KCA UNIVERSITY

SCHOOL OF TECHNOLOGY

BIT 03206 – ARTIFICIAL INTELLIGENCE

ASSIGNMENT TWO

DATE: JULY, 2025

🌐 Domain: Restaurant Management System

1. Types of Knowledge to Represent

To effectively manage a restaurant, we need to represent the following categories of knowledge:

a. Declarative Knowledge (Facts)

- **Menu items**: Names, categories (appetizer, main course, dessert, beverage), ingredients, dietary tags (vegan, gluten-free), prices.
- **Staff**: Roles (chef, waiter, manager), schedules, contact info.
- **Tables**: Table IDs, capacity, location (indoor/outdoor).
- **Customers**: Profiles, preferences, reservation history.
- **Reservations**: Date/time, table assignment, customer, special requests.

b. Procedural Knowledge

- **Order workflow**: Taking orders → kitchen processing → delivery to table → billing.
- **Reservation management**: Check availability → assign table → confirm.
- **Inventory management**: Update stock → notify when low → reorder.

c. Heuristic Knowledge

- Preferred seating assignments based on customer profile.
- Time required for preparing specific dishes.
- Rules for optimal staff scheduling.

d. Meta-Knowledge

- Confidence in the freshness of ingredients.

- Reliability of suppliers.

2. Method of Knowledge Representation

Hybrid Approach:

a. Ontology-Based Representation (using OWL/RDF)

- Build a formal ontology for:
  - Classes: `MenuItem`, `Ingredient`, `Table`, `Reservation`, `Customer`, `Staff`
  - Properties: `hasIngredient`, `isAvailableAtTime`, `servedBy`, `hasPreference`, etc.
  - Hierarchies: `MenuItem` subclassed into `Appetizer`, `MainCourse`, etc.

b. Rule-Based System (using Production Rules)

- Example:
- `IF reservation_time is during peak_hours AND party_size > 4`
- `THEN assign a waiter with at least 2 years of experience.`

c. Frames (for Procedural Knowledge)

- A `Reservation` frame:
- `Reservation:`
- `  - ID`
- `  - Customer`
- `  - TimeSlot`
- `  - TableAssigned`
- `  - SpecialRequests`
- `  - Status (Confirmed/Cancelled)`

3. Reasoning Capabilities

a. Inference

- Deduce available tables for a given time slot.
- Infer dietary options for a customer based on previous orders.

b. Consistency Checking

- Ensure a table isn't double-booked.
- Validate ingredient availability for ordered dishes.

c. Classification

- Automatically classify new dishes into categories based on ingredients.
- Classify customer types (e.g., frequent diner, first-time visitor).

d. Temporal Reasoning

- Understand past vs. future reservations.
- Plan staff schedules over shifts and days.

e. Recommendation

- Suggest dishes based on customer preferences or allergies.
- Recommend reservations based on historical patterns.

4. Implementation Stack (Example)

Component	Technology/Tool
Ontology	OWL, Protégé
Inference Engine	SWRL, Apache Jena, Pellet
Rule Engine	Drools or CLIPS
Database	PostgreSQL or Graph DB (Neo4j)
UI/API	React + RESTful APIs

Example Scenario

User query: "Can I reserve a table for 4 people at 7 PM tonight, and can you recommend a vegan meal?"

System reasoning:

- Uses temporal reasoning to check table availability at 7 PM.
- Uses inference to filter menu items with vegan dietary tag.
- Uses rule-based logic to assign a waiter based on party size and time.

Response:

“Yes, a table for 4 is available at 7 PM. We recommend our ‘Grilled Tofu Bowl’ and ‘Vegan Chocolate Cake’.”