Hybrid AI Certification Workflow

As described in IAC Abstract

Input

Goal:
To obtain data + Establish Context

Pre-Process

Goal:
To convert unstructured data to auditable facts and establish relationships between data for Knowledge Graphs

Process 1

Goal:
To set fundamentals for creating knowledge graphs by establishing relationships between components

Process 2

Goal:
To enable graphical representation, ensure compliance, and auditability

---

Basic Docs (including raw test data), process via LLMs

→ LLM for structuring different input documents such as: Requirements doc, test plan docs, verification and validation docs, specification docs, etc.

Questionnaire based on input doc to establish context

→ Ask a questionnaire for correction purposes. Example, verification of document type, authors, date, etc.

↓ Display confirmed data

Extract Data such as component type, document revision number, and progress level

→ Ask a questionnaire for correction purposes. Example, verification of document type, authors, date, etc.

Suggest top 3 relevant categories that the tests come under, and user will select the most relevant

Named Entity Recognition:
Identify system components involved, parameters set, units, test cases, risks involved

Define relationship between recognized

Extract progress / create reference progress report.
Eg: Anomaly Detected, Doc Generated

Knowledge Graphs created are stored in a graph data base?

Set rules for maintaining compliance using rule engines.

→ Must be able to represent these rules. Main purpose of rules is to identify violations in compliance

Generate decision tree

→ User can directly question logic of decision tree (via LLM); important for **back-tracking.** Helps add a layer of reliability in the automation process

---

Process 3

Goal:
To detect anomaly in component test or verification and predictive analysis to produce a PASS/FAIL result

Output

Goal:
Display final result while prioritizing traceability mapping, subsequently reliability

Anomaly Detection with ML, for component's test validation

Use historical data of parameters: Test coverage, Number of issues found, Design Maturity, etc to predict Test Pass/Fail

Section based or component based document generation as per global standards

Can display the decision tree on user request

Keywords

Adaptability of software: Rules and regulations may change over time. Software must be easy to use in such cases.
Context Setting via Questionnaire/MCQ: helps maintain a "User-in-the-loop" architecture
Knowledge Graph: for organizing data; A technical upper hand; and **maintains compliance** even during test validation
Decision Tree: Helps in back-tracking output. Must be displayed to user, for the "User-in-the-loop" architecture, as it helps with auditability
Hybrid AI: Includes LLMs as the interface, ML for anomaly detection and predictive analysis (certification PASS/FAIL detection)