FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

SEMANTIC SYSTEMS

Enabling Exploratory Search on Manufacturing Knowledge Graphs

Kevin Haller

VOILA @ ISWC 22

Co-Authors: Fajar J. Ekaputra, Marta Sabou and Florina Piroi

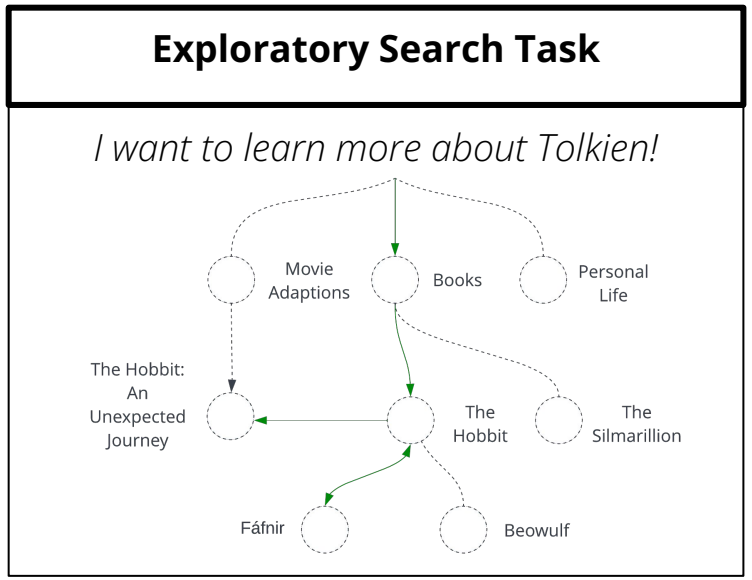
What is Exploratory Search?

Exploratory Search

Traditional search systems heavily rely on the popular query-response paradigm.

=> successful for well-defined information need with precise goal in mind.

Exploratory search systems enable information seeking tasks such as learning and investigating.

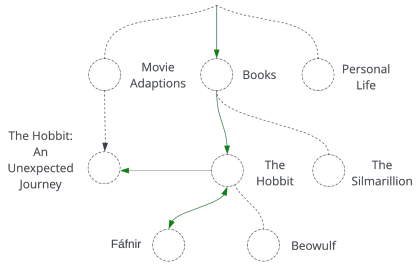


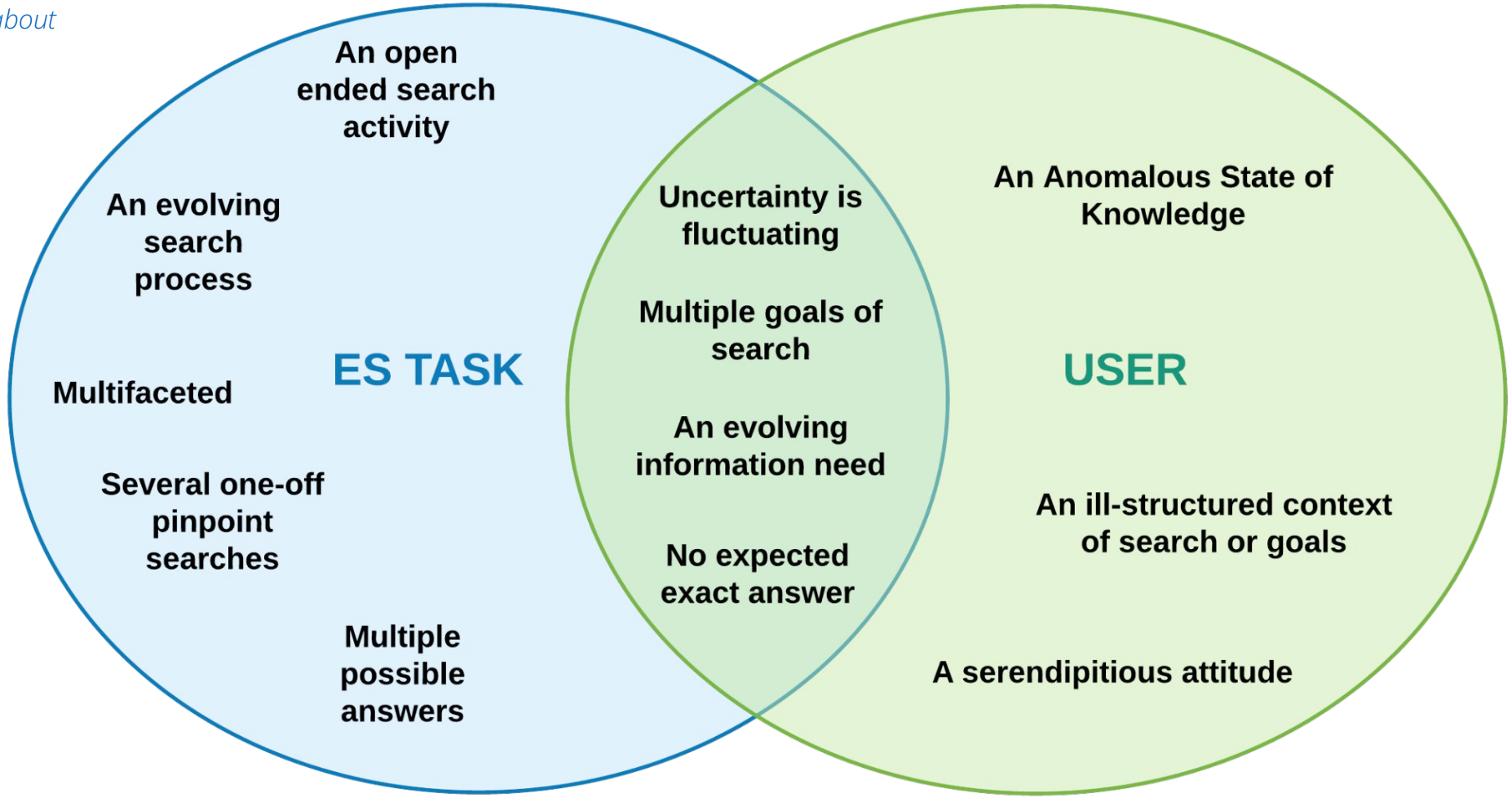
=> increased level of interaction between user and search system

=> more active engagement in search process

Exploratory Search Characteristics

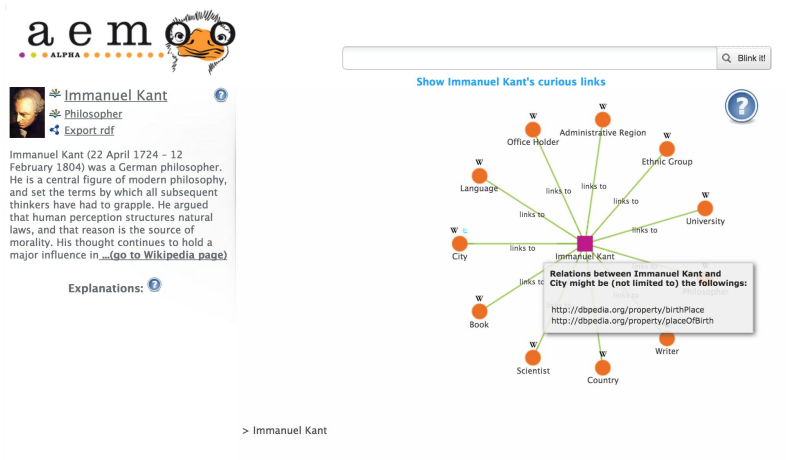
ES Task: I want to learn more about Tolkien!



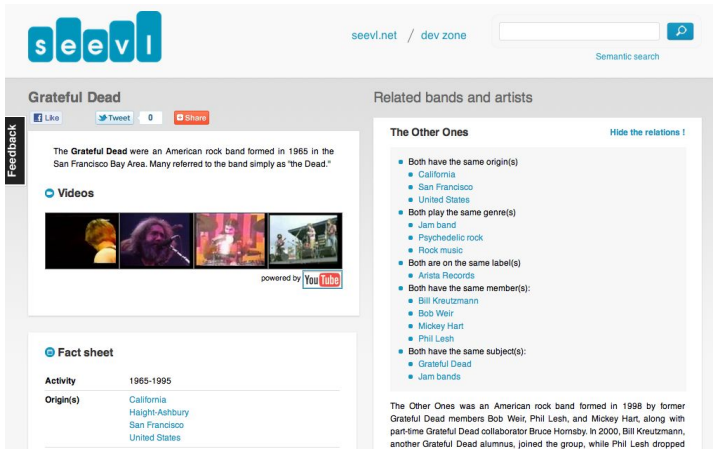


[Pal18] Evaluating exploratory search engines: designing a set of user-centered methods based on a modeling of the exploratory search process, Palagi Emelie, PhD Thesis, 2018

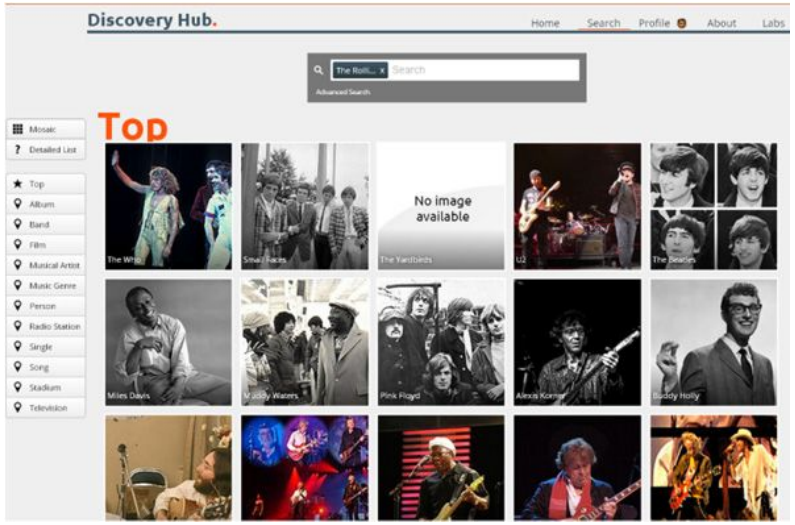
KG-based Exploratory Search Systems



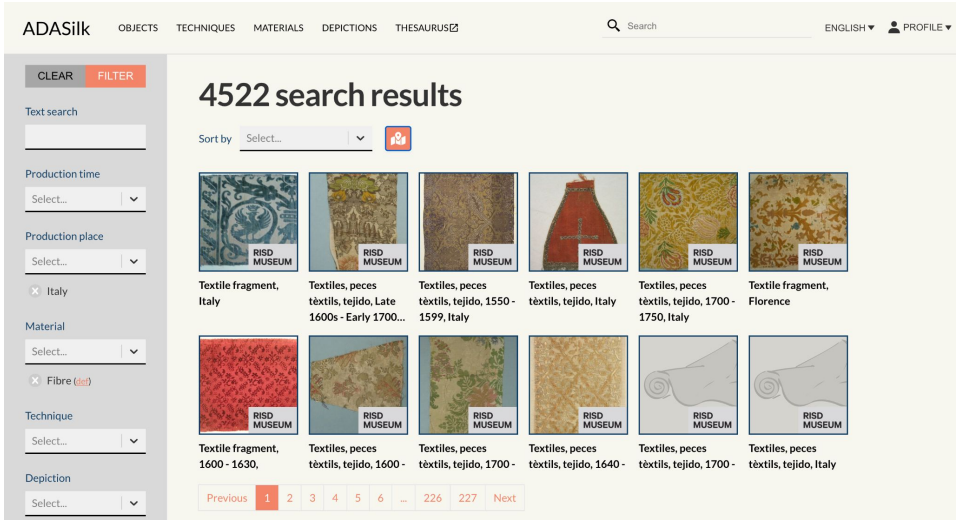
[Nuz17] **Aemoo**



[Pas12] **Seevl**



[Mar13] **Discovery Hub**

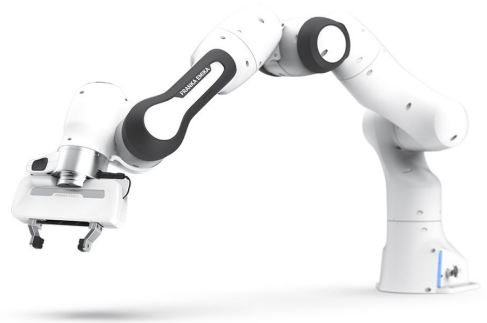


[Ehr21] **KG Explorer**

What about Manufacturing?

Use Cases

1) Pilotfabrik Aspern - Industrie 4.0



- hosts:
  - collaborative and industrial robotic arms
  - wide range of supporting tools (grippers, sensors, projectors, etc.)

Goal: increase usage degree of equipment by letting students & researchers learn about it.

* Pictures from Franka Emika website.

2) OntoTrans - H2020 EU Project

ONTOTRANS

- Material science and manufacturing domain
- Focus on the translator role

Goal: support in innovation challenge to improve a certain manufactured product in respect to several key performance indicators.

Requirement Gathering

Informal interviews with stakeholders:

- smart manufacturing researchers from industry and university
- translators to companies in OntoTrans
- simulation expert from a production plant manufacturer

=> requirements were aggregated.

<u>Special attention needed for:</u>

A) Multiperspectival exploration
=> Proposal: adaptive UI

B) Provenance visibility
=> Proposal: adaptive UI

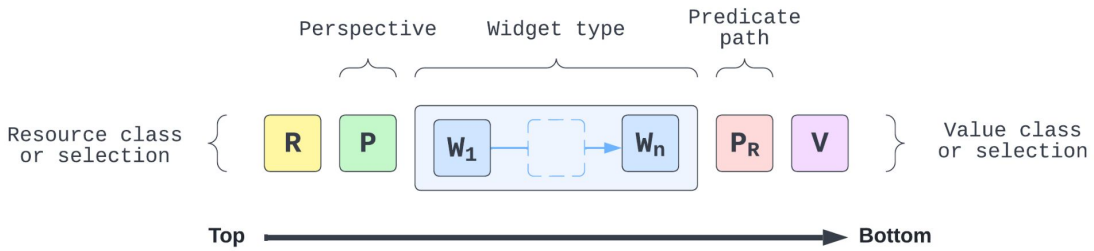
C) Hierarchical browsing
=> Proposal: simple tree view

Adaptive UI

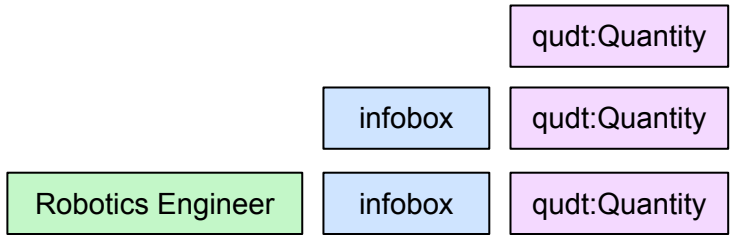
Adaptive UI Concepts

- based on the concepts of ***Scopes*** and ***Configurations***
  - introduced by Linked Data Reactor

Scope:



Examples:



Configuration:

```
(10)  class = "cobot:RobotType"
(11)
(12)  perspective _ widget infobox {
(13)    handler = "GeneralInfoBox"
(14)    config {
(15)      sections = ["prop_table",
(16)                  "recommendations"]
(17)    }
(18)  }
(19)  perspective _ widget infobox section prop_table {
(20)    handler = "ProvenanceTableSection",
(21)    config {
(22)      neighbourhood {
(23)        include = ["cobot:degreesOfFreedom",
(24)          "cobot:handlingPayload", "cobot:reach",
(25)          "cobot:skills"],
(26)      }
(27)    }
(28)  }
(29)  perspective _ widget infobox {
(30)    property "cobot:reach" {
(31)      handler = "LinkedProperty"
(32)      value _ {
(33)         handler = "TextValue"
(34)       }
(35)       value "qudt:Quantity" {
(36)        handler = "QudtQuantityValue"
(37)       }
(38)    }
(39)  }
```

Adaptive UI Concepts

- defining different entry points for exploration

Software Engineer | entry

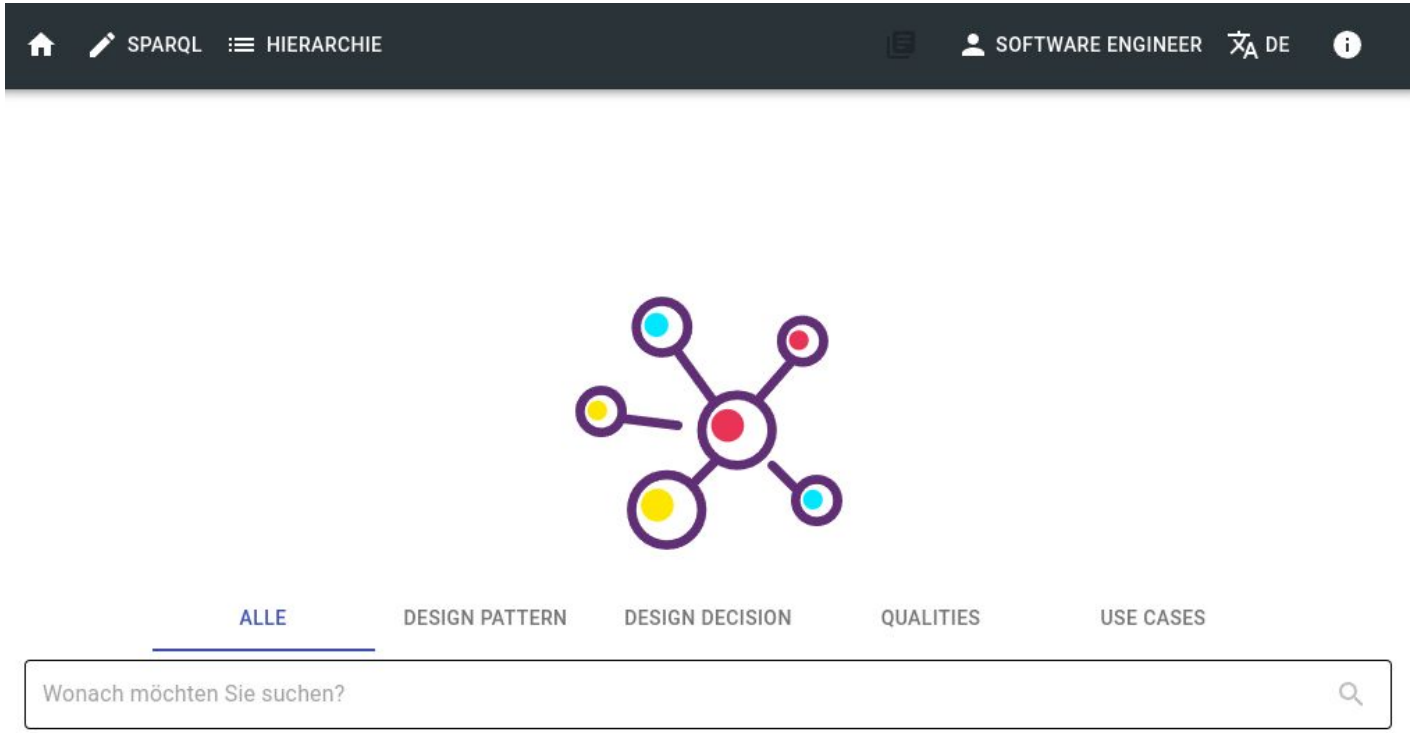
=> search page with relevant categories

Robotics Engineer | entry

=> faceted navigation

Knowledge Engineer | entry

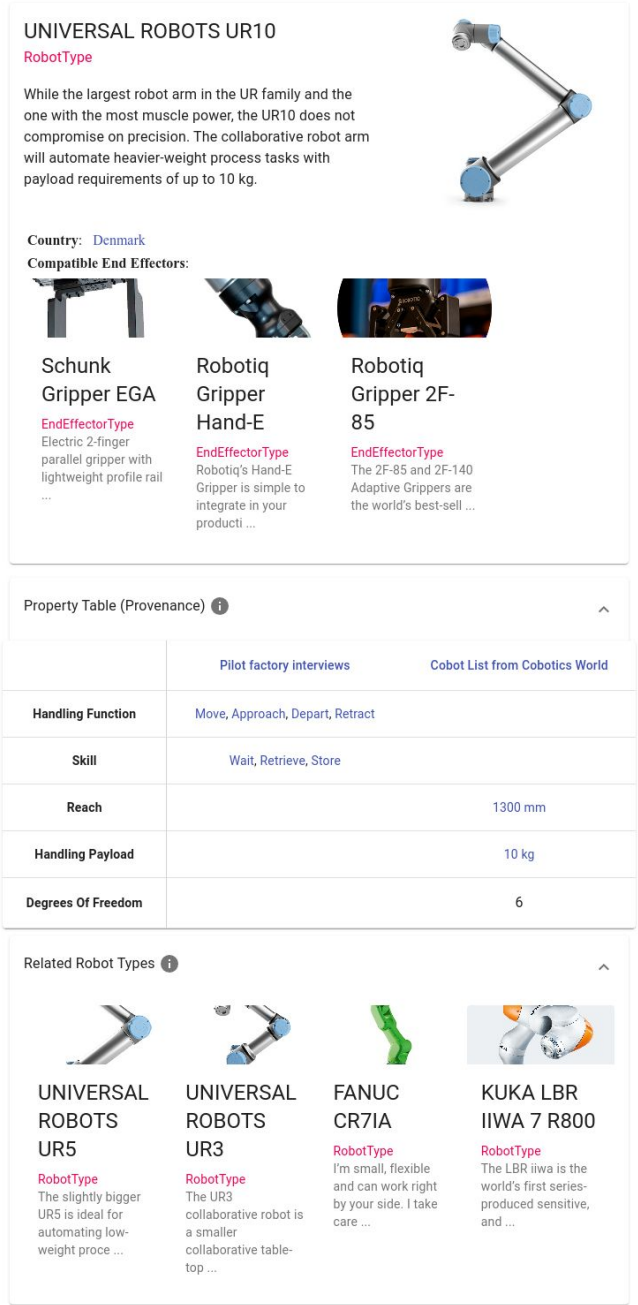
=> Protegé-like UI



Adaptive UI Concepts

- changing metrics based on perspective

```
(40)  perspective _ widget infobox {
(41)    section recommendations {
(42)      handler = "SimilaritySection"
(43)      config {
(44)        number = 4
(45)        ranking = {
(46)          ldsd {
(47)            step = "esm.exploit.sim.ldsd"
(48)            weight = -1.0
(49)          }
(50)        }
(51)      }
(52)    }
(53)  }
(54)  perspective RoboticsEngineer widget infobox {
(55)    section recommendations {
(56)      config {
(57)        classes = ["cobot:RobotType"]
(58)      }
(59)    }
(60)  }
(61)  perspective SoftwareEngineer widget infobox {
(62)    section recommendations {
(63)      config {
(64)        classes = ["cobot:HandlingFunction",
(65)                   "star:ArchitecturalElement"]
(66)      }
(67)    }
(68)  }
```



System Architecture

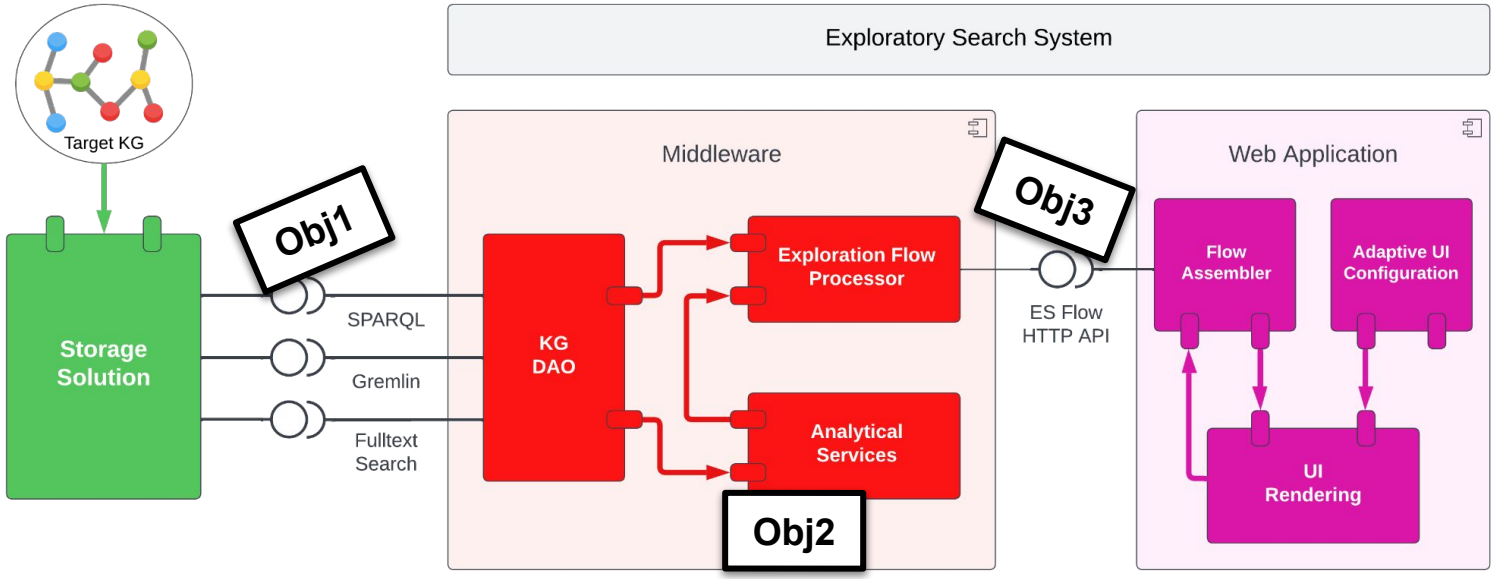
Exploratory Search System Architecture

- Many KG-based ES systems rely only on SPARQL
  - we introduce a middleware

Obj1) adaptable to different legacy environments

Obj2) easily plugin new algorithms for knowledge graph analytics

Obj3) allow experimentation with different interface paradigms



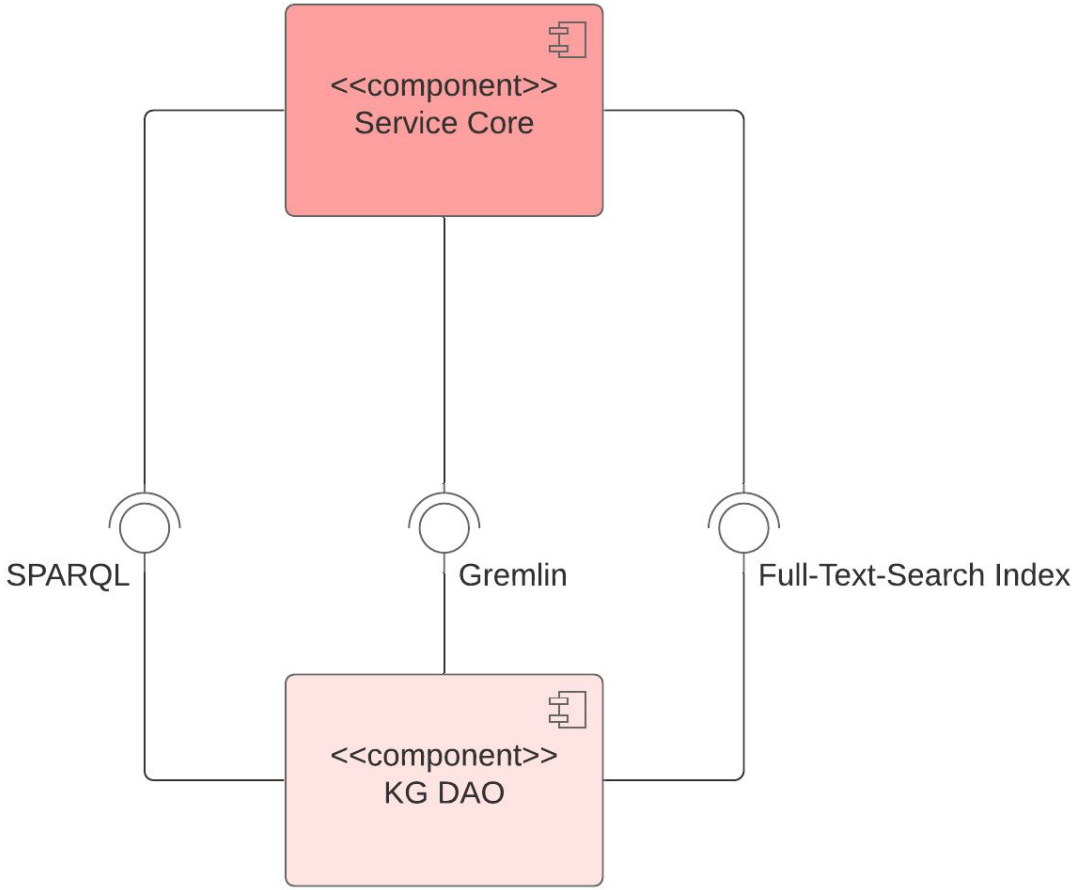
Obj1) Semantic Storage Abstraction

Challenge:

- SPARQL not well suited for some graph algorithms
- No standardized SPARQL feature for full-text searches

=> abstraction of storage solution with three interfaces:

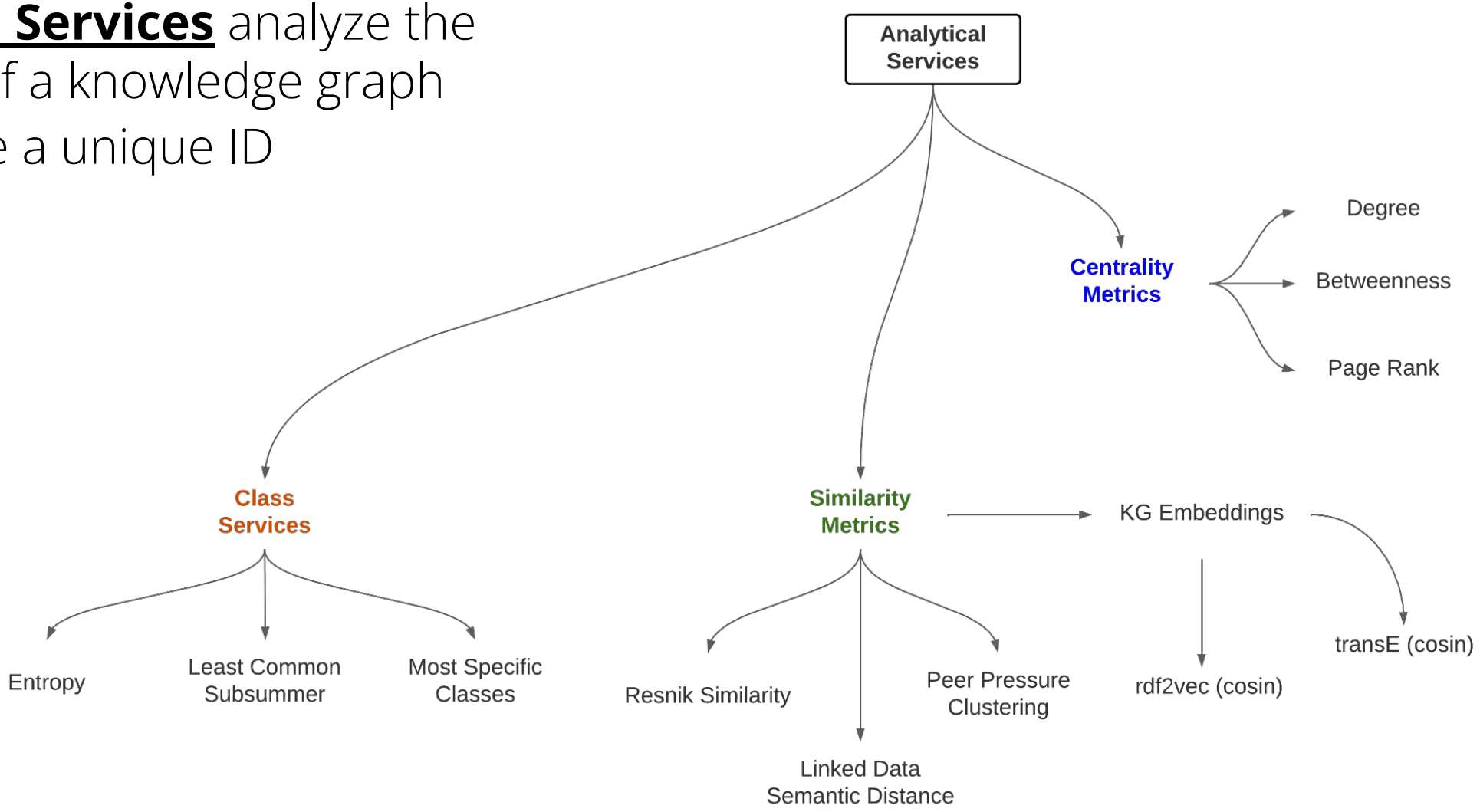
1. SPARQL
2. Gremlin
3. Full-Text-Search Index



Obj2) Analytical Pipeline

<u>Analytical Services</u> analyze the structure of a knowledge graph

- have a unique ID

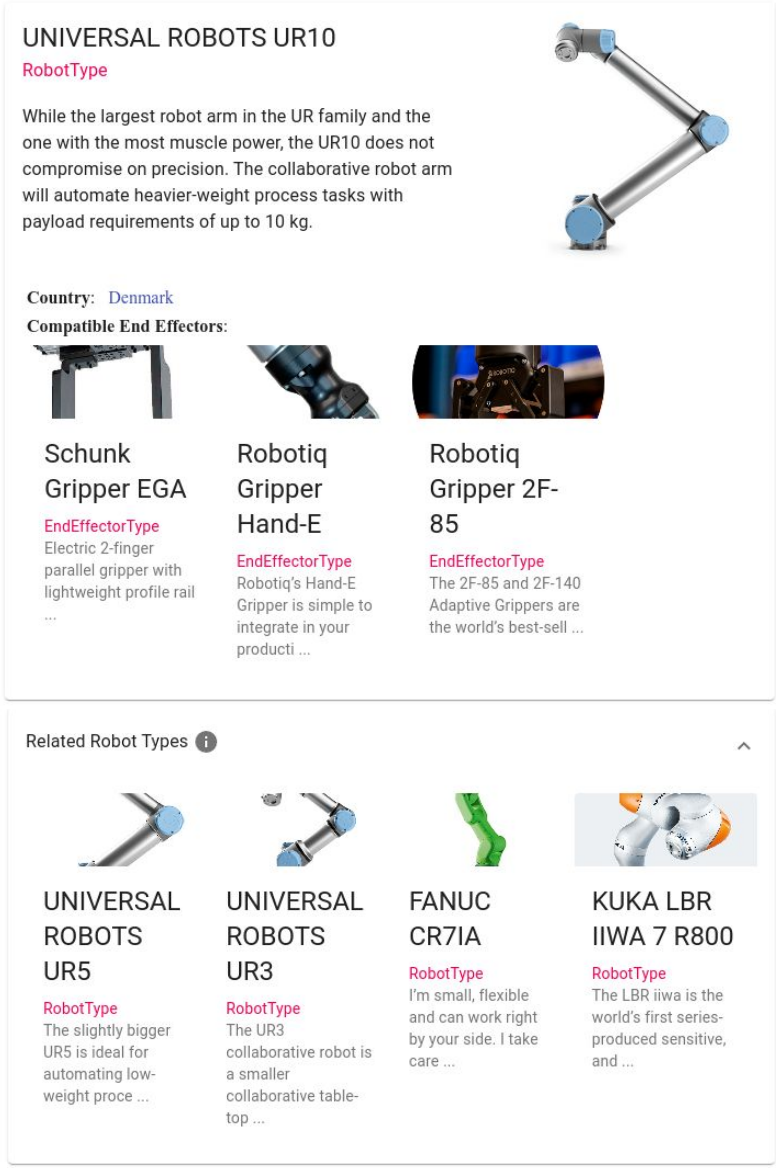


Obj3) Exploration Flow API

Flow API exposes only the basic concepts of RDF to UI.

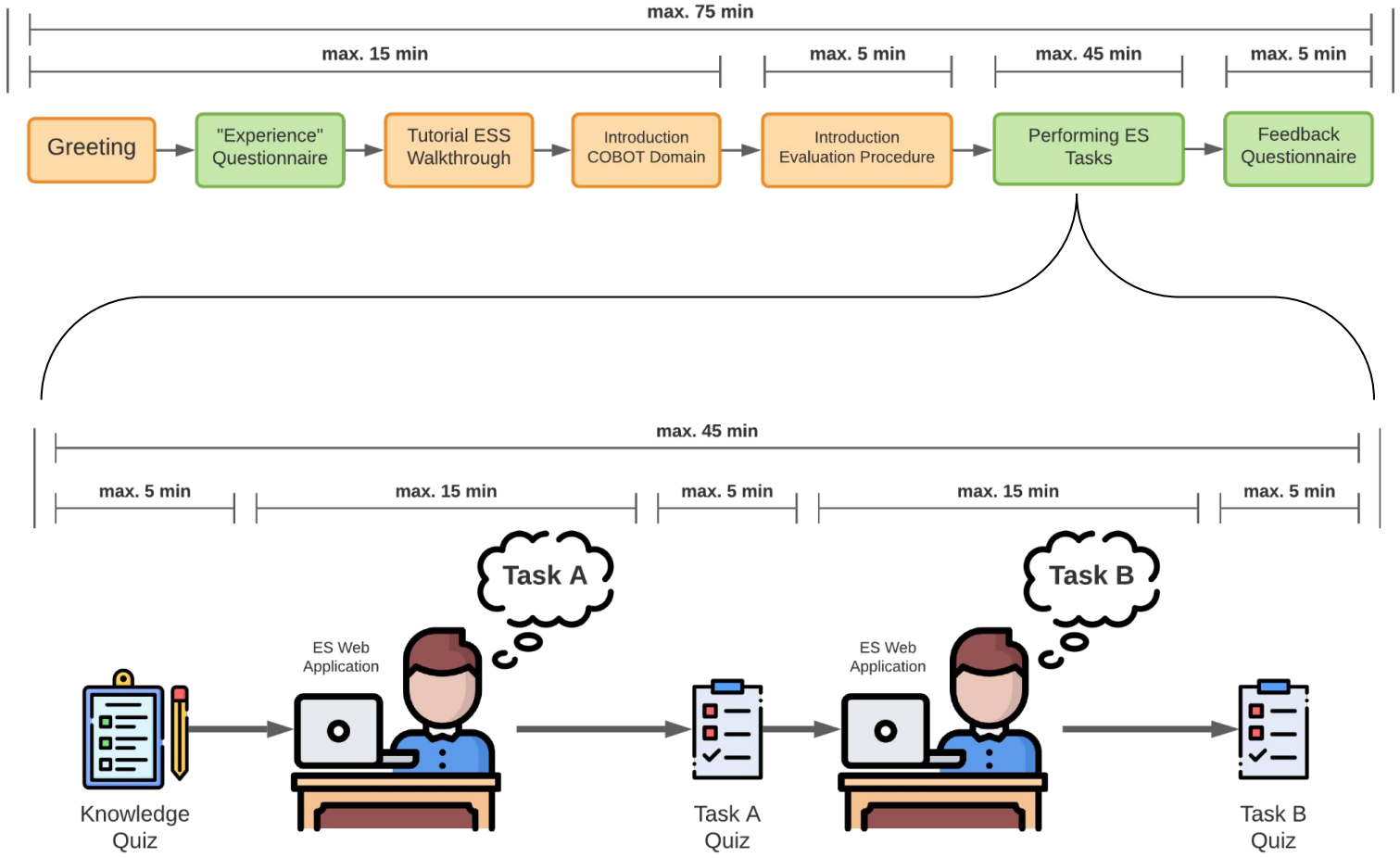
- flow = a sequence of steps

```
f = Single(resource='robot:universal-robots-ur10') \
   >> PairWith(flow=All(include=['cobot:RobotType'])) \
    >> LDSD() >> OrderBy(Sim.ldsd, strategy=Order.DESC) \
   >> Limit (n=4) >> Describe()
resp = FlowAPI("http://localhost:8080").execute(f)
```



Evaluation

Evaluation Procedure



Exploratory search system applied to a small knowledge graph about cobots and software architectural knowledge.

Outcome:

- Screen recordings
- SUS
- “Usefulness” of certain features
- Knowledge gain

Conclusion

Conclusion

- Participants were able to present reasonable solutions to task

- Issues:
  - memorization feature was ignored in favor of browser tabs
  - tree view had a low click rate and participants struggled with it

- Future work:
  - Evaluation with more participants (>20)
  - RDF-star is a challenge for adaptive UI

Questions?

https://kevinhaller.dev/papers/22-voila-slides.pdf

References

[Pas12] Passant, Alexandre. "Seevl: mining music connections to bring context, search and discovery to the music you like." 2012.

[Mar13] Nicolas Marie, Fabien Gandon, Myriam Ribière, and Florentin Rodio. 2013. “Discovery hub: on-the-fly linked data exploratory search.” In Proceedings of the 9th International Conference on Semantic Systems (I-SEMANTICS '13).

[Nuz17] Nuzzolese, Andrea Giovanni, et al. "Aemoo: Linked data exploration based on knowledge patterns." Semantic Web 8.1 (2017): 87-112.

[Pal18] Evaluating exploratory search engines: designing a set of user-centered methods based on a modeling of the exploratory search process, Palagi Emelie, PhD Thesis, 2018

[Ehr21] Ehrhart, Thibault, Pasquale Lisena, and Raphaël Troncy. "KG Explorer: a Customisable Exploration Tool for Knowledge Graphs." VOILA 2021, International Workshop on the Visualization and Interaction for Ontologies and Linked Data. 2021.