ACADEMIC RESEARCH ONLINE AGENT
A HYBRID BDI APPROACH

BACKGROUND & OBJECTIVES

Manual academic searches are inefficient and time-consuming.

Objective 1: Automate research data retrieval through intelligent agents.

Objective 2: Apply BDI reasoning for goal-oriented decision making.

Objective 3: Ensure ethical, GDPR-compliant information processing.

SYSTEM ARCHITECTURE

Modular flow: Crawler → Parser → NLP Filter → BDI Reasoner → Storage → Testing.

Supports Agile development and independent testing.

Facilitates scalability and module substitution.

[Diagram Placeholder: System Flowchart]

BDI REASONING MODEL

Beliefs – known keywords (e.g., agent, BDI, NLP, data).

Desires – identify and store relevant academic papers.

Intentions – perform data cleaning, filtering, and storage.

Hybrid design combining reactive and deliberative reasoning.

PYTHON IMPLEMENTATION

Developed in Python 3.11 with NLTK, re, sqlite3.

Crawler gathers simulated research articles.

Parser cleans text and removes stopwords.

BDI Reasoner filters articles based on belief keywords.

Storage module saves data to SQLite (simulating PostgreSQL/MongoDB).

EXECUTION & TESTING

Program executes preprocessing, filtering, and reasoning.

Filtered Articles:

1. Hybrid BDI Framework for Academic Search Systems

2. Natural Language Models in Research Information Systems

3. Optimising Web Scraping with Python

Test Result: Database contains 3 stored articles – Passed

[Insert Screenshot: output1.png]

DATABASE RESULTS

Stored results viewed using DB Browser for SQLite.

Each record includes ID, Title, Abstract snippet.

Simulates hybrid PostgreSQL + MongoDB architecture.

[Insert Screenshot: output2.png]

ETHICAL & COMPLIANCE MEASURES

Applies privacy-by-design principle (Del-Real et al., 2024).

Uses synthetic data for safe testing.

Ensures GDPR compliance and academic ethics.

Prepares for secure API-based real-world deployment.

CHALLENGES & SOLUTIONS

Crawler limitations → solved with simulated datasets.

Library version conflicts → resolved via isolated sprints.

Text inconsistency → fixed using regex and normalisation.

Keyword optimisation → balanced precision and recall.

REFLECTION & FUTURE WORK

Integration ensured seamless communication among modules.

Key learning: combining theory with software practice.

Future: ML-based classifier, GUI interface, hybrid DB deployment.

As Wooldridge (2009) stated: agents should act rationally and explainably.

Thank you for listening.