AGENTIC AI AND IBM WATSONX ORCHESTRATE USAGE IN MENUPILOT

OVERVIEW OF AGENTIC AI IMPLEMENTATION

MenuPilot uses agentic AI through a multi-step workflow orchestrated in IBM watsonx Orchestrate. The solution is implemented as an agentic flow that plans and executes actions rather than responding with a single LLM prompt. In Orchestrate Flow Builder, a User Activity collects the user's request (weekly action plan, quick summary, or deep diagnostics). A Branch/Condition step routes the request to the appropriate path, which acts as the planning layer of the agentic system.

MenuPilot then uses a deterministic computation step (code/tool action) to enrich the menu dataset by generating derived columns such as food_cost_pct, margin, popularity/profit labels, quadrant classification (Stars/Plowhorses/Puzzles/Dogs), and operational/quality flags (for example: cost_warning, slow_prep_flag, low_rating_flag, complaints_flag). This tool step ensures numerical logic is computed reliably and reduces hallucination risk.

Finally, a Generative Prompt in Orchestrate consumes the enriched dataset and produces structured outputs: Output A tables, quadrant counts, top/bottom item rankings, and a prioritized weekly action plan with specific recommendations and expected impact. The agentic AI component is the combination of intent routing, tool execution (data enrichment), and LLM-driven reasoning to convert computed metrics into actionable decisions. IBM watsonx Orchestrate is used as the orchestration layer that connects these steps end-to-end, controls execution paths, and delivers the final user-facing outputs in a repeatable workflow.

ARCHITECTURE: FLOW-BASED AGENTIC SYSTEM

The solution is built as a **custom watsonx Orchestrate Flow** named "MenuPilot Flow Assistant" that implements multi-agent routing and intelligent decision-making.

AGENT COMPONENTS

	Agent 1: Intake & Routing Agent	**Agent 2: Conditional Routing Agent**	**Agent 3: Weekly Action Plan Agent**	**Agent 4: Quick Summary Agent**	**Agent 5: Deep Diagnostics Agent**
Purpose	User activity node that collects user intent and context	Branch node that autonomously routes requests to appropriate analysis agent	Generative AI prompt node connected to IBM watsonx foundation model	Uses complex multi-factor scoring	Generative AI prompt node
Inputs Collected	`user_goal`: Determines analysis type (weekly_plan | quick_summary | deep_diagnostics) `focus_week`: Optional business context (e.g., "reduce food cost," "improve ratings")	Path 1: Routes to Weekly Action Plan agent if user_goal contains "weekly" or "weekly_plan" Path 2: Routes to Quick Summary agent if user_goal contains "summary" or "quick_summary" Path 3: Routes to Deep Diagnostics agent if user_goal contains "diagnostic" or "diagnostics" Path 4 (Default): Fallback to weekly plan if input doesn't match patterns	menupilot.csv (menu item-level performance data) complaints_flag OR low_rating_flag (customer impact) cost_warning / high food_cost_pct (financial risk) Dogs quadrant (low profit, low popularity) Plowhorses (high popularity, lower profit) Puzzles (high profit, low popularity)	Stars (high profit, high popularity) **Contextual Reasoning**: Weighs stated focus_week priorities against data-driven urgency **Impact Calculation**: Generates expected outcomes for each recommendation **Operational Specificity**: Translates analytics into actionable steps (price changes, recipe tweaks, staff training, menu positioning)	menupilot.csv

Agentic Behavior	Captures natural language intent and translates it into structured routing parameters	Makes autonomous routing decisions based on intent classification without explicit user instruction	Markdown table with 5 rows containing: priority_rank | item_name | menu_quadrant | issue_detected | recommended_action | expected_impact	**Autonomous Data Aggregation**: Calculates quadrant distributions across menu portfolio **Comparative Analysis**: Identifies top 3 performers (by units_sold) and bottom 3 (by margin) **Pattern Recognition**: Synthesizes trends into executive-level insights **Consolidated Recommendations**: Distills complex analysis into top 5 actionable items	**Multi-Dimensional Analysis**: Cross-references complaints, ratings, costs, prep times, margins **Root Cause Inference**: Determines underlying issues from symptom patterns **Severity Assessment**: Autonomously assigns Critical/High/Medium/Low severity levels **Evidence-Based Reasoning**: Cites specific field values and metrics to support conclusions **Correlation Detection**: Identifies relationships like "high cost items with quality issues".
				Quadrant distribution table (Stars | Plowhorses | Puzzles | Dogs counts) Top/bottom performers lists Top 5 action plan table	Problem List table: item_name | issue_type | evidence_fields | severity | recommended_fix Top 5 Actions table

HOW AGENTS WORK TOGETHER

Autonomous Workflow Execution

User initiates request via natural language in watsonx Orchestrate:

"I need a weekly plan focusing on reducing food costs"

Agent 1 (Intake) captures:

- user_goal: "weekly plan"
- focus_week: "reducing food costs"

Agent 2 (Router) analyzes user_goal:

- Detects keyword "weekly"
- Autonomously routes to Path 1 (Agent 3)

Agent 3 (Weekly Plan) executes:

- Retrieves menu data from knowledge source
- Applies prioritization algorithm with cost_warning emphasis

- Generates 5 specific recommendations
- Calculates expected financial impact
- Returns structured output

Output delivered to user through watsonx Orchestrate interface

Agent Collaboration Patterns

Sequential Processing: Agents 1 → 2 → (3, 4, or 5) → Output

Linear flow where each agent contributes specialized capability

No agent revisits decisions (efficient single-pass architecture)

Contextual Awareness: All analysis agents (3, 4, 5) access shared context:

- focus_week` parameter informs prioritization across all agents
- Knowledge source provides consistent data view
- Ensures coherent recommendations regardless of path

Autonomous Decision-Making: System requires no human intervention between agents:

- Router selects appropriate agent without confirmation
- Analysis agents determine their own analytical approach
- Prioritization happens algorithmically based on data + context
- Recommendations are generated autonomously based on patterns detected

Why This Is Truly Agentic AI

Traditional BI tools require users to specify:

- Which metrics to analyze
- How to calculate derived values
- What thresholds constitute problems
- How to prioritize multiple issues
- What actions to recommend

MenuPilot's agents autonomously:

- Determine which analysis is needed based on intent
- Select appropriate metrics to cross-reference
- Calculate composite scores and correlations
- Prioritize findings using weighted logic
- Generate operational recommendations
- Quantify expected outcomes
- Adapt to business context (focus_week).