Finding Best-fit Agent Practices in Industrial CPS Using Graph Databases

CHANDAN SHARMA
ROOPAK SINHA
PAULO LEITÃO
OVERVIEW

• Problem: rank and select best-fit Industrial Agent (IA) practices\(^1\).

• Approach:
  • Construct a graph database for storing practices.
  • Identify template query patterns for ranking, data extraction and selection.
  • Use MVC to construct a front-end for users.

• Contribution:
  • IASelect, a tool for selecting best-fit Industrial Agent (IA) practices in P2660.1.

---

1. P. Leitão et al 2017 “Common practices for integrating industrial agents and low level automation functions,”

chandan.sharma@aut.ac.nz
BACKGROUND

IEEE P2660.1: Recommended Practices on Industrial Agents
• Multi-agent systems in Industrial control systems
• Identify best-fit practices for a given context

Graph databases:
• Highly interconnected data
• Easier to visualise than relational databases
• Just as easy to query
RELATED LITERATURE

- Available tools (storage and recommendation):
  - [1] and [2] use relational and XML databases respectively.
  - [3] uses a graph database for data visualization.
  - [4] and [5] use graph databases in domains such as Chemistry and Biology.

- Novelty:
  - **IASelect** provides query **boiler-plates** [6].
  - Practitioners **do not** require expertise in querying graph databases (unlike [1-3]).
  - **IASelect** enforces **schema-based topological restrictions** which reduce the risk of **data corruption** (unlike [2]).

---

1. S. Morimoto, D. Horie, and J. Cheng, “A security requirement management database based on iso/iec 15408,” in International Conference on Computational Science and Its Applications. Springer, 2006, pp. 1–10.
2. S. Morimoto and J. Cheng, “A security specification library with a schemaless database,” in International Conference on Computational Science. Springer, 2007, pp. 890–893.
3. R. Loof and K. Pussinen, “Visualisation of requirements and their relations in embedded systems,” 2014.
4. R. J. Hall, C. W. Murray, and M. L. Verdonk, “The fragment network: A chemistry recommendation engine built using a graph database,” Journal of medicinal chemistry, vol. 60, no. 14, pp. 6440–6450, 2017.
5. S. M. Graves, E. R. Bergeman, and C. B. Lawrence, “Querying a genome database using graphs,” in Proceedings of the 3th International Conference on Bioinformatics and Genome Research, 1994.
6. R. Sinha, S. Patil, C. Pang, V. Vyatkin, and B. Dowdeswell, “Requirements engineering of industrial automation systems: Adapting the CESAR requirements meta model for safety-critical smart grid software,” in Industrial Electronics Society, IECON 2015-41st Annual Conference of the IEEE. IEEE, 2015, pp. 002 172–002 177.
APPRAOCH FOR DESIGNING IASelect

- P2660.1 RANKING DATASET
- REQUIREMENTS FOR IASelect
- GRAPH DATABASE DESIGN FOR P2660.1 DATASET
- ARCHITECTURE OF IASelect
- IASelect CLIENT INTERFACE
|                          | Need host agents | Energy | Factory automation | Building automation | Monitor | Control | Simulation | Scalability | Time behaviour | Reusability |
|--------------------------|-----------------|--------|--------------------|---------------------|---------|---------|------------|-------------|----------------|-------------|
| **Hybrid, Tightly**      |                 |        |                    |                     |         |         |            |             |                |             |
| #HT, 1                   |                 |        |                    |                     |         |         |            |             |                |             |
| API client: Java         | 0               | 3      | 5                   | 3                   | 4       | 1       | 2          | 2            | 3              | 3           |
| Channel: Modbus          |                 |        |                     |                     |         |         |            |             |                |             |
| **On-device, Tightly**   |                 |        |                    |                     |         |         |            |             |                |             |
| #OT, 1                   |                 |        |                    |                     |         |         |            |             |                |             |
| API client: Java         | 1               | 3      | 4                   | 3                   | 4       | 4       | 1          | 2            | 5              | 2           |
| Channel: Modbus          |                 |        |                     |                     |         |         |            |             |                |             |
| **Hybrid, Loosed**       |                 |        |                    |                     |         |         |            |             |                |             |
| #HL, 1                   |                 |        |                    |                     |         |         |            |             |                |             |
| API client: Apache Milo  | 0               | 4      | 5                   | 4                   | 5       | 1       | 5          | 4            | 3              | 5           |
| Channel: OPC-UA          |                 |        |                     |                     |         |         |            |             |                |             |
| #HL, 2                   |                 |        |                    |                     |         |         |            |             |                |             |
| API client: Apache Paho  | 0               | 3      | 4                   | 5                   | 5       | 1       | 5          | 4            | 2              | 5           |
| Channel: MQTT            |                 |        |                     |                     |         |         |            |             |                |             |
|                         |                 |        |                     |                     |         |         |            |             |                |             |
| Broker: Eclipse Mosquit  |                 |        |                     |                     |         |         |            |             |                |             |
| **On-Device, Loosed**    |                 |        |                    |                     |         |         |            |             |                |             |
| #OL, 1                   |                 |        |                    |                     |         |         |            |             |                |             |
| API client: Apache Milo  | 1               | 3      | 4                   | 4                   | 5       | 3       | 3          | 3            | 4              | 5           |
| Channel: OPC-UA          |                 |        |                     |                     |         |         |            |             |                |             |
| #OL, 2                   |                 |        |                    |                     |         |         |            |             |                |             |
| API client: Apache Paho  | 1               | 2      | 4                   | 4                   | 5       | 3       | 3          | 3            | 4              | 5           |
| Channel: MQTT            |                 |        |                     |                     |         |         |            |             |                |             |
|                         |                 |        |                     |                     |         |         |            |             |                |             |
| Broker: Eclipse Mosquit  |                 |        |                     |                     |         |         |            |             |                |             |

chandan.sharma@aut.ac.nz
### IA CHARACTERISTICS IDENTIFIED IN P2660.1

| Domain               | Function               | Maintenance                        | Performance Efficiency     |
|----------------------|------------------------|------------------------------------|-----------------------------|
| Factory Automation   | Monitoring             | Re-usability                       | Time behaviour              |
| Building Automation  | Control                | Capacity To Host                   | Scalability                 |
| Energy               | Simulation             | agents                             |                             |
# Scenario defined by the user:

| Context                  |  |
|--------------------------|--------------------------|
| Function?                | Control                  |
| Application domain?      | Factory automation       |
| Technology capable to host agents is available? | Yes                     |

| Weights for criteria     |  |
|--------------------------|--------------------------|
| Scalability              | 10%                      |
| Time behaviour           | 10%                      |
| Reusability              | 80%                      |
|                          | 100%                     |

# Results:

## Recommended interface practice

- **#OL, 1**

## Suggested technologies

### Details

| Hybrid, Tightly | Score | Score final |
|-----------------|-------|-------------|
| #HT, 1          | 2.9   | 0.6         |
| API client: Java Channel: Modbus |

| On-device, Tightly | Score | Score final |
|--------------------|-------|-------------|
| #OT, 1             | 2.3   | 1.8         |
| API client: Java Channel: Modbus |

| Hybrid, Loosed     | Score | Score final |
|--------------------|-------|-------------|
| #HL, 1             | 4.7   | 0.9         |
| API client: Apache Milo Channel: OPC-UA |
| #HL, 2             | 4.6   | 0.9         |
| API client: Apache Paho Channel: MQTT Broker: Eclipse Mosquito |

| On-Device, Loosed  | Score | Score final |
|--------------------|-------|-------------|
| #OL, 1             | 4.7   | 2.8         |
| API client: Apache Milo Channel: OPC-UA |
| #OL, 2             | 4.7   | 2.8         |
| API client: Apache Paho Channel: MQTT Broker: Eclipse Mosquito |

chandan.sharma@aut.ac.nz
GRAPH DB DESIGN FOR P2660.1 DATASET

GRAPH SCHEMA

GRAPH DATABASE
ARCHITECTURE OF IASelect
TECHNOLOGY STACK OF IA*Select

SERVER

Implemented in JAVA using Gradle and Maven.

Gradle → Spring Boot Plugin → Embedded TomCat Server

Maven → Neo4j Database

CLIENT

HTML 5

CSS 3

JAVASCRIPT

AJAX (XmlHttpRequest)
DEMO OF IASelect

| NAME | API CLIENT   | CHANNEL | FINAL SCORE |
|------|--------------|---------|-------------|
| HL.1 | Apache Milo | OPC-UA  | 4.6         |
| HL.2 | Apache Paho | MQTT    | 4.5         |
| OL.1 | Apache Milo | OPC-UA  | 4.5         |
| OL.2 | Apache Paho | MQTT    | 4.5         |
| HT.1 | Java         | Modbus  | 2.24        |
| OT.1 | Java         | Modbus  | 1.84        |
ARCHITECTURAL ADVANTAGES OF IASelect

Functional Suitability: Ability for Users to query the graph database to rank available practice.

Usability: Allow Users to enter information in an interactive manner, provide appropriate user error protection and present results clearly.

Availability: Accessible to multiple Users, present in different locations at same time.

Portability: Independent of Users’ computer configuration.
LIMITATIONS AND FUTURE WORK

• **Performance:** As database grows in size, loading the entire graph into RAM will result in slow query processing.
  • Databases are I/O bound so what would be the performance of GraphDBs?
  • This is an open question related to graph database research.

• **Future Work:** Add a Admin console for practitioners to perform CRUD operations.
Thank You!