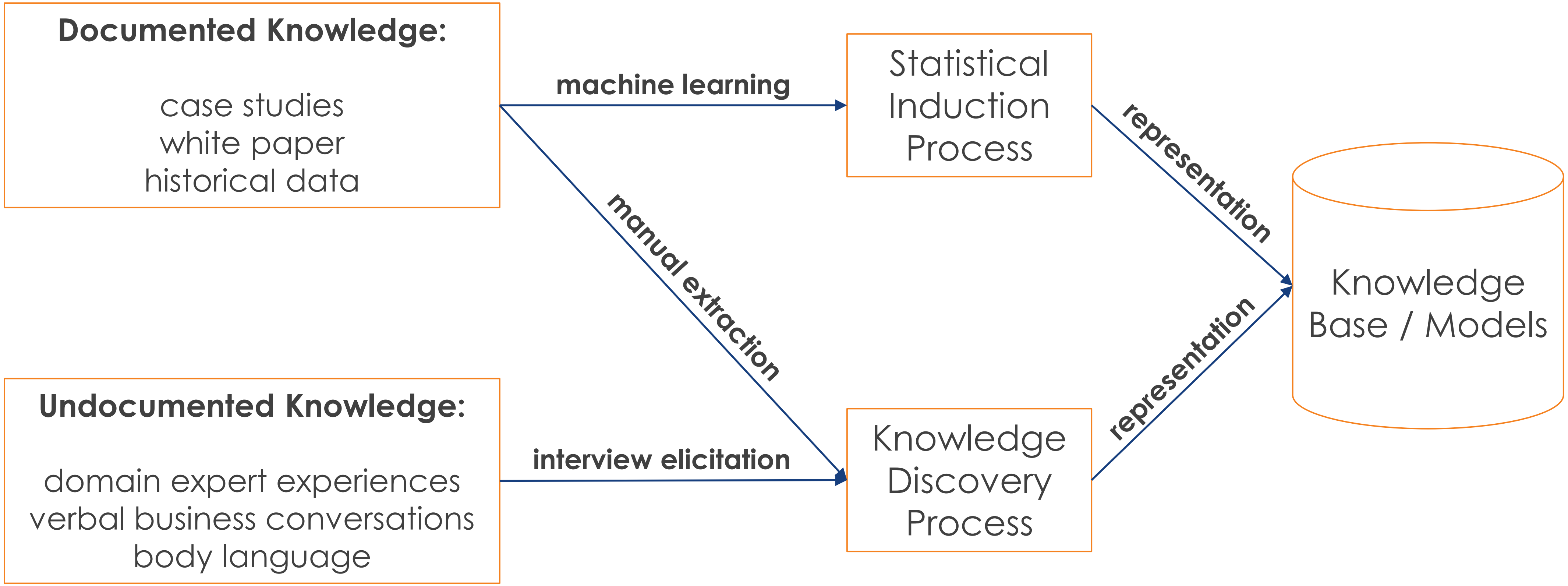
2.1 KNOWLEDGE ACQUISITION (BUSINESS RULES)

Knowledge Sources

- **Two types of knowledge can be used to build KBS/RBS:**
  - **Documented sources**
    - Collection from printed sources
    - Machine learning (rule induction, decision tree, neural network, deep learning, etc.)
  - **Undocumented sources**
    - Tacit knowledge that can only be captured by elicitation from human experts

2.1 KNOWLEDGE ACQUISITION (BUSINESS RULES)

Acquisition Methods



Documented Knowledge:

case studies
white paper
historical data

Undocumented Knowledge:

domain expert experiences
verbal business conversations
body language

2.1 KNOWLEDGE ACQUISITION (BUSINESS RULES)

Interview Elicitation

- **Elicitation is acquisition of tacit knowledge from a subject matter expert**
- **The Knowledge elicitation approach:**
  - Capture knowledge using interviews
  - Interpret & Analyze the transcripts and data obtained
  - Build knowledge models (knowledge representation)
  - Use the knowledge models to guide further elicitation
  - Verify & Validate the captured knowledge
  - Stop when the knowledge model is enough for building business reasoning system

2.1 KNOWLEDGE ACQUISITION (BUSINESS RULES)

Interview Best Practices

- **More Beneficial interviewing expert at their workplace**
- **Make sure the meeting place is quiet and free from interruptions**
- **Before the interview:**
  - Do background research on the domain area
  - Background check on the domain expert
  - Design and phrase your questions
  - Email questions to domain expert
  - Acquire and prepare the tools for the interview
- **During the interview:**
  - Introductions & Social preliminaries
  - State purpose of interview
  - Give a brief on the roles and responsibilities
  - Be courteous; Listen closely; Avoid arguments
  - Investigate each topic in detail
  - Evaluate session outcome
- **Observe confidentiality**

2.2
KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

- **After acquiring domain knowledge from the experts and other sources, how do we present a comprehensive view of this knowledge?**
- **Knowledge Models (Templates for knowledge representation)**
  - A knowledge model is a group of **structured representations** of knowledge that allows us to better understand the domain and the processes involved in decision making.
  - Documented models provide rich descriptions of domain knowledge that is **independent** of any particular software implementation.
  - These models also serve as a basis for communication among stakeholders: experts, analysts, developers, and end users.

2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

Document Templates

• Concept Dictionary	
• Concept Tree	
• Composition Tree	
• Decision Tree	
• Goal Reduction Tree	
• Inference Diagram	KIE Rule Flow Groups: Task (Activity/Sub-Goal) level
• Attribute Worksheet	KIE Data Model: Object; Field; Type
• HMI/UI System-User Dialogue	KIE Form: Task level
• Rules & Decision Table	KIE Guided Rules; Decision Table
• Dependency Diagram	KIE Rule Flow Groups: Rule level
• Flowchart (Workflow)	KIE Process Flow: Task level for Business Functions
• Activity Flow Diagram	KIE Process Flow: Task level for Business Teams/Roles
• RACI Matrix	KIE Business Teams/Departments/Roles/Groups

2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

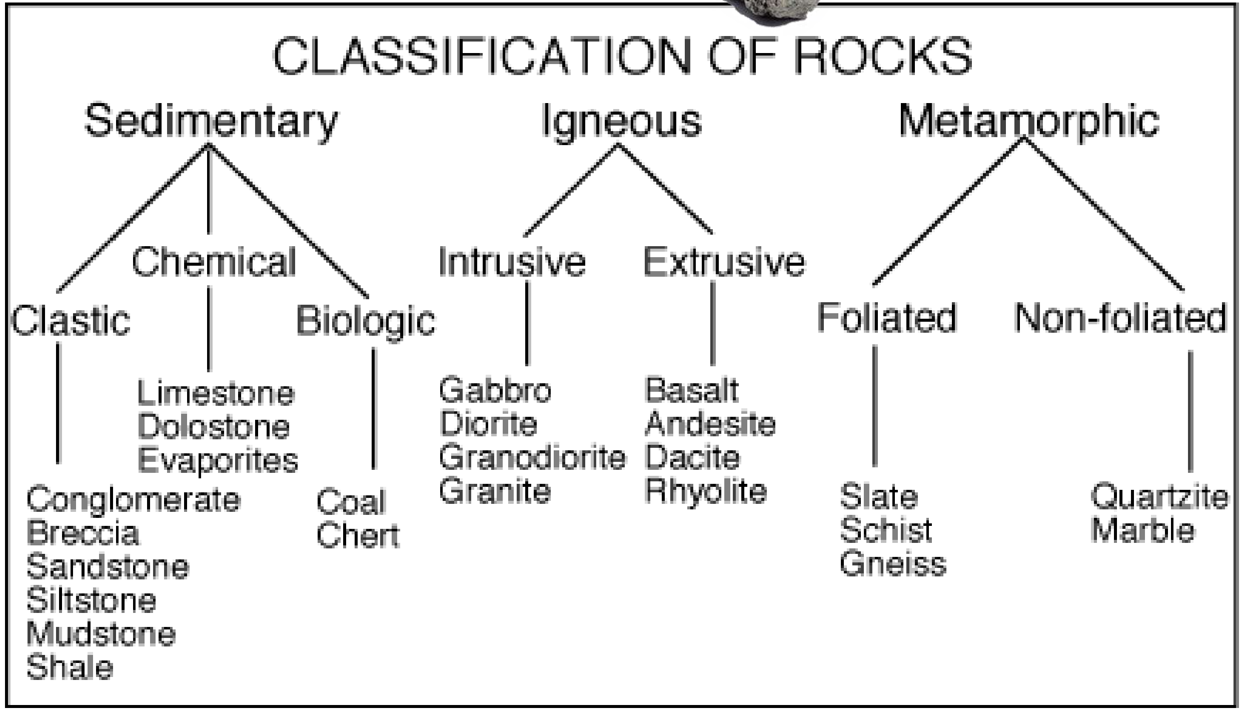
Concept Dictionary

Concept	Synonyms Abbr	Meaning
Esophagus	Gullet Oesophagus	Sometimes known as the gullet. Muscular tube through which food passes from pharynx to the stomach
Duodenum		The first section of the small intestine
Peptic ulcer	PUD	Area of the gastrointestinal tract that is extremely painful. Mucosal erosions equal to or greater than 0.5cm.
Hyperacidity	Acid dyspepsia, Amalpitta	A condition of excreting more than the normal amount of hydrochloric acid in the stomach

- A concept dictionary contains the list of **all relevant concepts** that are used in the problem domain to solve the problem.
- The dictionary provides a detailed explanation of the concept and can include any information that is useful for a good understanding of the concept.
- It can be similar to a **glossary**.
- It does not have any particular format.

2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

Concept Tree



CLASSIFICATION OF ROCKS

Sedimentary

Clastic
Conglomerate
Breccia
Sandstone
Siltstone
Mudstone
Shale

Chemical
Limestone
Dolostone
Evaporites

Biologic
Coal
Chert

Igneous

Intrusive
Gabbro
Diorite
Granodiorite
Granite

Extrusive
Basalt
Andesite
Dacite
Rhyolite

Metamorphic

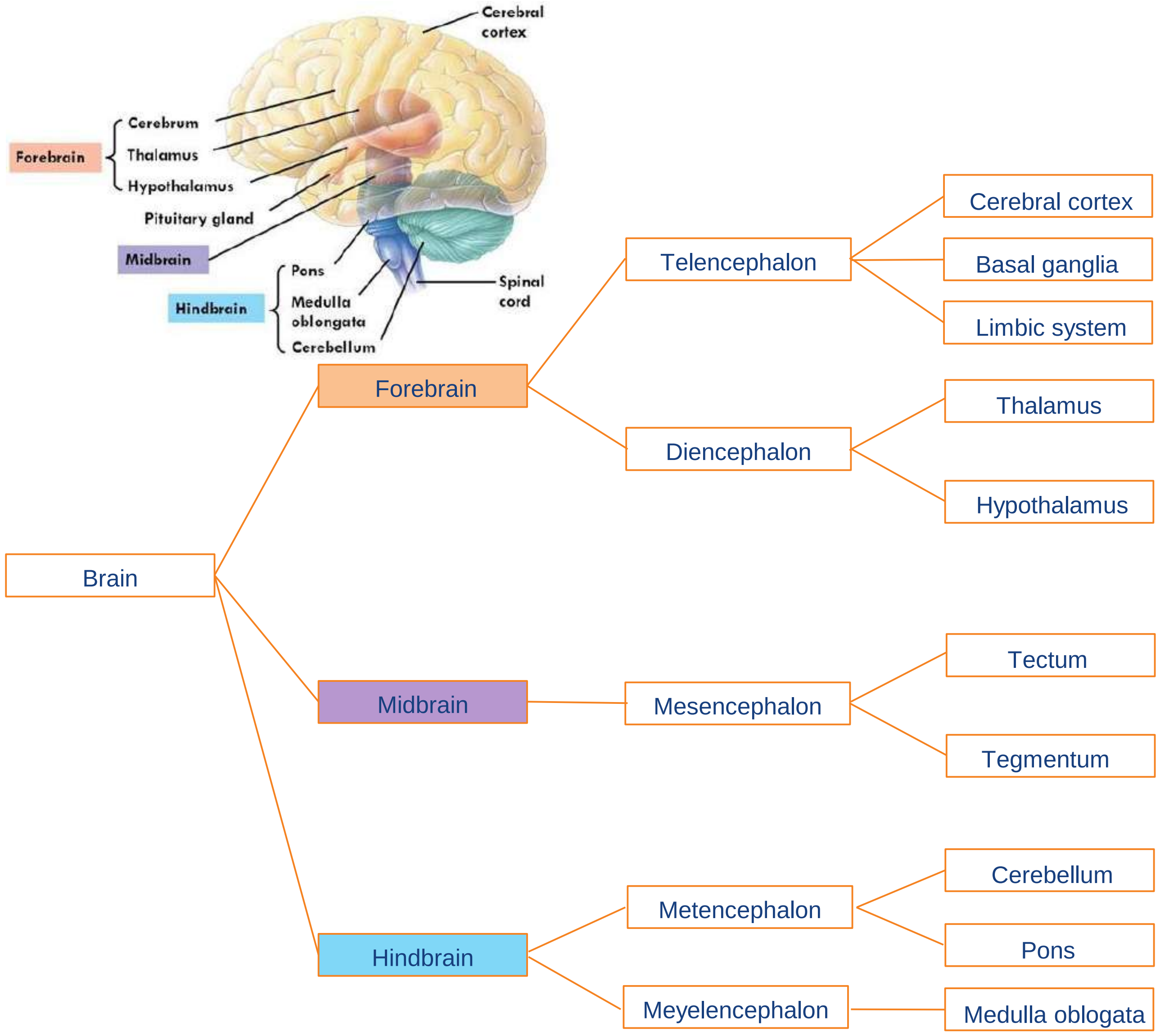
Foliated
Slate
Schist
Gneiss

Non-foliated
Quartzite
Marble

- Tree that shows concepts and the classes and sub-classes.
- All relationships must be “is a”.
- Check the tree by looking at the lowest and highest nodes and asking “is <sub-concept> a type of <concept>”.
- Nodes should have clear & complete names.
- Captured terminology and landscape of the domain.

2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

Composition Tree



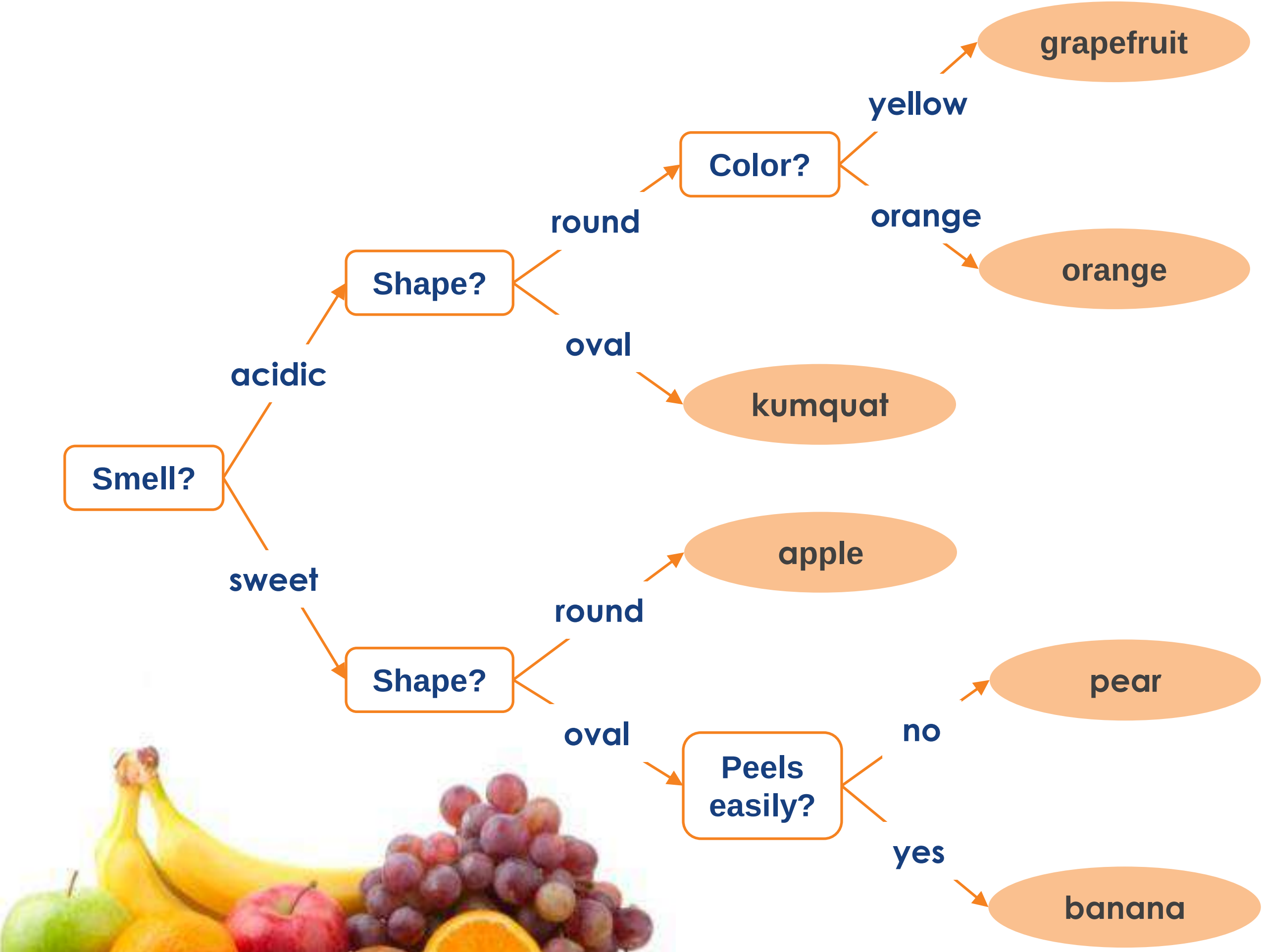
- Brain
  - Forebrain
    - Telencephalon
      - Cerebral cortex
      - Basal ganglia
      - Limbic system
    - Diencephalon
      - Thalamus
      - Hypothalamus
  - Midbrain
    - Mesencephalon
      - Tectum
      - Tegmentum
  - Hindbrain
    - Metencephalon
      - Cerebellum
      - Pons
    - Meyelencephalon
      - Medulla oblogata

http://blog.askiitians.com/wp-content/uploads/2016/05/Brain-1.jpg

- **Detailed concept breakdowns into its constituent parts.**
- **All relationships must be "part of".**
- **Understand things as**
  - Products (parts of a machinery)
  - Organisations (your organisation chart)
  - Documents (the table of contents)

2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

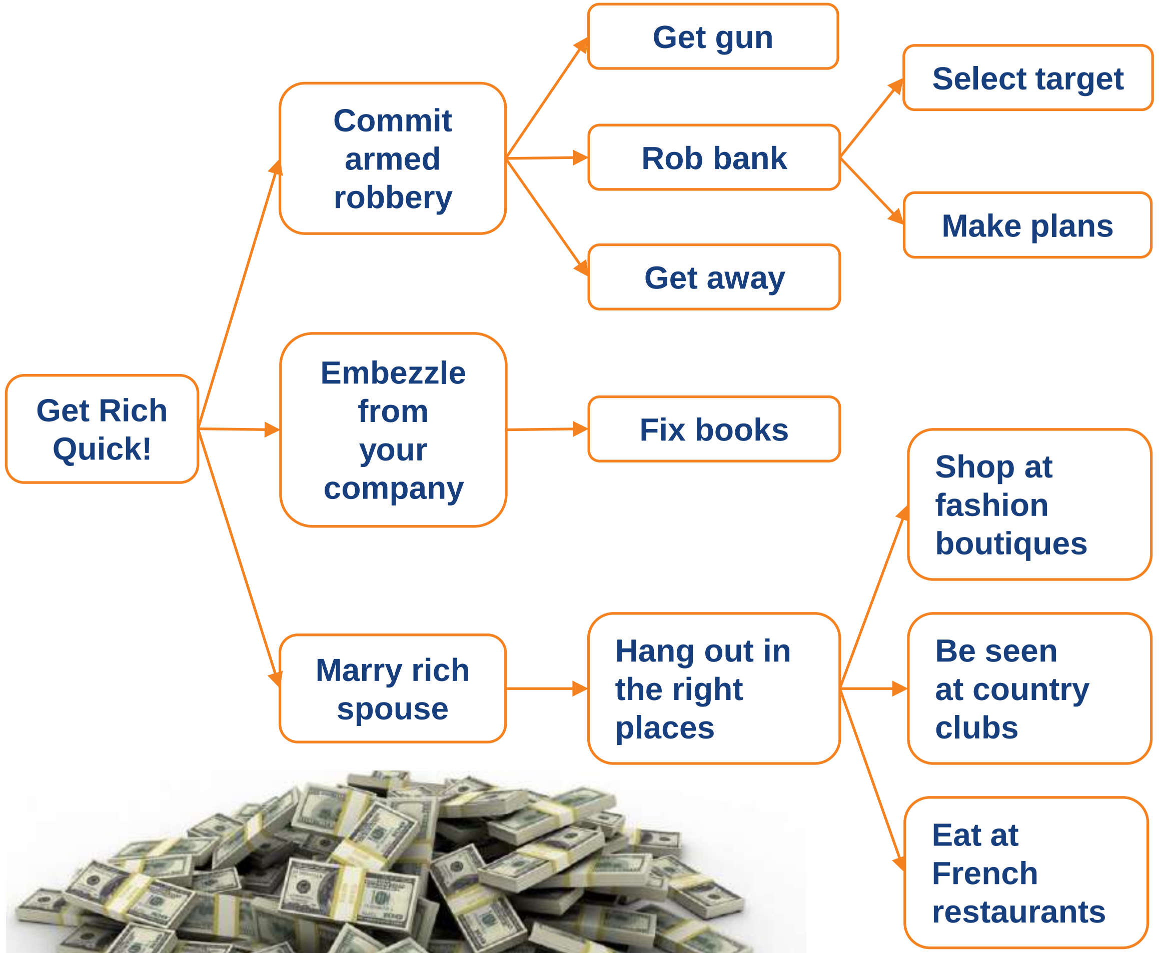
Decision Tree



- Tree shows the alternative courses of action or casual consequences for a particular decision.
- Condition/Rule based domain knowledge
- A snapshot of the experts knowhow

2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

Goal Reduction Tree



- Indicate relations between goal & sub-goals
- Incorporates AND/OR links (OR is implicit)
- Typical patterns of problem-solution behavior can be analyzed

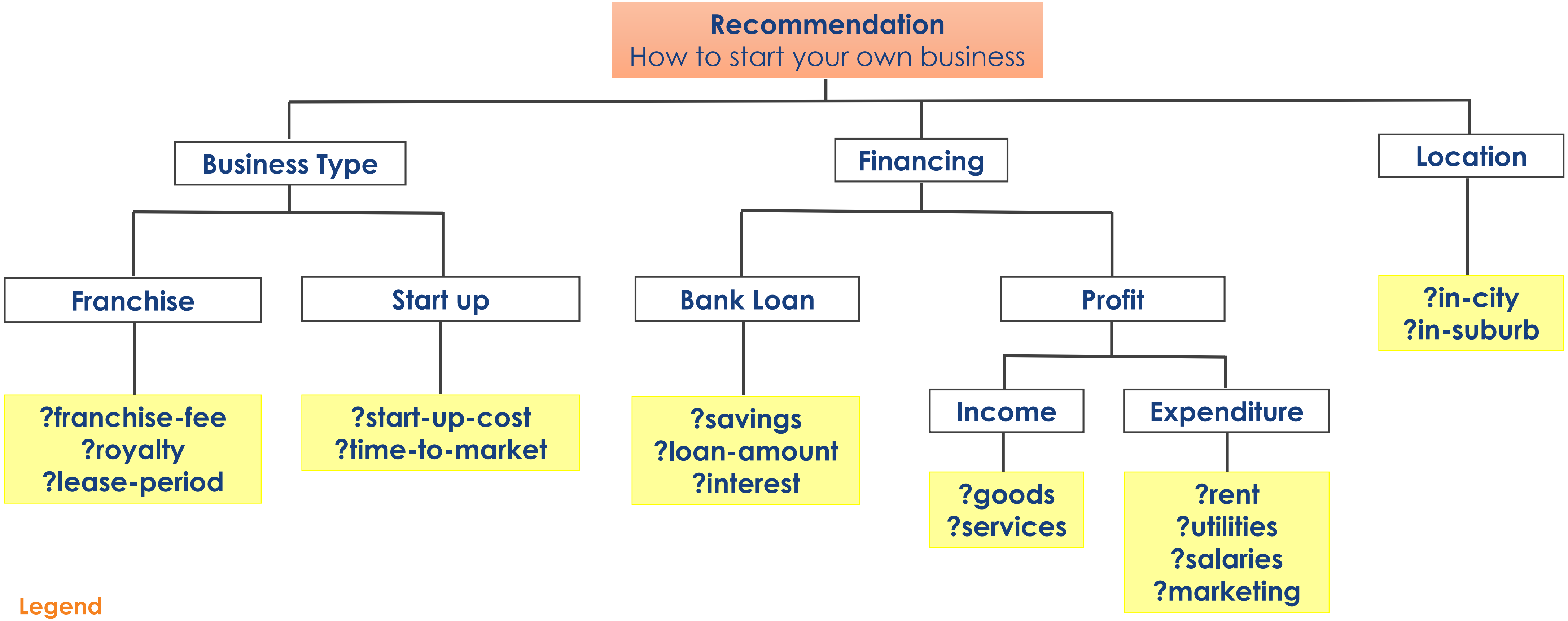
2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

Goal Reduction Tree vs. Decision Tree

- **Goal Reduction Trees are**
  - More "action" oriented while the Decision Tree is more static and causal.
  - Useful for planning a overall strategy "sub-goals" for problem solving
- **Decision Trees are**
  - More detailed conditional rule sets
  - Have "labeled links" where every node has a specific question/condition to check

2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

Inference Diagram KIE Rule Flow Group; Task



Legend

Top-level Inference

Inferable Sub-goals

?Observables / Facts

2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

Inference Diagram KIE Rule Flow Group; Task

- A Inference Diagram is a type of goal reduction tree, extended with detailed observable/controllable input factors.
- A Inference Diagram consists of problem-solving factors arranged in a hierarchical tree structure.
- The top-level (tree root) node is the final decision goal.
- The intermediate nodes are the sub-factors or sub-goals.
- The bottom-level (tree leaves) nodes are the input factors.
- An Inference Diagram shows the different level of inferences in the decision process.

2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

Attribute Worksheet   KIE Data Model: Object; Field; Type

Sub-goal	Attribute	Inferable or Observable	KIE Field Type & Value			English Translation
KIE Data Model: Data Object	**KIE Data Model: Object Field**	**KIE Form: User Interface**	**String, Integer, Float, Boolean, Date, etc.**	**Value Range**	**Value Unit**	**KIE Data Model: Comments**
Franchise	franchise-fee	Observable	Float	1 - 50,000	SGD $	The price to be paid for the franchise
	royalty	Observable	Float	1 - 10,000	SGD $	The monthly fee payable to franchisor
	lease-period	Observable	Integer	1 - 5	years	The Franchise Lease period
Profit	**Income**	**Inferable**	Float	1 – 1,000,000	SGD $	Annual income from sale of goods
	Expenditure	Inferable	Float	1 – 1,000,000	SGD $	Annual expenditure from sale of goods
Income	goods	Observable	Float	1 – 1,000,000	SGD $	The sales proceeds from goods sold
	services	Observable	Float	1 – 1,000,000	SGD $	Sales proceeds from services rendered
Location	in-city	Observable	Boolean	True or False	N.A.	Planned shop in city area
	in-suburb	Observable	Boolean	True or False	N.A.	Planned shop in suburb area

2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

UI System-User Dialogue KIE Form

Sub-goal	Franchise

System Questions / KIE Forms	Example User Response
How much is the franchise Fee?	$10,000
How much is the monthly royalty fee?	$2,000
How long is the lease period?	3 years

- Compose User Interface (UI) System-User Dialogues
- Define **questions** that will be asked by the reasoning system; Or **forms** to be filled by end user.
- Define responses that the user is expected to give
- Use System-User Dialogue tables

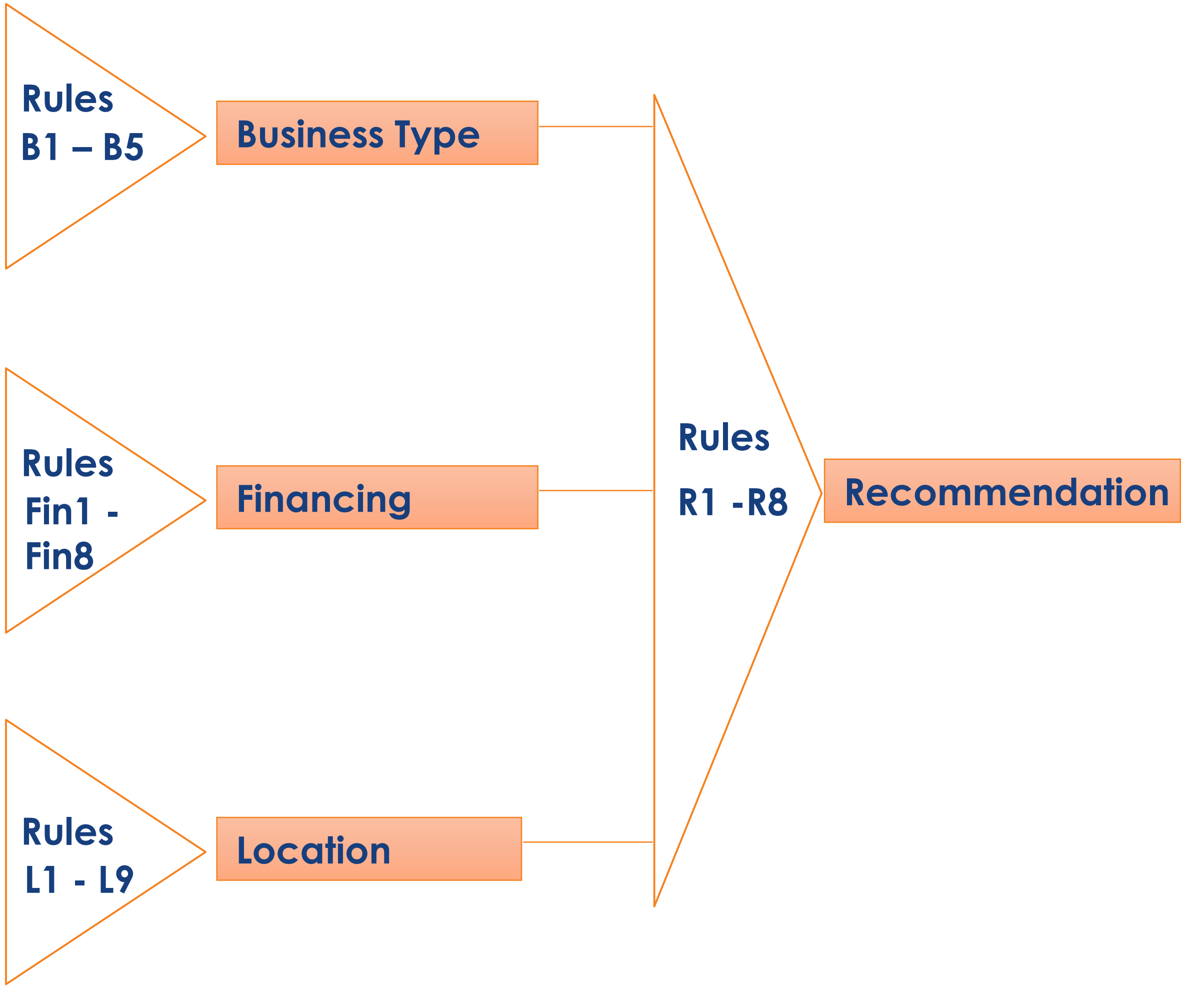
2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

Rules & Decision Table KIE Guided Rules; Decision Table

Rule No.	Condition 1	Logical Operand	Condition 2	Sub-goal
F-1	franchise-fee ≤ threshold1	AND	royalty ≤ threshold2	Franchise = ok
F-2	franchise-fee ≤ threshold1	AND	royalty > 20% x franchise-fee	Franchise = not-ok
F-?	…	…	…	…

2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

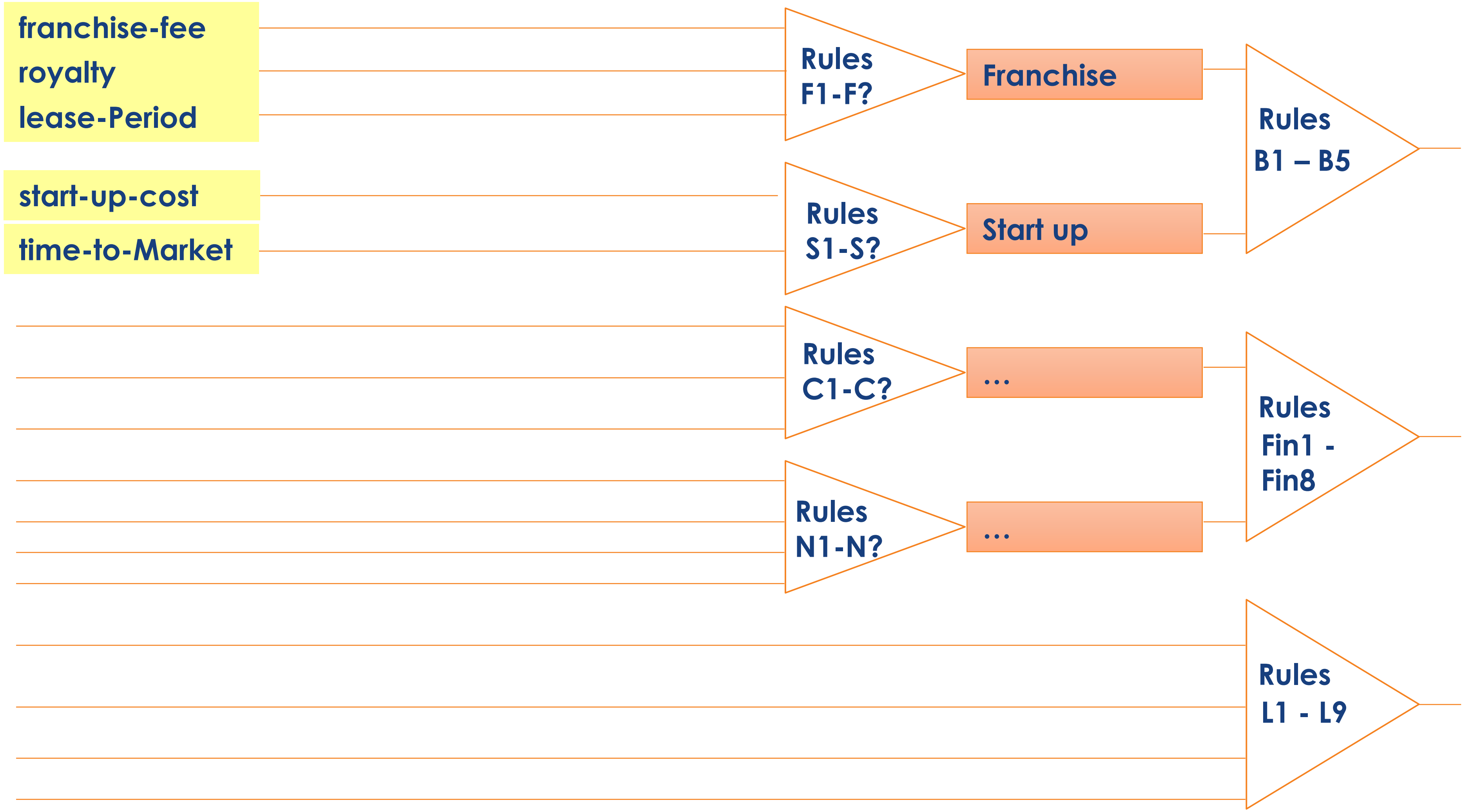
Dependency Diagram KIE Rule Flow Groups



- Dependency Diagram captures **sub-goals** with corresponding **rules**.
- It also shows the relationship between the different sub-goals (rule groups).
- To be used to construct rule flow groups based on sub-goals, in KIE Task.

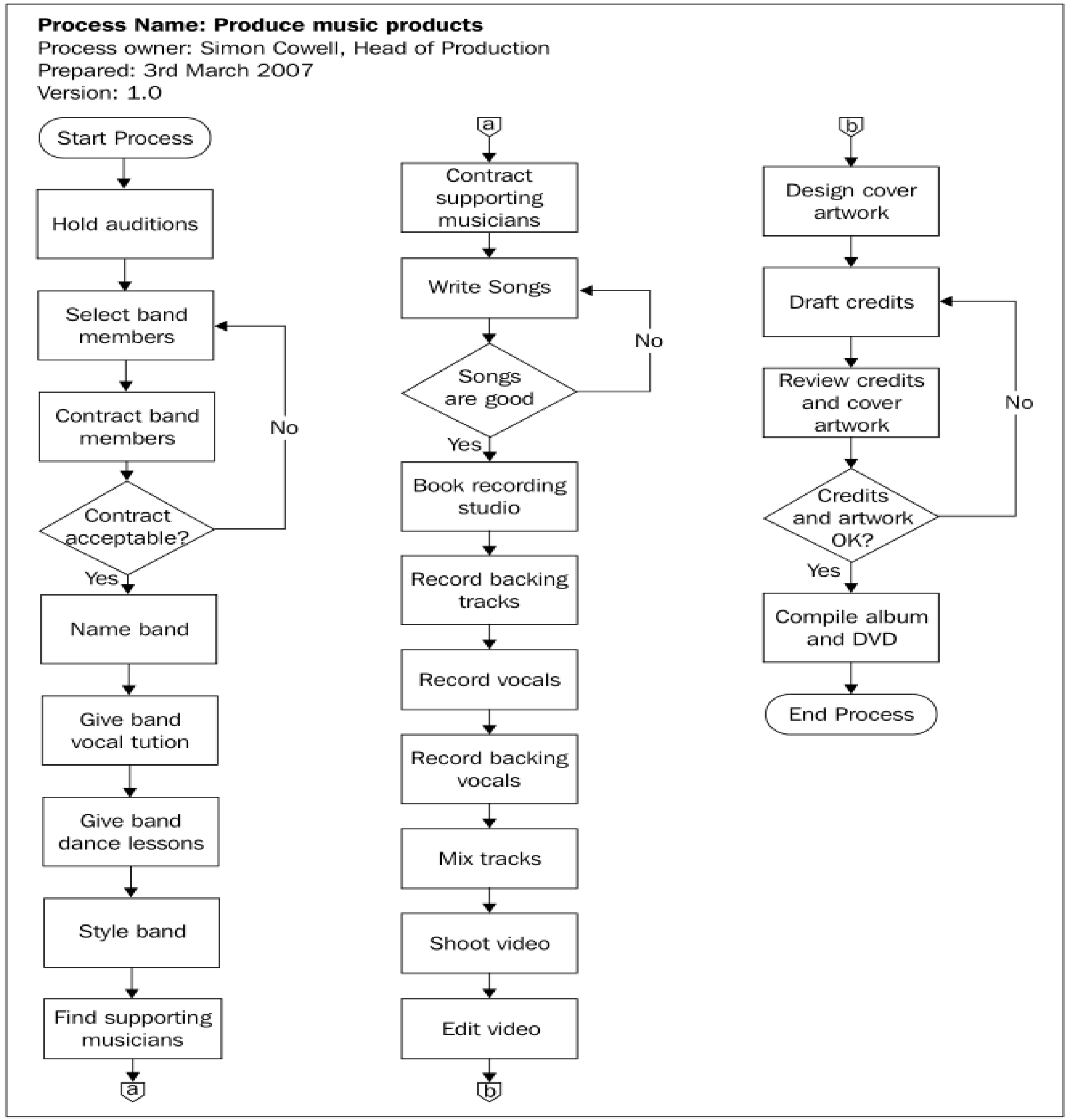
2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

Dependency Diagram KIE Rule Flow Groups



2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

Flowchart KIE Process Flow: Task level for Business Functions

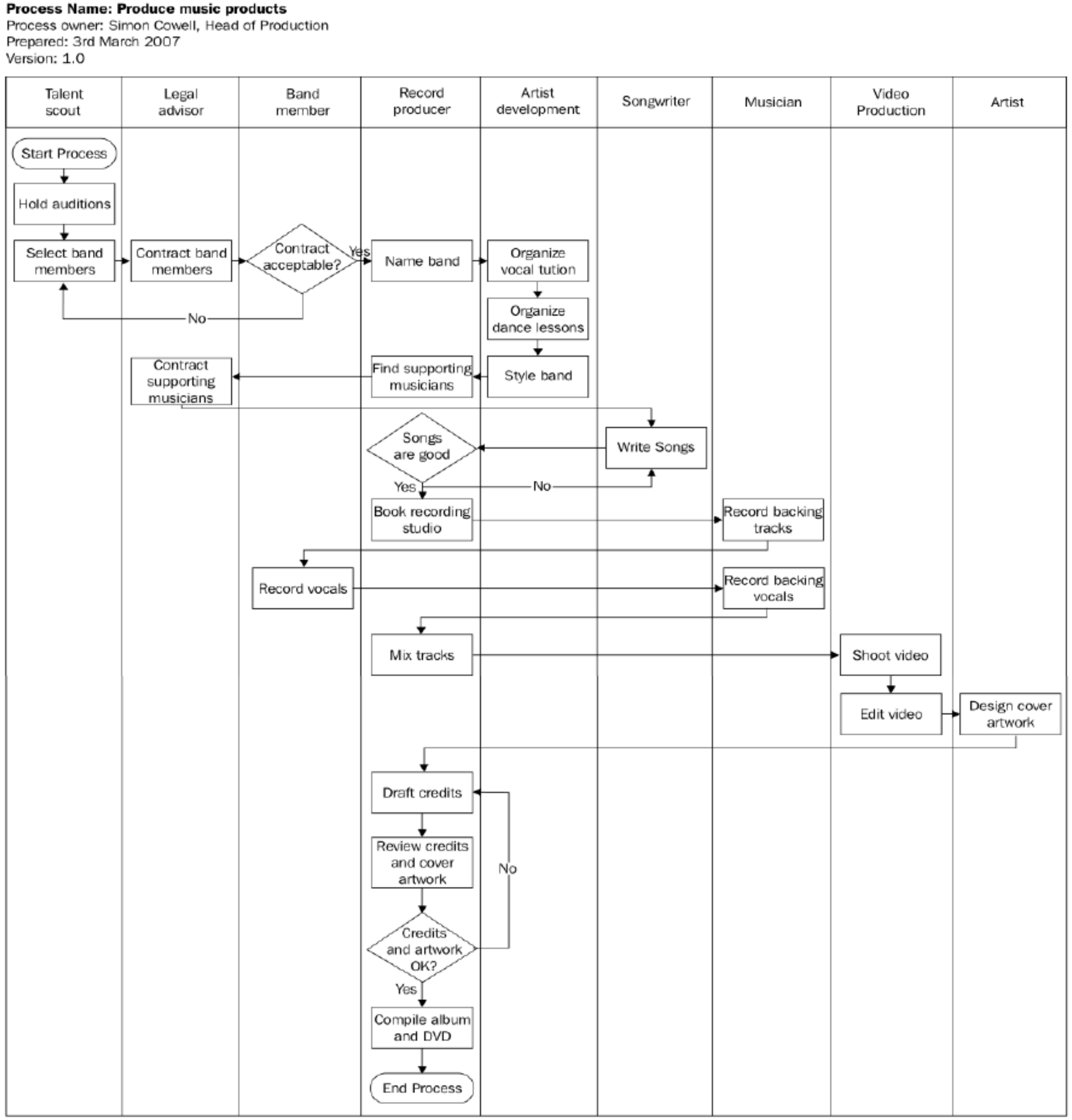


- The key to develop Flowchart to model/capture business process workflow, is to define the **sequence of activities**, and to identify those points where the flow can go two ways, depending on the circumstances.
- Write the activity name in as few words as possible, e.g. **Verb + Noun** pairs
- Write **decision points** as clear questions to which the answer is either "yes/true" or "no/false".

© 2007 Matt Cumberlidge. Business Process Management with JBoss jBPM. Packt Publishing Ltd

2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

Activity Flow Diagram KIE Process Flow: Task level for Business Teams/Roles



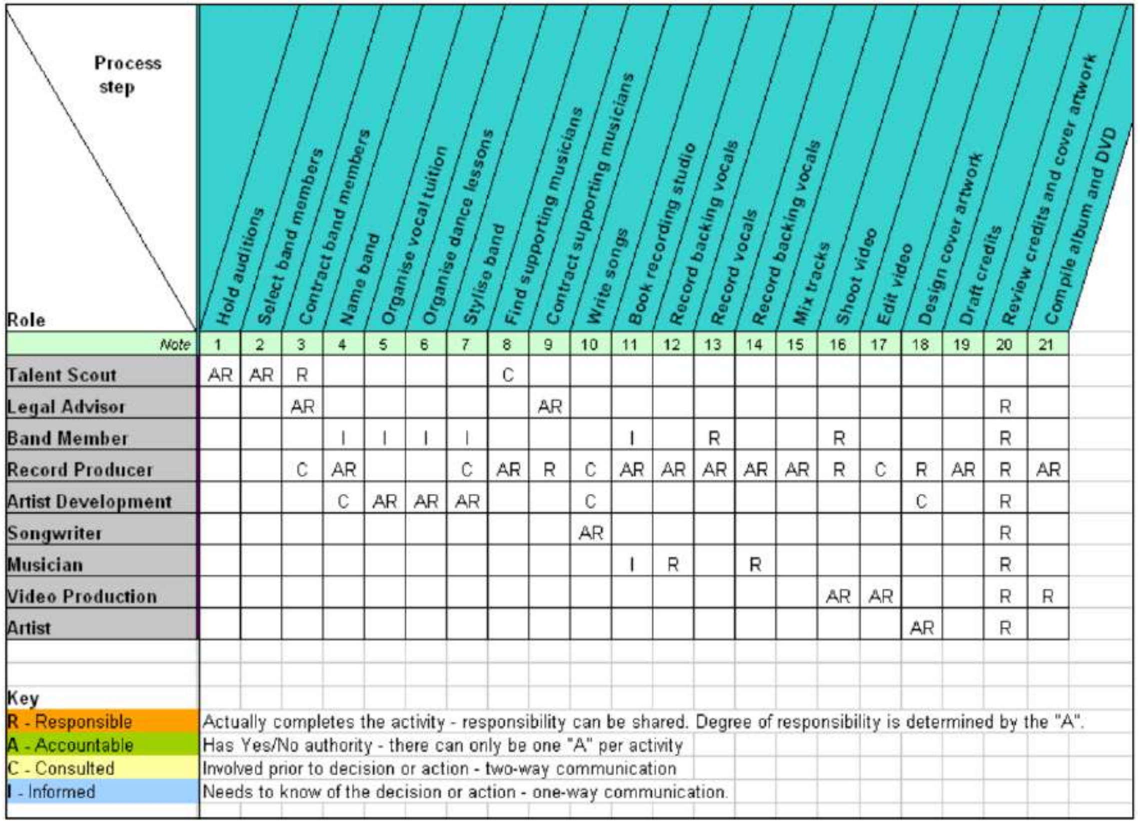
Process Name: Produce music products
Process owner: Simon Cowell, Head of Production
Prepared: 3rd March 2007
Version: 1.0

- Activity Flow Diagram captures "who does what (activity)" in the workflow.
- Identify roles and responsibilities (Swimlanes)
- A single activity should map to a single role. If this doesn't seem possible, then consider whether the activity should actually be split out into multiple activities.
- Expand flowchart by drawing swimlanes for each activity under the identified roles/teams.

© 2007 Matt Cumberlidge. Business Process Management with JBoss jBPM. Packt Publishing Ltd

2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

RACI Matrix   KIE Business Teams/Departments/Roles/Groups



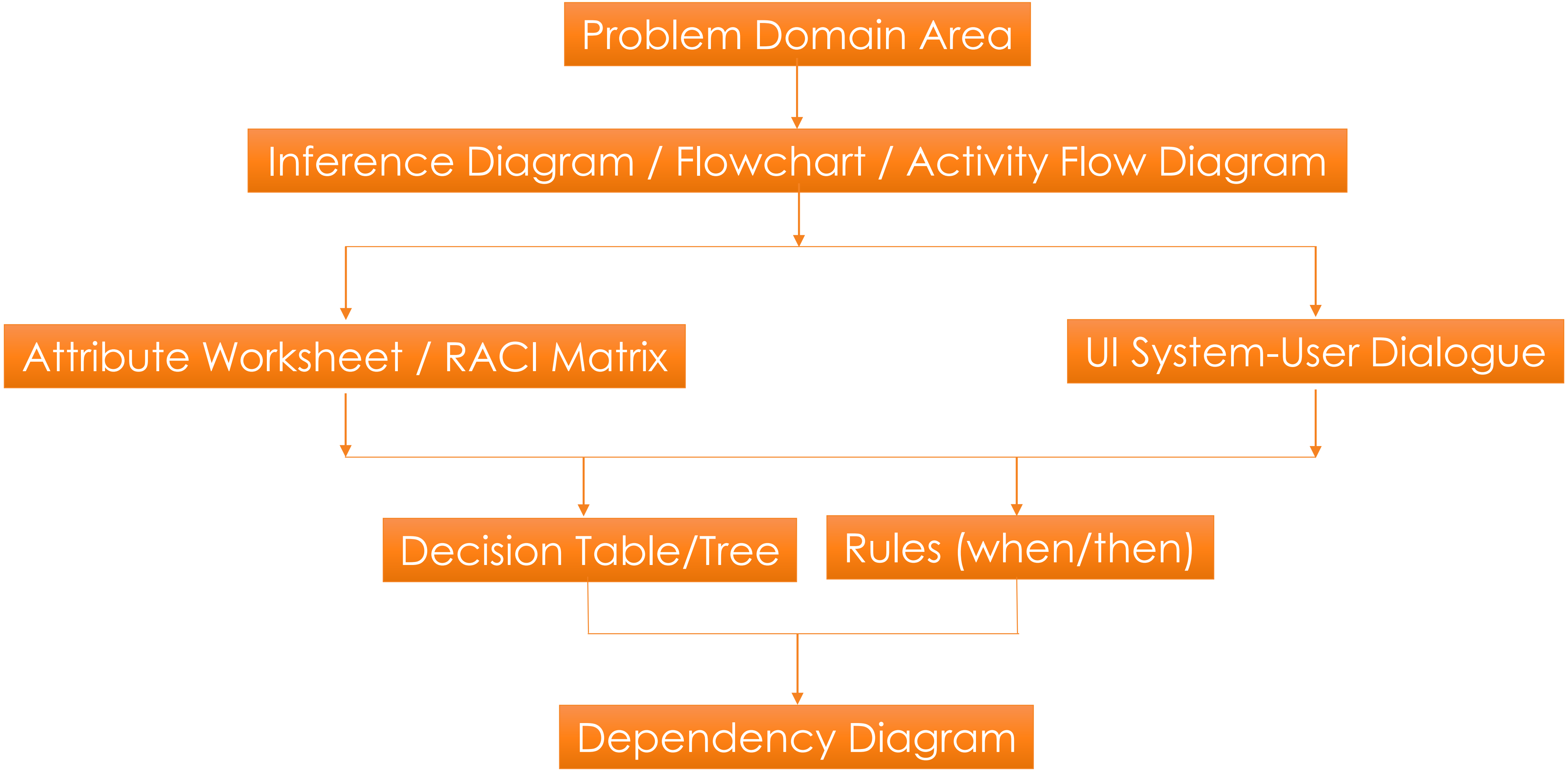
Process step / Role	Hold auditions	Select band members	Contract band members	Name band	Organise vocal tuition	Organise dance lessons	Stylise band	Find supporting musicians	Contract supporting musicians	Write songs	Book recording studio	Record backing vocals	Record vocals	Record backing vocals	Mix tracks	Shoot video	Edit video	Design cover artwork	Draft credits	Review credits and cover artwork	Compile album and DVD
Note	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Talent Scout	AR	AR	R					C													
Legal Advisor			AR						AR											R	
Band Member				I	I	I	I				I		R			R				R	
Record Producer			C	AR			C	AR	R	C	AR	AR	AR	AR	AR	R	C	R	AR	R	AR
Artist Development				C	AR	AR	AR			C								C		R	
Songwriter										AR										R	
Musician											I	R		R						R	
Video Production																AR	AR			R	R
Artist																		AR		R	

Key	
R - Responsible	Actually completes the activity - responsibility can be shared. Degree of responsibility is determined by the "A".
A - Accountable	Has Yes/No authority - there can only be one "A" per activity
C - Consulted	Involved prior to decision or action - two-way communication
I - Informed	Needs to know of the decision or action - one-way communication.

- **Responsible; Accountable; Consulted; Informed**
  - **R** for responsible means, "the person who actually does the activity". Responsibility for an activity can be shared, if necessary.
  - **A** for accountable means, "the buck stops here", and the role has ultimate yes/no authority.
  - **C** for consulted means, "kept in the loop", and implies two-way communication prior to the activity.
  - **I** for informed means, "kept in the picture", and implies one-way communication after the activity.
- **Best Practices:**
  - There can only be one accountability (**A**) per activity.
  - Recommend one responsibility (**R**) only per activity.
  - Roles can combine both accountability and responsibility for activities.
  - Minimize the number of consults (**C**) and informs (**I**).
  - Authority must accompany accountability.
  - Don't map decision points on the RACI matrix, only activities.

2.2 KNOWLEDGE MODELS (ACQUIRED → REPRESENTED)

Design Process



Problem Domain Area

Inference Diagram / Flowchart / Activity Flow Diagram

Attribute Worksheet / RACI Matrix

UI System-User Dialogue

Decision Table/Tree

Rules (when/then)

Dependency Diagram

Exercise 2.2

Exercise

- Refer to Airport Gate Assignment System (AGAS) case.
- Create relevant Knowledge Models.

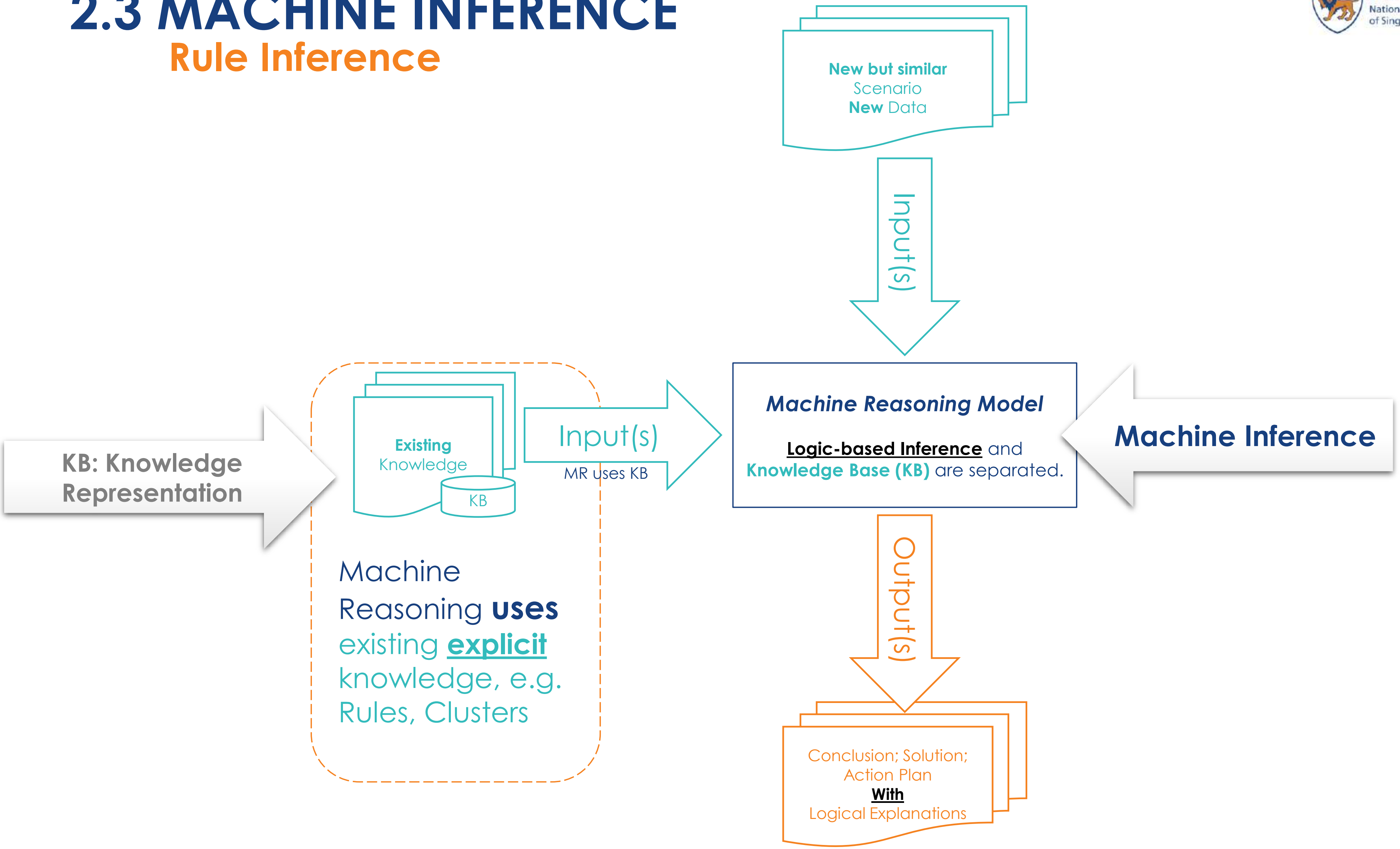
2.3

MACHINE INFERENCE

(PART 1)

2.3 MACHINE INFERENCE

Rule Inference



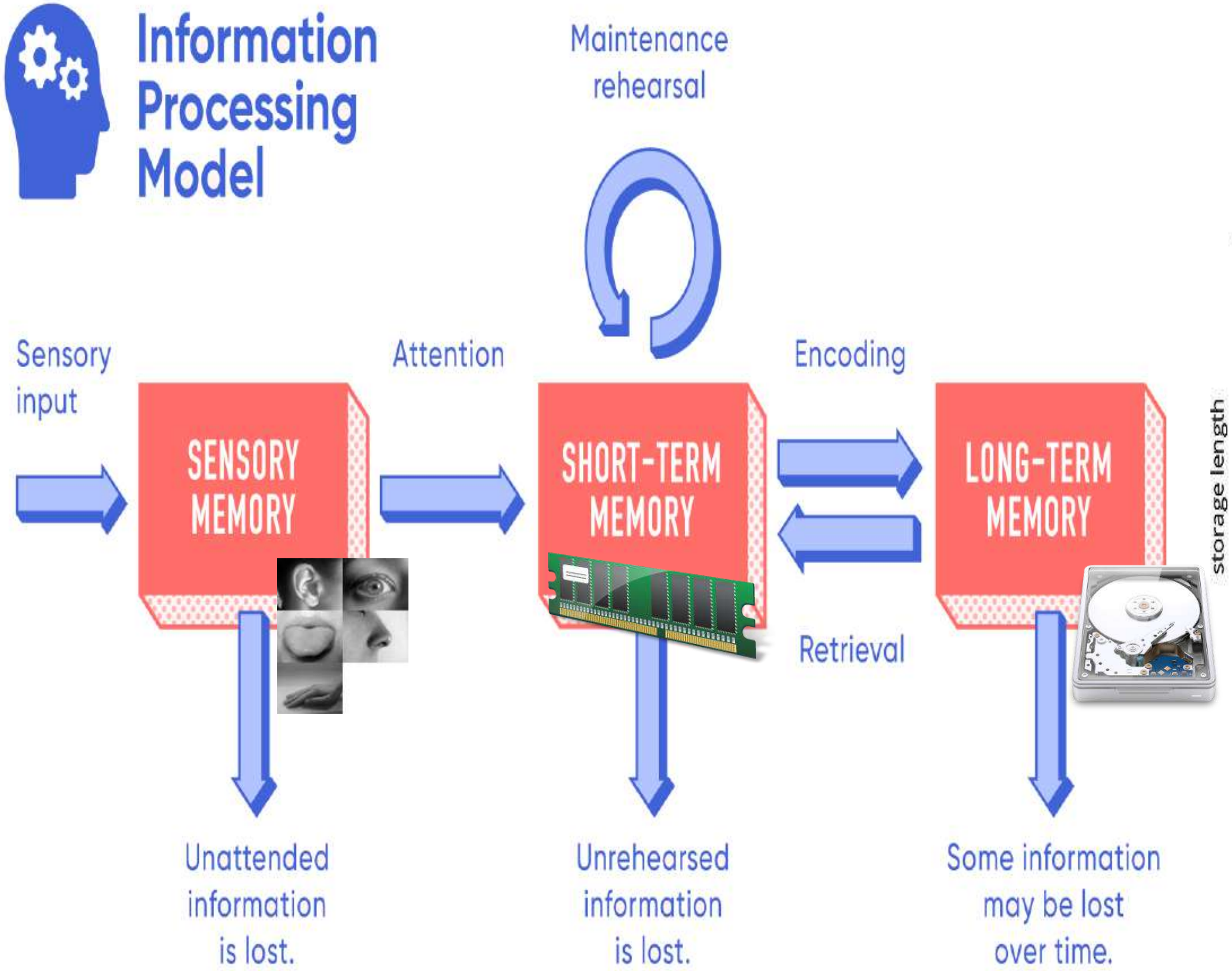
Machine Reasoning **uses** existing **explicit** knowledge, e.g. Rules, Clusters

Machine Reasoning Model

Logic-based Inference and **Knowledge Base (KB)** are separated.

2.3 MACHINE INFERENCE

[Psychology] Information Processing Theory



Information Processing Model

Maintenance rehearsal

Sensory input

Attention

Encoding

Retrieval

SENSORY MEMORY

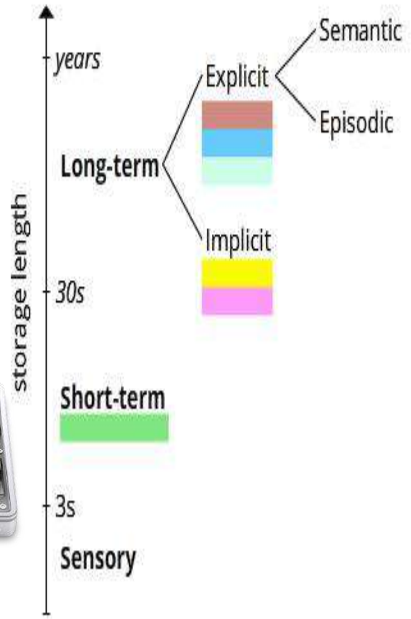
SHORT-TERM MEMORY

LONG-TERM MEMORY

Unattended information is lost.

Unrehearsed information is lost.

Some information may be lost over time.



storage length

years

30s

3s

Long-term

Short-term

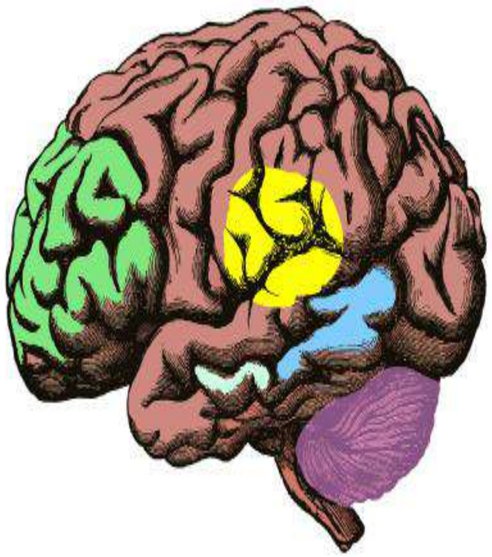
Sensory

Explicit

Implicit

Semantic

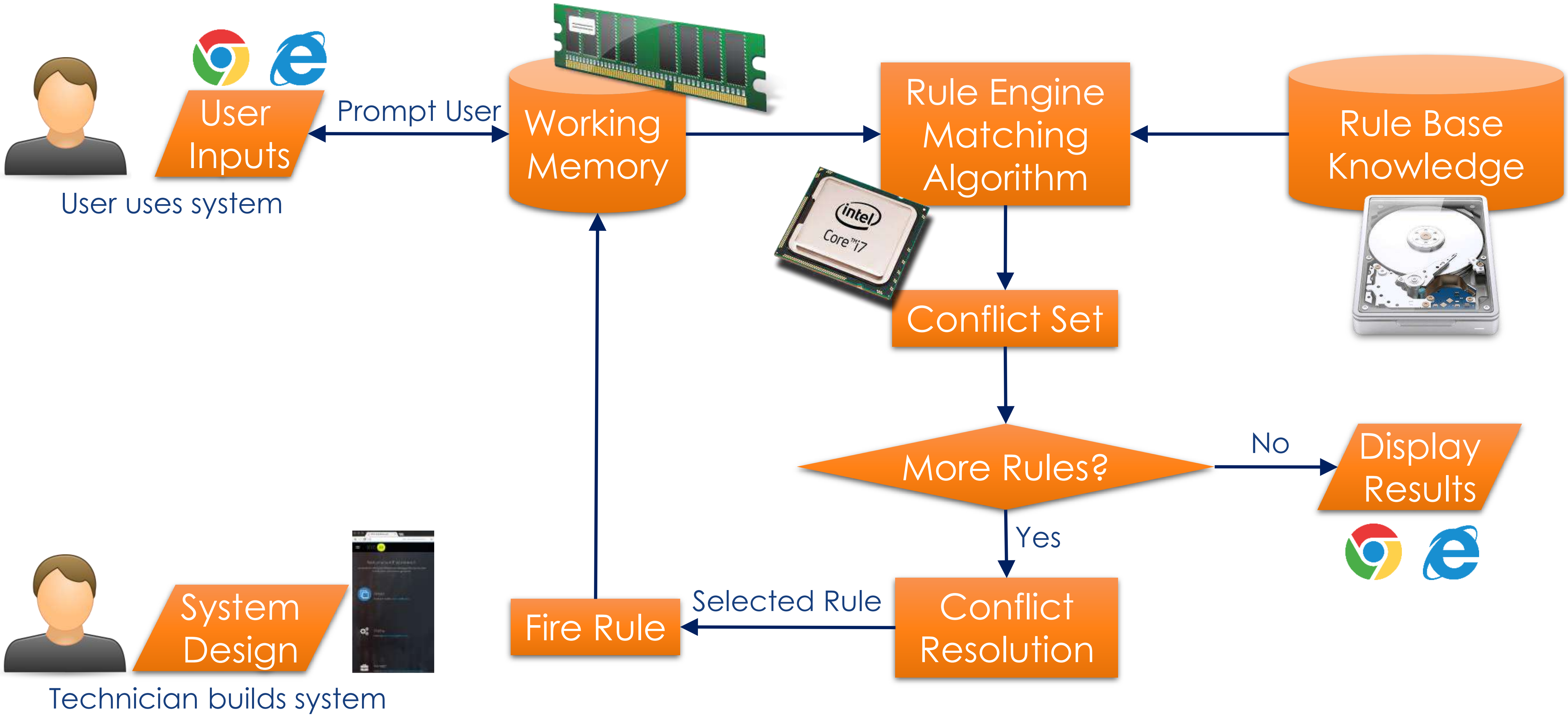
Episodic



Source https://www.learnupon.com/blog/what-is-information-processing-theory/

2.3 MACHINE INFERENCE

[AI] Recognise-Act Control Cycle



User Inputs

Prompt User

User uses system

Working Memory

Rule Engine Matching Algorithm

Rule Base Knowledge

Conflict Set

More Rules?

No

Display Results

Yes

Conflict Resolution

Selected Rule

Fire Rule

System Design

Technician builds system

2.3 MACHINE INFERENCE

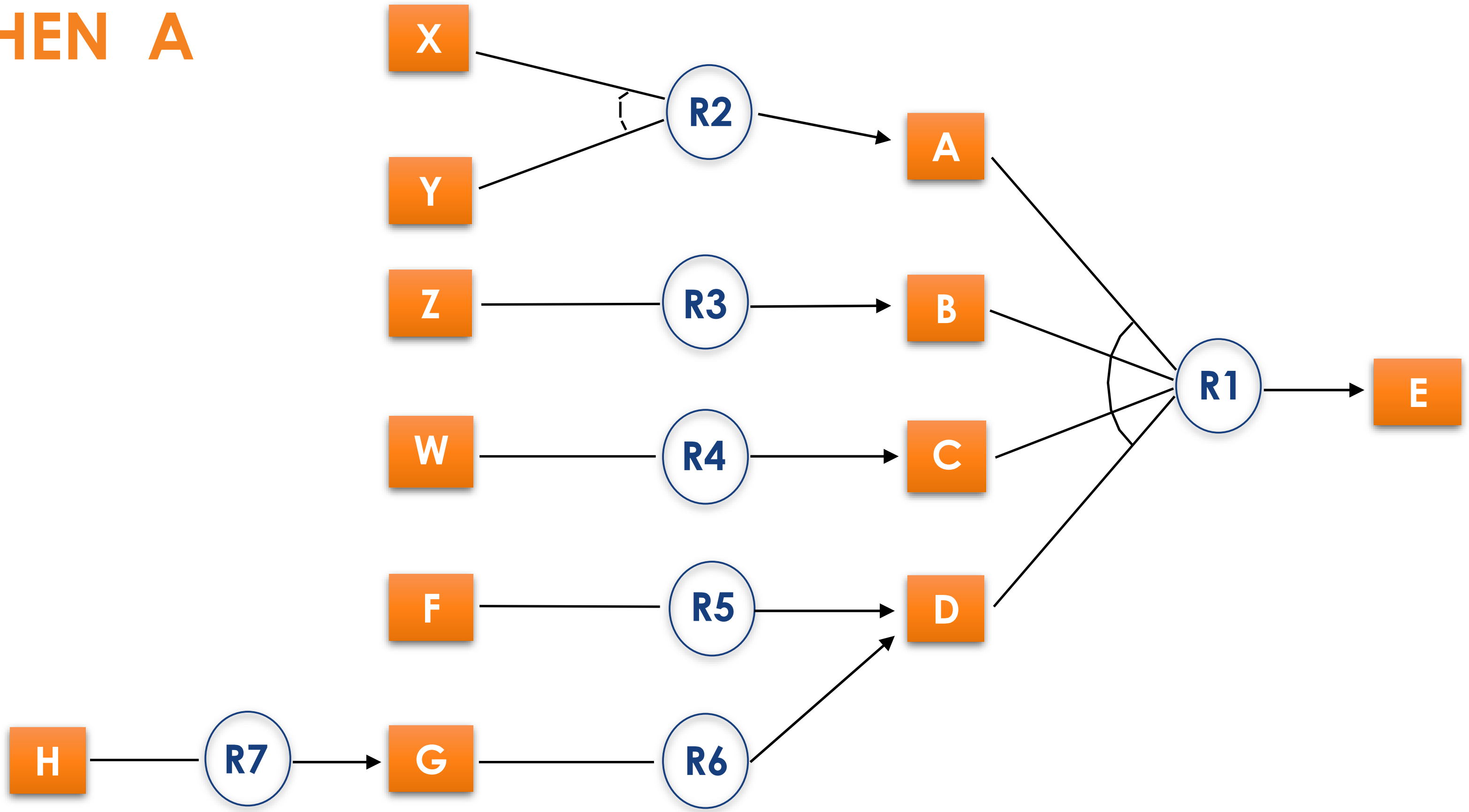
Three main components of Rule Based System (RBS)

- **A set of business rules (Knowledge Base)** **[long term memory]**
  - A rule represents a single chunk of problem-solving knowledge
- **A working memory (WM)** **[short term memory]**
  - Contains Rules and Data (current state of program execution)
- **A rule engine** **[pattern matching controller]**
  - A computational system that implements the control strategy and applies (fire) the rules
  - The patterns in WM are matched against the conditions of the rules
  - Matched rules are called the conflict set
  - The control strategy determines the order in which the rules are fired and resolves any rule conflicts
  - Uses the Recognise-Act cycle

2.3 MACHINE INFERENCE

Rules form a Search Tree

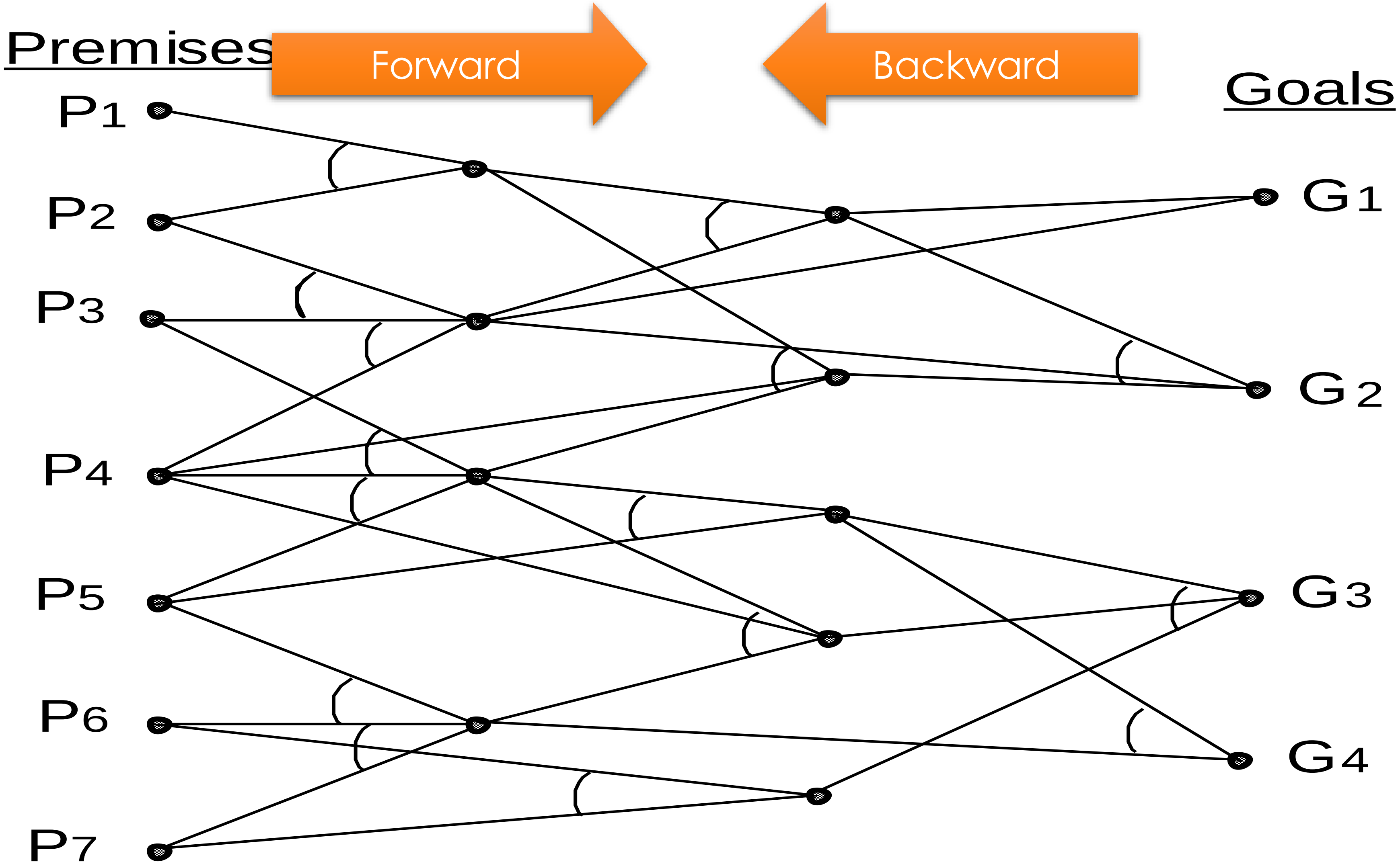
- **R1** IF A and B and C and D THEN E
- **R2** IF X and Y THEN A
- **R3** IF Z THEN B
- **R4** IF W THEN C
- **R5** IF F THEN D
- **R6** IF G THEN D
- **R7** IF H THEN G



- Inference strategy is known as “**chaining**”.

2.3 MACHINE INFERENCE

Forward Chaining & Backward Chaining



2.3 MACHINE INFERENCE

Forward Chaining Example

Rule 1:

IF the patient has a sore throat
AND we suspect a Bacterial infection
THEN we believe the Illness is a strep throat

Rule 2:

IF the patient's temperature is > 100
THEN the patient has a fever

Rule 3:

IF the patient has been sick for over a month
AND the patient has a fever
THEN we suspect a bacterial infection

Facts:
Temperature = 104°F (40°C) F1
Patient has been sick for 2 months F2
Patient has a sore throat F3

2.3 MACHINE INFERENCE

Forward Chaining Example

Rule 1:

IF the patient has a sore throat
AND we suspect a Bacterial infection
THEN we believe the Illness is a strep throat

Rule 2:

IF the patient's temperature is > 100
THEN the patient has a fever

Rule 3:

IF the patient has been sick for over a month
AND the patient has a fever
THEN we suspect a bacterial infection

Facts:	Temperature = 104°F (40°C)	F1 ←
	Patient has been sick for 2 months	F2
	Patient has a sore throat	F3
	patient has a fever	→ F4

2.3 MACHINE INFERENCE

Forward Chaining Example

Rule 1:

IF the patient has a sore throat
AND we suspect a Bacterial infection
THEN we believe the Illness is a strep throat

Rule 2:

IF the patient's temperature is > 100
THEN the patient has a fever

Rule 3:

IF the patient has been sick for over a month
AND the patient has a fever
THEN we suspect a bacterial infection

Facts:		
	Temperature = 104°F (40°C)	F1
	Patient has been sick for 2 months	F2
	Patient has a sore throat	F3
	patient has a fever	F4
	suspect a bacterial infection	F5

2.3 MACHINE INFERENCE

Forward Chaining Example

Rule 1:

IF the patient has a sore throat
AND we suspect a Bacterial infection
THEN we believe the illness is a strep throat

Rule 2:

IF the patient's temperature is > 100
THEN the patient has a fever

Rule 3:

IF the patient has been sick for over a month
AND the patient has a fever
THEN we suspect a bacterial infection

Facts:			
	Temperature = 104°F (40°C)	F1	
	Patient has been sick for 2 months	F2	
	Patient has a sore throat	F3	←
	patient has a fever	F4	
	suspect a bacterial infection	F5	←
	illness is a strep throat	→ F6	

2.3 MACHINE INFERENCE

Forward Chaining Example

Initial state of working memory WM: [F1, F2, F3]

Rule-2 [F1]

Add F4: “patient has a fever”

Rule-3 [F2, F4]

Add F5: “suspect a bacterial infection”

Rule-1 [F3, F5]

Add F6: “illness is a strep throat”

No more rules to fire → halt

Conclusion → illness is a strep throat

2.3 MACHINE INFERENCE

Forward Chaining Definition

- **A forward chaining system**
  - Begin with a set of facts in the Working Memory, then apply rules to generate new facts until the desired goal is reached.
- **Rules whose premise (IF..) is known to be true are fired, and their conclusions (THEN..) are declared true.**
  - This process continues until no more rules can be triggered/fired. The system then reports its conclusions.

2.3 MACHINE INFERENCE

Backward Chaining Example

Rule 1:

IF the patient has a sore throat
AND we suspect a Bacterial infection
THEN we believe the illness is a strep throat

Rule 2:

IF the patient's temperature is > 100°F
THEN the patient has a fever

Rule 3:

IF the patient has been sick for over a month
AND the patient has a fever
THEN we suspect a bacterial infection

F1: Temperature = 104°F (40°C)

F2: Patient has been sick for 2 months

F3: Patient has a sore throat

F4: Is the illness a strep throat?

goal / hypothesis to prove

2.3 MACHINE INFERENCE

Backward Chaining Example

Rule 1:

IF the patient has a sore throat
AND we suspect a Bacterial infection
THEN we believe the illness is a strep throat

Rule 2:

IF the patient's temperature is > 100°F
THEN the patient has a fever

Rule 3:

IF the patient has been sick for over a month
AND the patient has a fever
THEN we suspect a bacterial infection

F1: Temperature = 104°F (40°C)

F2: Patient has been sick for 2 months

F3: Patient has a sore throat

F4: Is the illness a strep throat?

2.3 MACHINE INFERENCE

Backward Chaining Example

Rule 1:

IF the patient has a sore throat ☑
AND we suspect a Bacterial infection [?]
THEN we believe the illness is a strep throat

Rule 2:

IF the patient's temperature is > 100°F
THEN the patient has a fever

Rule 3:

IF the patient has been sick for over a month
AND the patient has a fever
THEN we suspect a bacterial infection

F1: Temperature = 104°F (40°C)
F2: Patient has been sick for 2 months
F3: Patient has a sore throat
F4: Is the illness a strep throat?

2.3 MACHINE INFERENCE

Backward Chaining Example

Rule 1:
IF the patient has a sore throat ☑
AND we suspect a Bacterial infection [?]
THEN we believe the illness is a strep throat

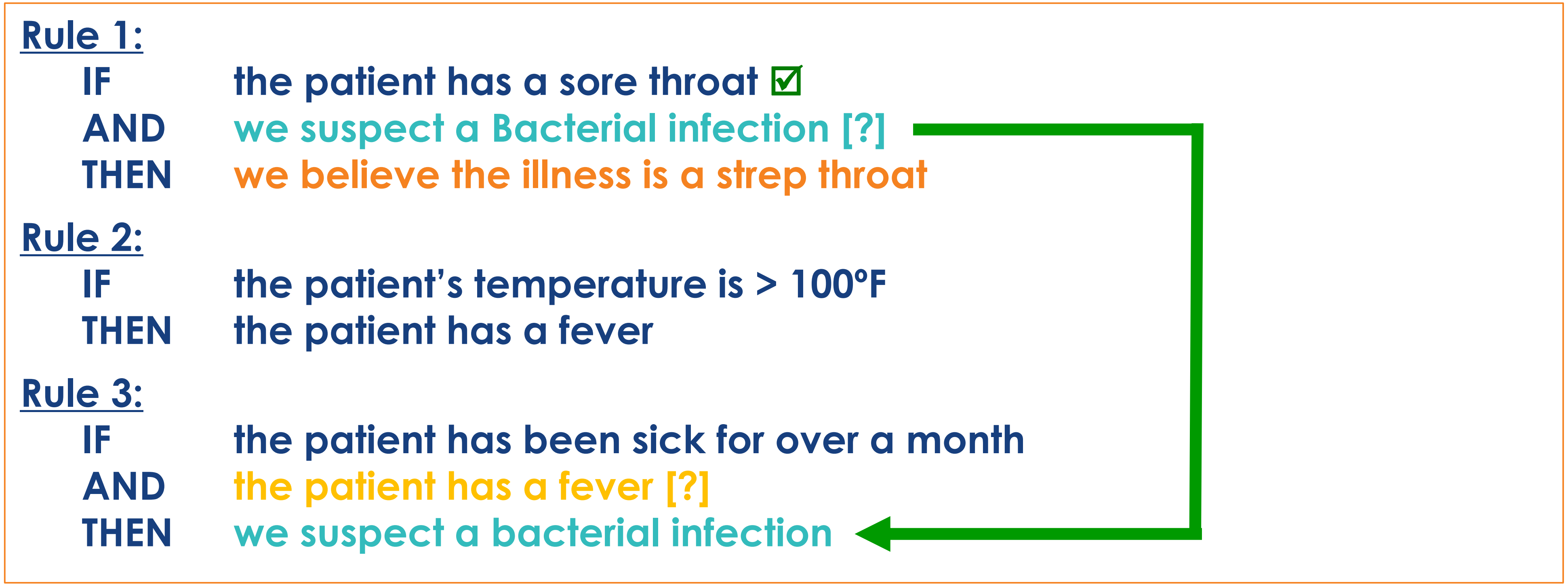
Rule 2:
IF the patient's temperature is > 100°F
THEN the patient has a fever

Rule 3:
IF the patient has been sick for over a month
AND the patient has a fever
THEN we suspect a bacterial infection

F1: Temperature = 104°F (40°C)
F2: Patient has been sick for 2 months
F3: Patient has a sore throat
F4: Is the illness a strep throat?
F5: Do we suspect a bacterial infection?

2.3 MACHINE INFERENCE

Backward Chaining Example



Rule 1:

IF the patient has a sore throat ☑

AND we suspect a Bacterial infection [?]

THEN we believe the illness is a strep throat

Rule 2:

IF the patient's temperature is > 100°F

THEN the patient has a fever

Rule 3:

IF the patient has been sick for over a month

AND the patient has a fever [?]

THEN we suspect a bacterial infection

F1: Temperature = 104°F (40°C)

F2: Patient has been sick for 2 months

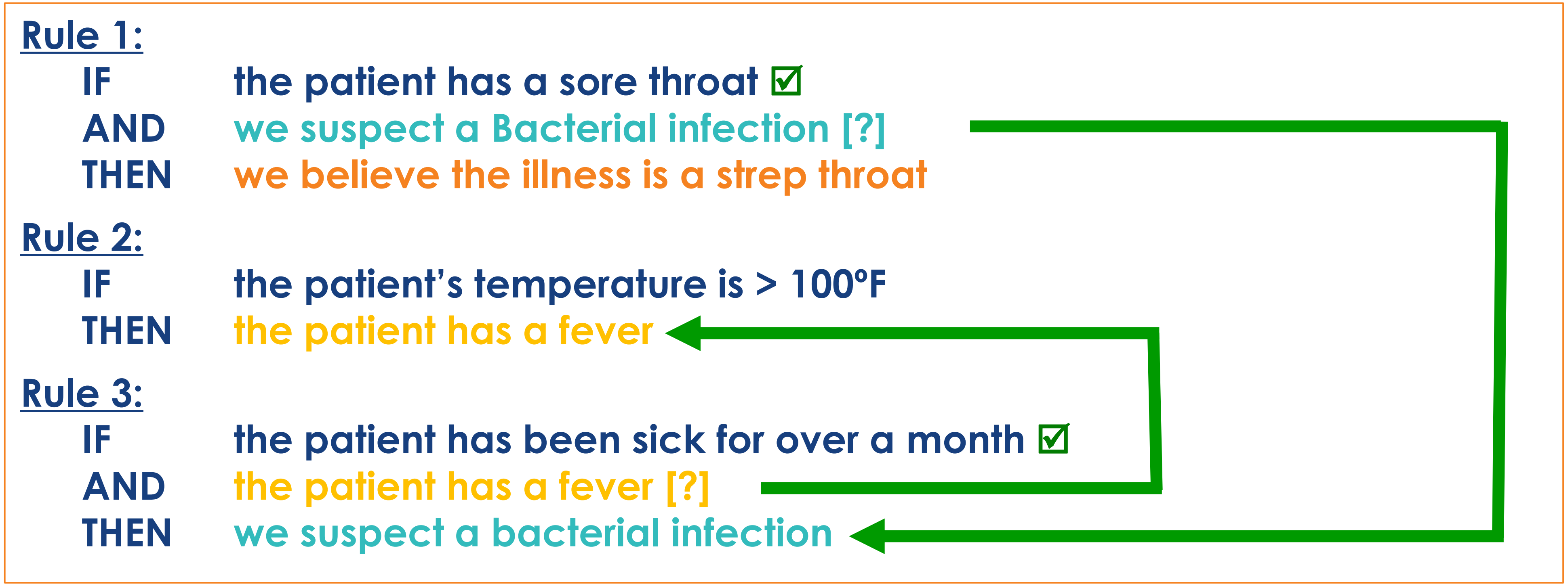
F3: Patient has a sore throat

F4: Is the illness a strep throat?

F5: Do we suspect a bacterial infection?

2.3 MACHINE INFERENCE

Backward Chaining Example



Rule 1:
IF the patient has a sore throat ☑
AND we suspect a Bacterial infection [?]
THEN we believe the illness is a strep throat

Rule 2:
IF the patient's temperature is > 100°F
THEN the patient has a fever

Rule 3:
IF the patient has been sick for over a month ☑
AND the patient has a fever [?]
THEN we suspect a bacterial infection

F1: Temperature = 104°F (40°C)

F2: Patient has been sick for 2 months

F3: Patient has a sore throat

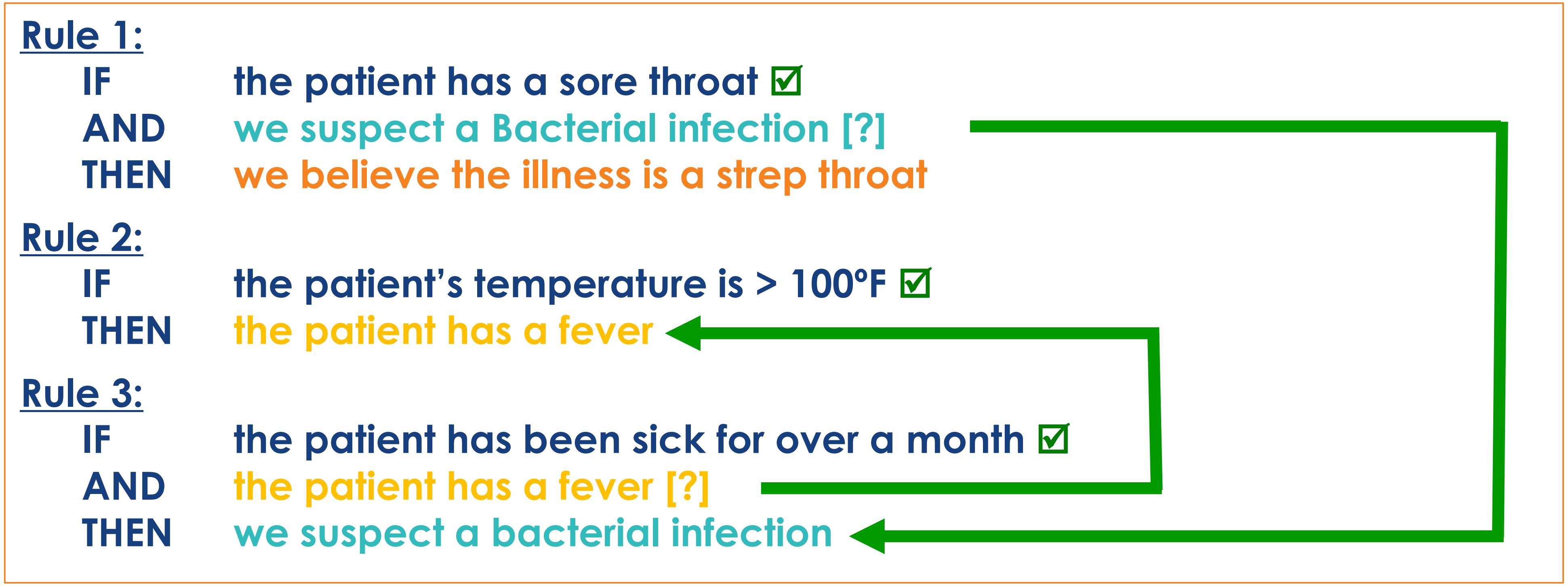
F4: Is the illness a strep throat?

F5: Do we suspect a bacterial infection?

F6: Does the patient have a fever?

2.3 MACHINE INFERENCE

Backward Chaining Example



Rule 1:

IF the patient has a sore throat ☑

AND we suspect a Bacterial infection [?]

THEN we believe the illness is a strep throat

Rule 2:

IF the patient’s temperature is > 100°F ☑

THEN the patient has a fever

Rule 3:

IF the patient has been sick for over a month ☑

AND the patient has a fever [?]

THEN we suspect a bacterial infection

F1: Temperature = 104°F (40°C)

F2: Patient has been sick for 2 months

F3: Patient has a sore throat

F4: Is the illness a strep throat?

F5: Do we suspect a bacterial infection?

F6: Does the patient have a fever?

Proved: Patient has a strep throat.

2.3 MACHINE INFERENCE

Backward Chaining Example

- Initial state of working memory WM: [Facts + Hypothesis]

Rule-1 [Hypothesis F4]

Goal: we suspect a strep throat? {new goal to pursue **Add F4**}

Check: patient has a sore throat? {proved by **F3**}

Check: we suspect bacterial infection? {new goal to pursue **Add F5**}

Rule-3 [F5]

Check: patient has been sick for over a week? {proved by **F2**}

Check: patient has a fever? {new goal to pursue **Add F6**}

Rule-2 [F6]

Check: patient’s temperature is > 100°F (37.8°C)? {proved by **F1**}

No more rules to fire {all proved} → halt

Conclusion → yes, illness is a strep throat

2.3 MACHINE INFERENCE

Backward Chaining Definition

- **A backward chaining inference engine starts from a goal or hypothesis.**
  - It works through the rules trying to match the goal with the action clauses (THEN part) of a rule.
  - When a match is found, the condition clauses (IF part) of the matching rule become "sub-goals".
  - The cycle is repeated until a verifiable set of condition clauses is found.

2.3 MACHINE INFERENCE

Forward Chaining vs. Backward Chaining

Forward Chaining	Backward Chaining
Planning, monitoring, surveillance, control, decision	Diagnosis, trouble-shooting
Present to future Antecedent to consequent	Present to past Consequent to antecedent
Data driven, bottom-up reasoning	Goal driven, top-down reasoning
Work forward to find what solutions that follow from the facts	Work backwards to find facts that support a given hypothesis
Facilitates Breadth-First-Search	Facilitates Depth-First-Search
Does not facilitate Explanation	Facilitates Explanation
CLIPS, KIE Drools	PROLOG, KIE Drools

2.3 MACHINE INFERENCE

Forward Chaining vs. Backward Chaining

- FC is data-driven
  - The focus of attention starts form known data & business rules
    - e.g., object recognition, routine decisions
    - May do lots of work that is irrelevant to the goal
- BC is goal-driven
  - Appropriate for problem-solving & investigation
    - e.g., Where are my keys? How do I get into a PhD program?

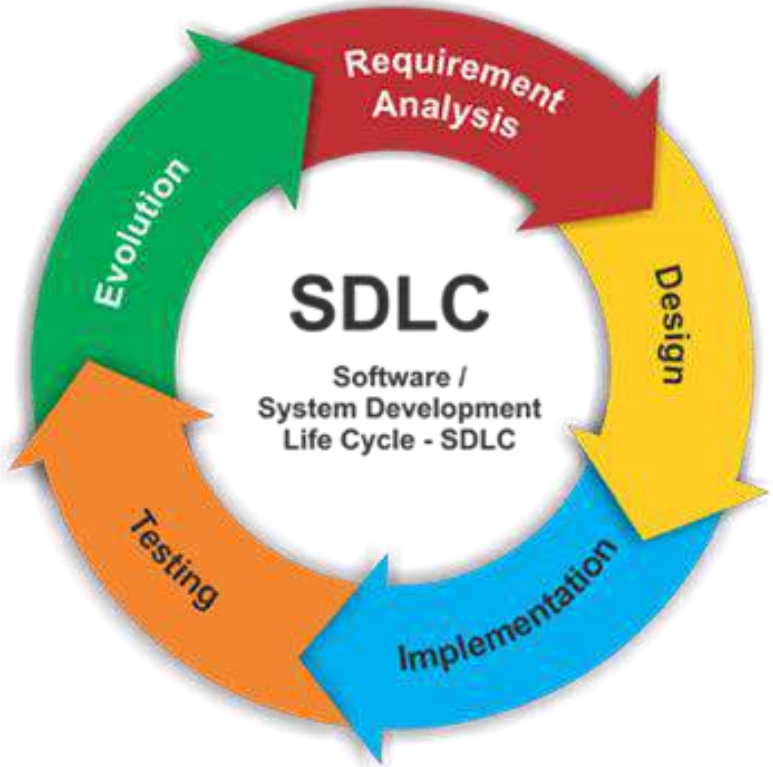
☺ Computer memory consumption of BC can be much less than FC when knowledge base is large.

2.4 WORKSHOP

KNOWLEDGE MODELLING

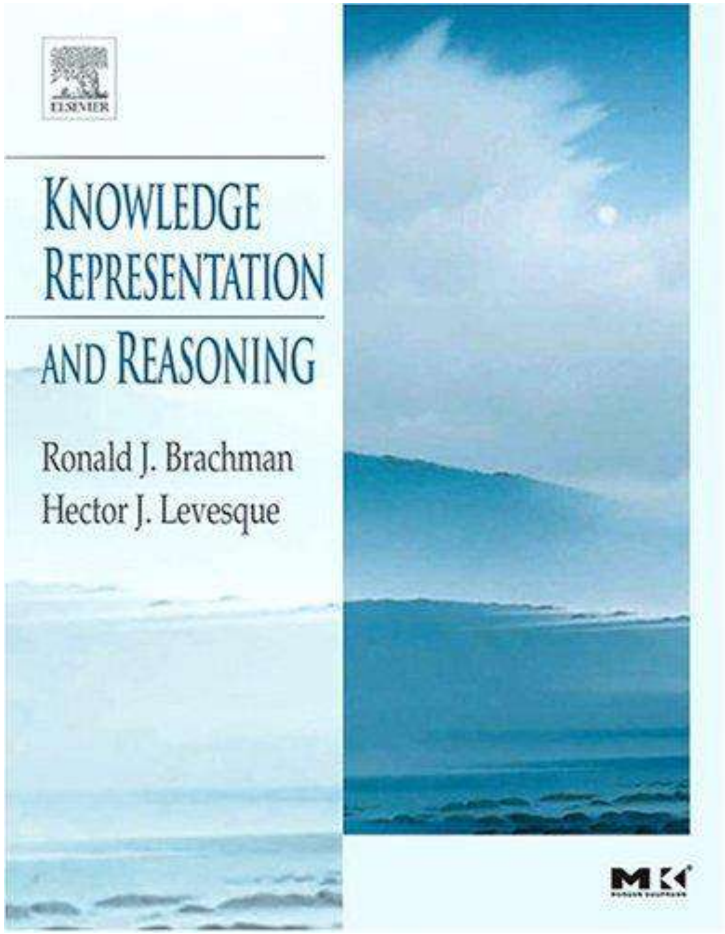
2.4 WORKSHOP KNOWLEDGE MODELLING

- **Requirement Analysis**
  - Problem selection: Identify business value and purposes
- **Design (Knowledge Representation and Acquisition)**
  - Knowledge acquisition, interviews
  - Definition of problem domain: Draw high level Inference Diagram
  - System design: Compose other relevant Knowledge Models
- **Implementation (KIE Development)**
  - System development: KIE tools
- **Testing**
  - Integrate, test, revise, deploy, and use
- **Evolution**



https://i1.wp.com/melsatar.blog/wp-content/uploads/2012/03/sdlc.png?fit=830%2C374&ssl=1

DAY 2 REFERENCE



1. Designing a decision service using guided rules

   https://access.redhat.com/documentation/en-us/red_hat_decision_manager/7.2/html-single/designing_a_decision_service_using_guided_rules/

2. KIE Workbench Tutorial : Human Machine Interaction

   https://www.youtube.com/watch?v=NfNnUsr66Cc

3. KIE Workbench Tutorial : Guided Decision Tables

   https://www.youtube.com/watch?v=qBgxVoc2qfw

4. Jay Pujara & Sameer Singh. (2018). Mining Knowledge Graphs from Text

   https://kgtutorial.github.io/

DAY 2 SUMMARY

2.1 Knowledge Acquisition (Business Rules)

- Documented sources vs. Undocumented sources
- Knowledge capture; Knowledge elicitation

2.2 Knowledge Models (Acquired → Represented)

- Concept Dictionary; Concept Tree; Composition Tree; Decision Tree; Goal Reduction Tree; Inference Diagram; Attribute Worksheet ; HMI/UI System-User Dialogue; Rules & Decision Table; Dependency Diagram; Flowchart; Activity Flow Diagram; RACI Matrix

2.3 Machine Inference (part 1)

- Knowledge/Rule Base; Inference Engine; Working Memory
- Forward Chaining & Backward Chaining

2.4 Knowledge Modelling Workshop

END OF LECTURE NOTES