

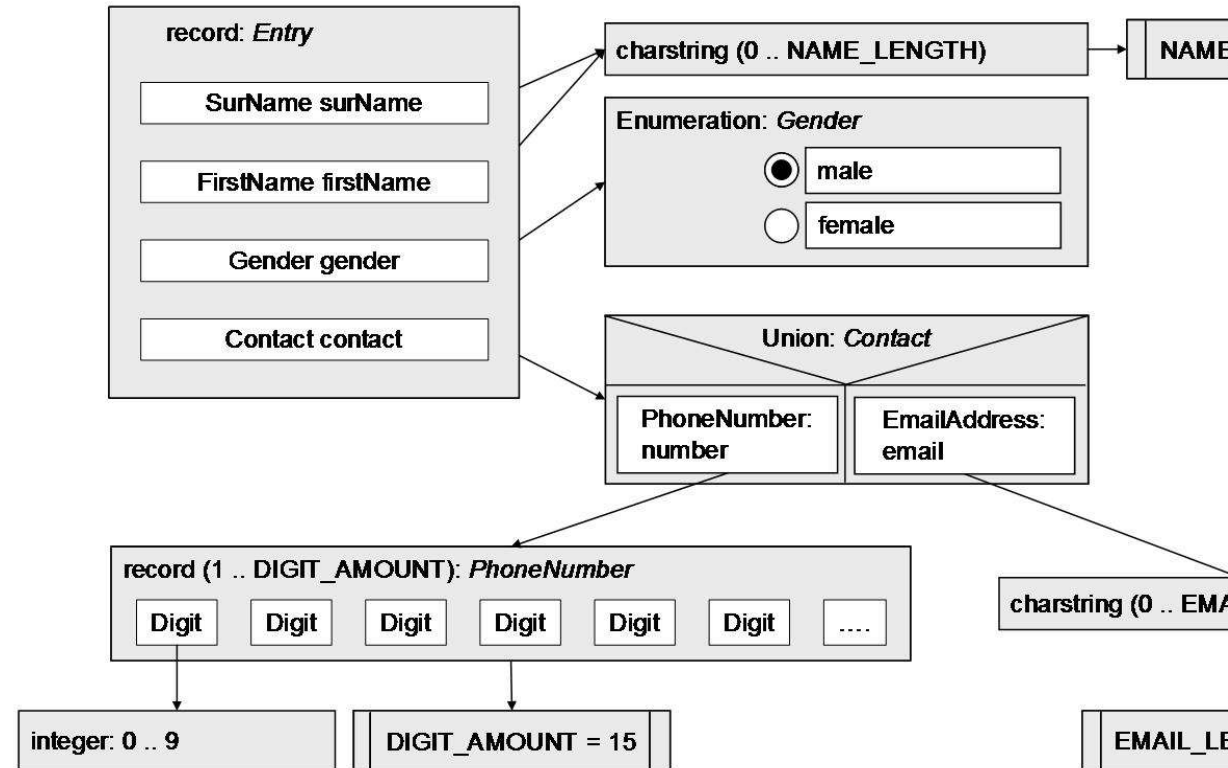
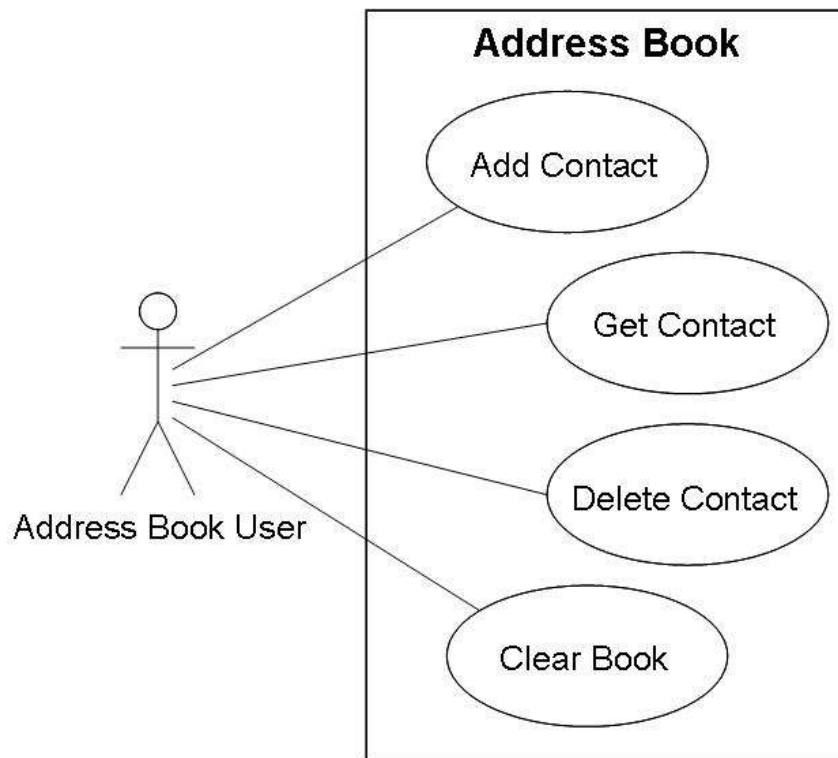
Mit KI zu besseren Testergebnissen

Prof. Dr.-Ing. Ina Schieferdecker | TAV 51, Hamburg, Jul 4, 2025

→ Generative KI als Link zwischen Test- und Produktivcode

Semjon Mössinger, WPS, Jul. 3, 2025

Unit Tests für ein AddressBook



Code Explanations AddressBook

The AddressBook class serves as a repository for storing and managing address book entries. It supports adding, retrieving, deleting, and erasing entries, as well as loading from and saving to a file.

The main functionalities of the AddressBook class include managing entries by adding, retrieving, deleting, and erasing them. It also supports loading entries from an XML file and saving them back to a file.

Methods

- `addEntry(Entry entry)`: Adds a new entry to the address book if it doesn't already exist and the size limit isn't reached.
- `getEntry(String surName, String firstName)`: Retrieves an entry by the given surname and first name.
- `deleteEntry(Entry entry)`: Deletes a specified entry from the address book.
- `erase()`: Clears all entries from the address book.
- `getEntries()`: Returns all entries in the address book.
- `load(File data)`: Loads entries from an XML file.
- `save(File data)`: Saves entries to an XML file.
- `toString()`: Provides a string representation of the address book.

Fields

- `SortedSet<Entry> entries`: Stores the address book entries sorted by name.
- `final static int sizeLimit`: Defines the maximum number of entries allowed in the address book, set to 10.

```
AddressBook addressBook = new AddressBook();
Entry entry = new Entry();
entry.setName("John", "Doe");
entry.setGender(Gender.Male);
entry.setContactPhoneNumber(123456789);

try {
    boolean added = addressBook.addEntry(entry);
    System.out.println("Entry added: " + added);
} catch (SizeLimitReachedException e) {
    System.out.println("Address book is full.");
}

Entry retrievedEntry = addressBook.getEntry("Doe", "John");
System.out.println("Retrieved Entry: " + retrievedEntry);

boolean deleted = addressBook.deleteEntry(entry);
System.out.println("Entry deleted: " + deleted);

addressBook.erase();
System.out.println("All entries erased.");

File file = new File("addressbook.xml");
try {
    addressBook.save(file);
    addressBook.load(file);
} catch (IOException e) {
    e.printStackTrace();
}
```

AI-assistierte Software Test-Werkzeuge Q2'2024

Tool	Key Features	Focus Areas
Testim.io	Machine learning for test authoring and maintenance	Automated UI tests
Applitools	Visual AI for automatic visual validation	Visual regression testing, cross-platform
Functionize	NLP for test creation, ML for maintenance	End-to-end tests, test analytics
Sauce Labs	AI and ML for visual testing and anomaly detection	Comprehensive testing suite, visual testing
mabl	Machine learning for end-to-end testing automation	Visual regressions, JavaScript errors, link checks
ReTest	AI for difference testing	Regression testing without explicit assertions
Parasoft	AI and ML for noise reduction in static analysis	Static analysis, test optimization
Sealights	AI and analytics for test process insights	Test quality, risk assessment, resource optimization
TestCraft	AI for test maintenance and execution	Codeless Selenium test automation
Katalon Studio	AI for healing test objects and UI test automation enhancement	UI test automation

Vor allem G
orientiert
Werkzeug

AI-assistierte Software Test-Werkzeuge Q1 '2025

Tool	Key Features	Focus Areas
Selenium	Equipped with AI to provide extensive flexibility for running and executing tests	Automating Web testing
<u>Code Intelligence</u>	Dynamic fuzz testing	AI-powered security testing for
Functionize	Machine learning-based tests use big data to understand website updates and self-heal workflows	End-to-end tests, automating C
Testsigma	Automate tests in plain English	Low code test automation for Web, n API
Katalon Studio	Testing at any scale	Fully fledged code, low code and AI-assisted test platform
Applitools	Visual AI for automatic visual validation	Visual regression testing, cross-browser/device test
<u>Eggplant Digital Automation Intelligence</u>	Model-based digital twin testing	Application and user interface testing
<u>Digital.ai Continuous Testing</u>	Continuous testing in DevOps environments	Web and mobile testing
<u>Testcraft</u>	Robust Selenium-based automated testing solution as a browser extension	Web testing
Testim	Machine learning for test authoring and maintenance	Web and mobile testing in agile environments

Mittlerwei
GUIs eber
Datenbanl
andere B
Servi

Passfähigkeit der Werkzeuge

Tool	Java	VSCode	Free
Testim	✗	✗	(✓)
Applitools	✓	✗	(✓)
Functionize	✓	✗	✗
Sauce Labs	✓	✗	(✓)
mabl	✗	(✗)	✗
ReTest	✓	✗	(✓)
Parasoft	✓	✗	✗
Sealights	✓	✗	✗
TestCraft	(✗)	✗	✗
Katalon Studio	✓	(✓)	(✓)
Code Intelligence	✓	✗	✓
Testsigma	✓	✗	✓
Eggplant Digital Automation Intelligence	✓	✓	✗
Digital.ai Continuous Testing	✓	✗	✗
<u>UnitTestAI</u>	✓	✓	✓
<u>Qodo Gen</u>	✓	✓	✓
<u>CoPilot</u>	✓	✓	(✓)

Erzeugte Tests

Tests	Exercise Functional Test	UnitTestAI	CoPilot	Qodo Gen
Positive	testAddContact	testAddEntry	testAddEntry	test_add_new_contact test_special_characters
		testDeleteEntry	testDeleteEntry	test_delete_contact test_update_contact
			testErase	
	testGetEntry	testFindEntryByName testGetAllEntriesNonEmpty	testGetEntry testGetEntries	test_get_all_contacts test_search_contacts
		testGetNumberOfEntriesNonEmpty		
				test_sort_contacts
			testSave testLoad	test_create_empty
Negative	testForException	testAddEntryWithDuplicate testDeleteEntryNotPresent testFindEntryByNameNotFound testGetNumberOfEntriesEmpty testGetAllEntriesEmpty	testAddDuplicateEntry testAddEntrySizeLimitReached testGetEntryNotFound testDeleteEntryNotFound	test_create_with_duplicates test_add_duplicate test_search_nonexistent test_load_corrupted

Korrekturen des generierten Test Codes

Tool	Corrections needed
UnitTestAI	Missing package Missing imports Wrong constructors and datatypes Wrong methods Unhandled exceptions
Qodo Gen	Missing package Missing imports Wrong return types Wrong methods Unhandled exceptions
CoPilot	Missing package Missing imports

- Source Code als Kontext setzen
- Erzeugte Unit Tests überprüfen – mit Metriken, Code Conventions und Prüfen
- Prompts zur Verbesserung der Unit Tests nutzen

Die Erwartungen an GenAI-unterstütztes Testen

Erleichterte(s)

- Generierung und Erweiterung von Testfällen: Erkennung von (Code-)Strukturen, Abläufen und relevanten Details
- Wartung von Testfällen: Self-Healing nach Software-Modifikationen
- Priorisierung von Tests
- Analyse von Testergebnissen
- Test-Reporting
- (Behebung von Fehlern)

The potential of AI in next-generation test automation

Prof. Dr.-Ing. Ina Schieferdecker | TAV 50, Kaiserslautern, Nov 14, 2024

ChatGPT 4: AI be able to test AI itself?

- In theory, it's possible for AI to test AI systems without human intervention, however, there are several challenges and limitations:
 - Complexity of AI Systems
 - Interpretability and Explainability
 - Biases and Blind Spots
 - Adaptability to Novel Scenarios
 - Ethical Considerations
- AI-based testing tools and techniques can augment human testers' capabilities and automate certain aspects of the testing process
- Human judgment, creativity, and domain expertise remain essential for ensuring the quality, reliability, and ethical integrity of AI systems.
- While AI may eventually play a significant role in testing AI systems, human involvement is likely to remain necessary **for the foreseeable future**.

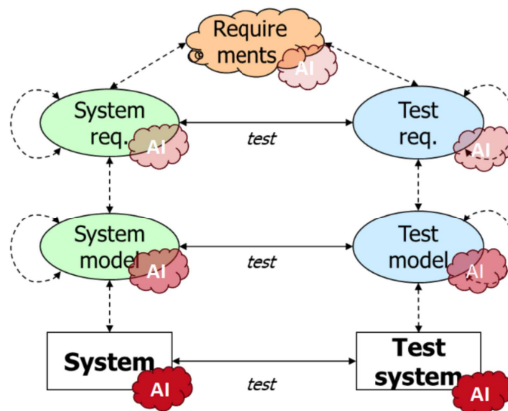
Seite 2

Prof. Dr.-Ing. Ina Schieferdecker | TAV 50, Kaiserslautern, Nov 14, 2024

Software Testing in the AI Era ?

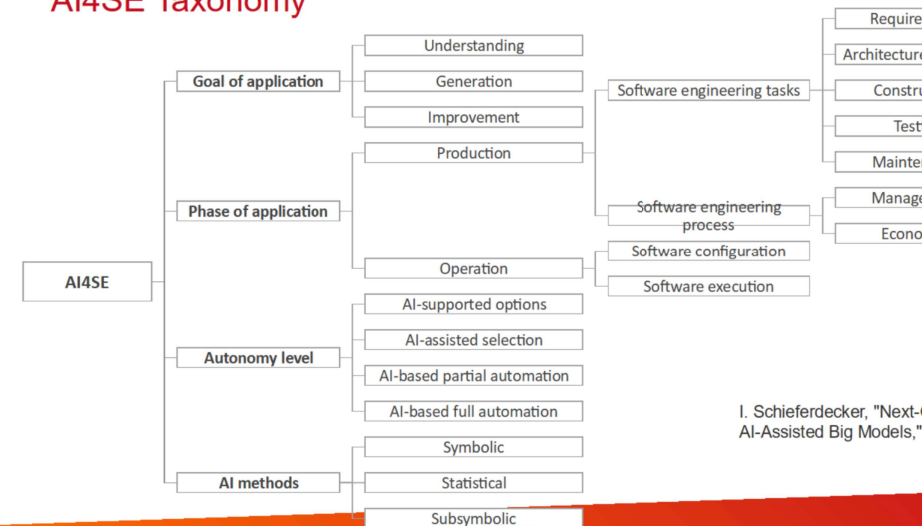


Model-based software
engineering



Model-based testing v4

AI4SE Taxonomy



I. Schieferdecker, "Next-
AI-Assisted Big Models,"

Seite 17

Prof. Dr.-Ing. Ina Schieferdecker | TAV 50, Kaiserslautern, Nov 14, 2024

Seite 23

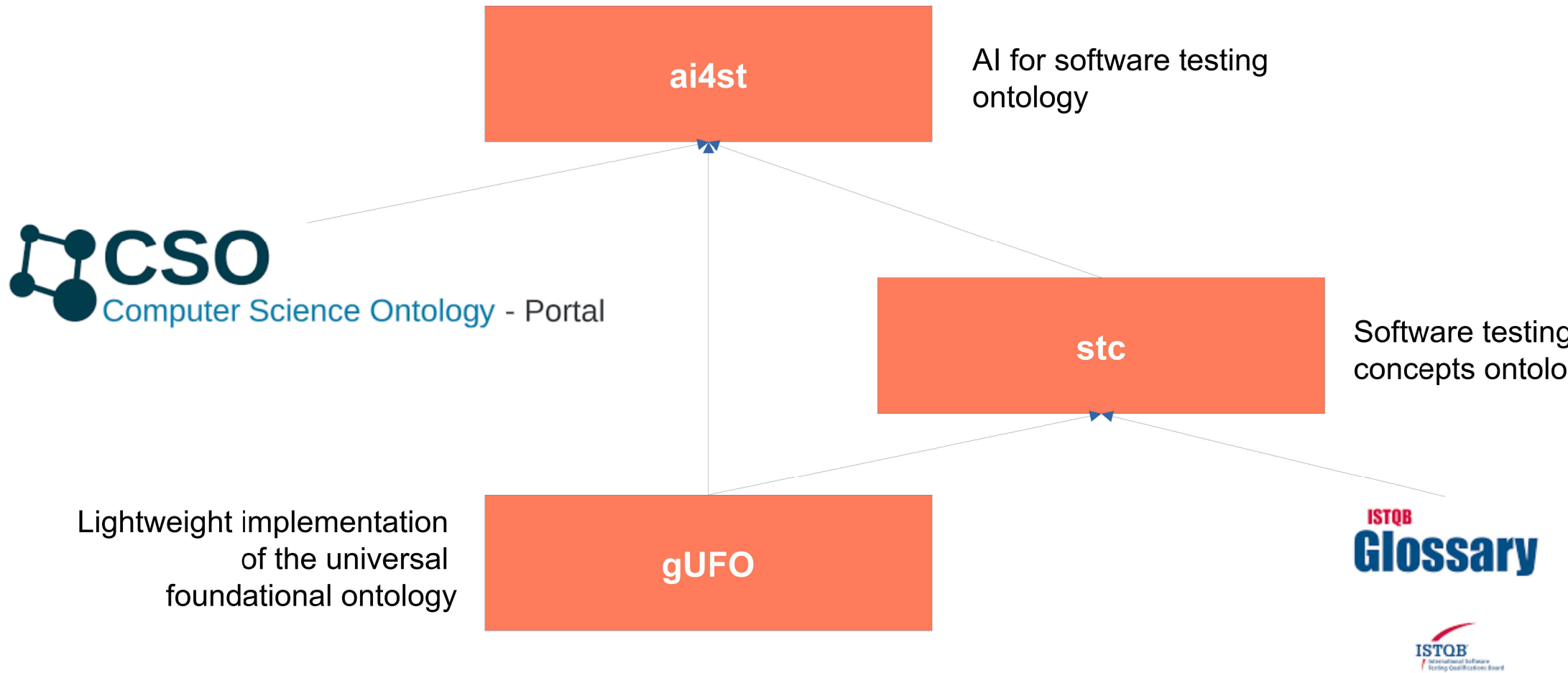
Prof. Dr.-Ing. Ina Schieferdecker | TAV 50, Kaiserslautern, Nov 14, 2024

Navigating the growing field of research on AI for software testing

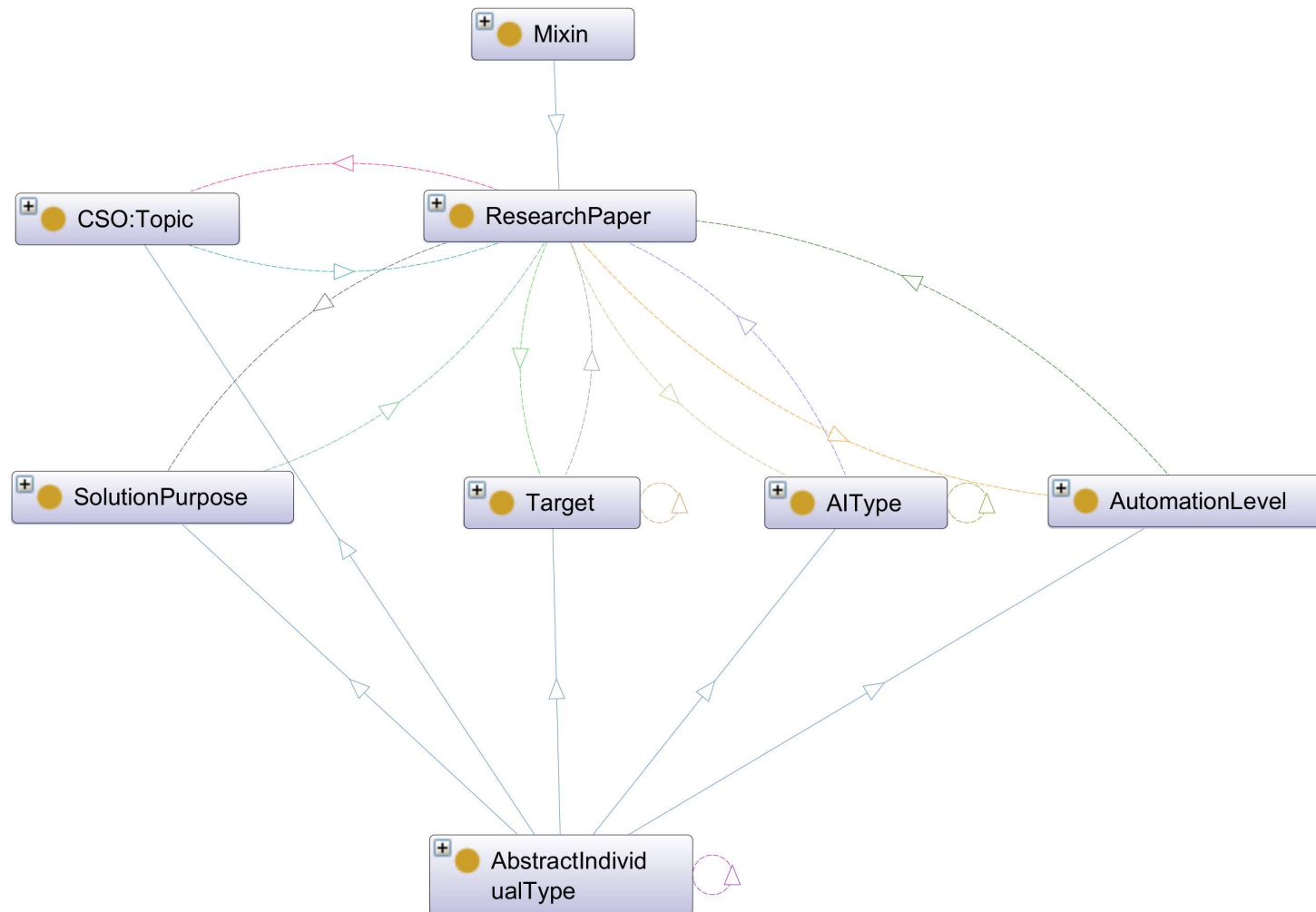
*The taxonomy for AI-augmented software testing
and an ontology-driven literature survey*

<https://www.arxiv.org/abs/2506.14640>

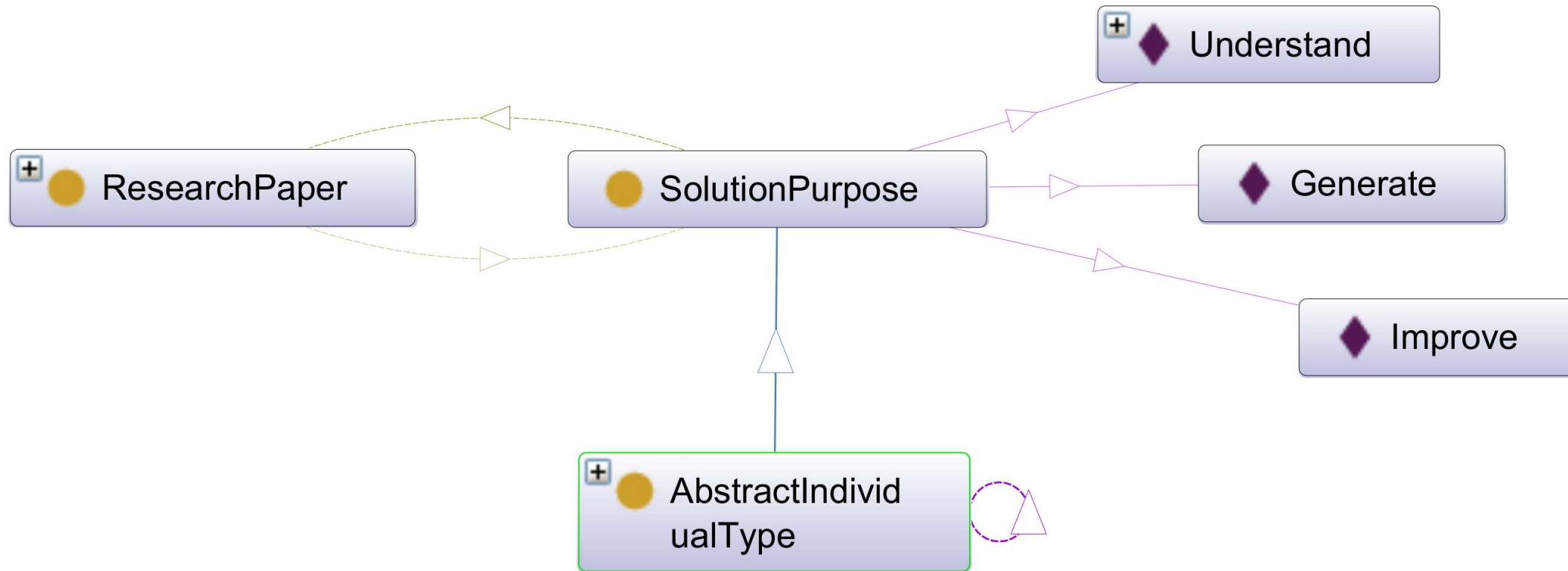
The AI4ST Ontology



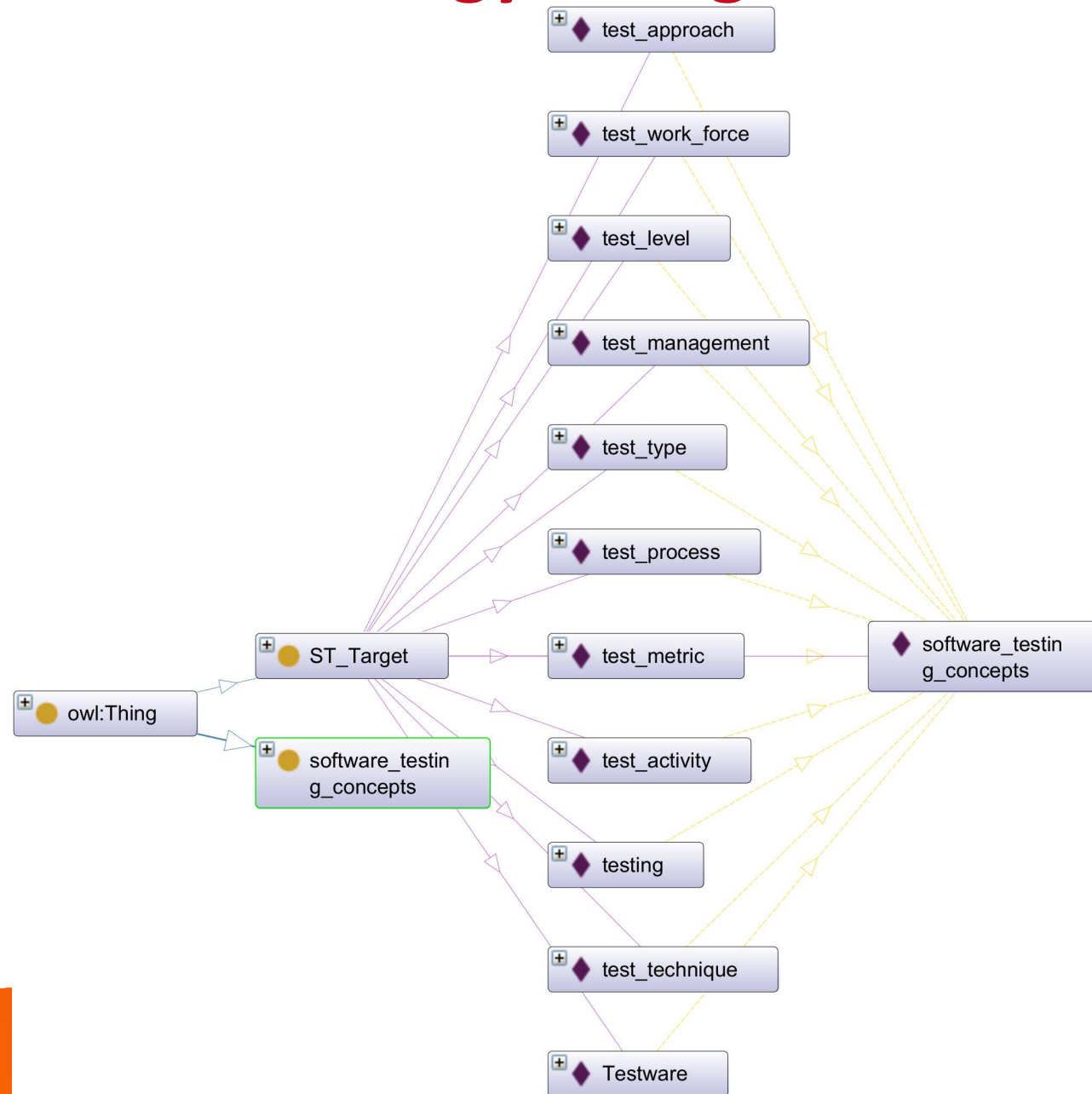
The AI4ST Ontology - Overview



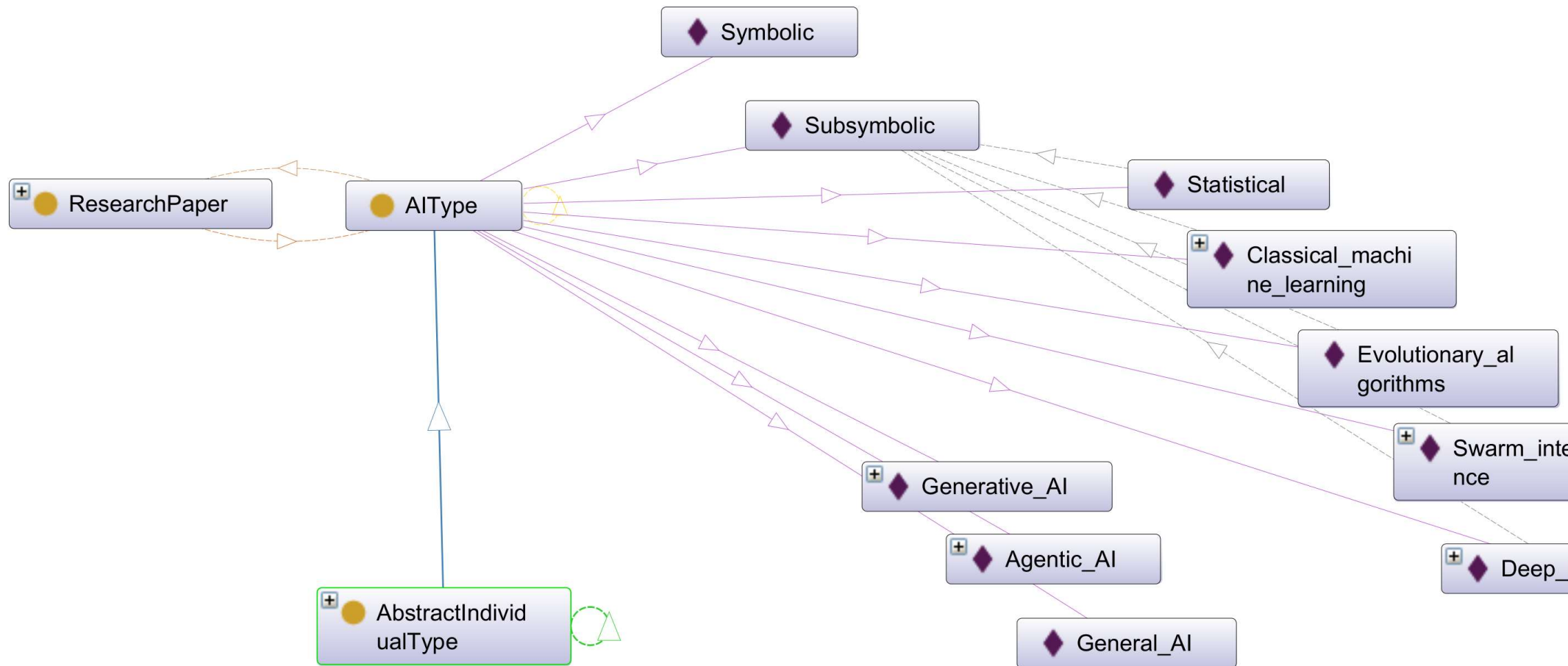
The AI4ST Ontology – Purpose Dimension



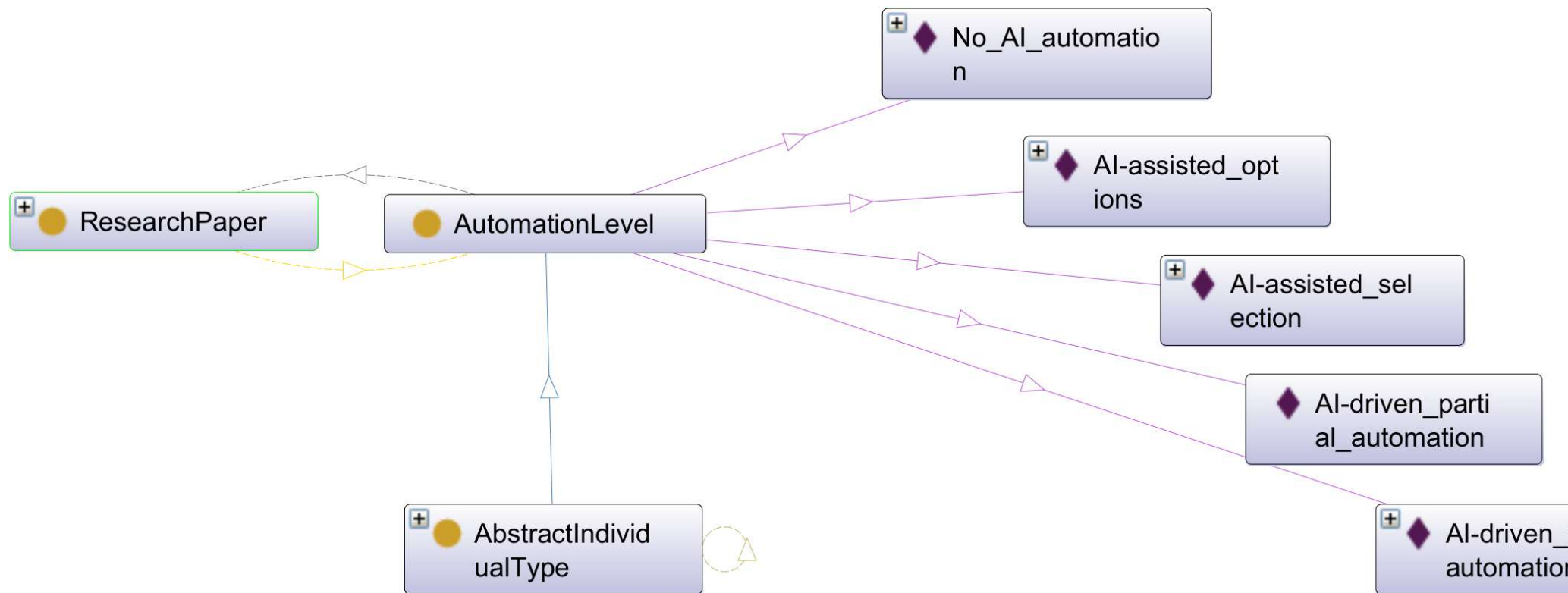
The AI4ST Ontology – Target Dimension



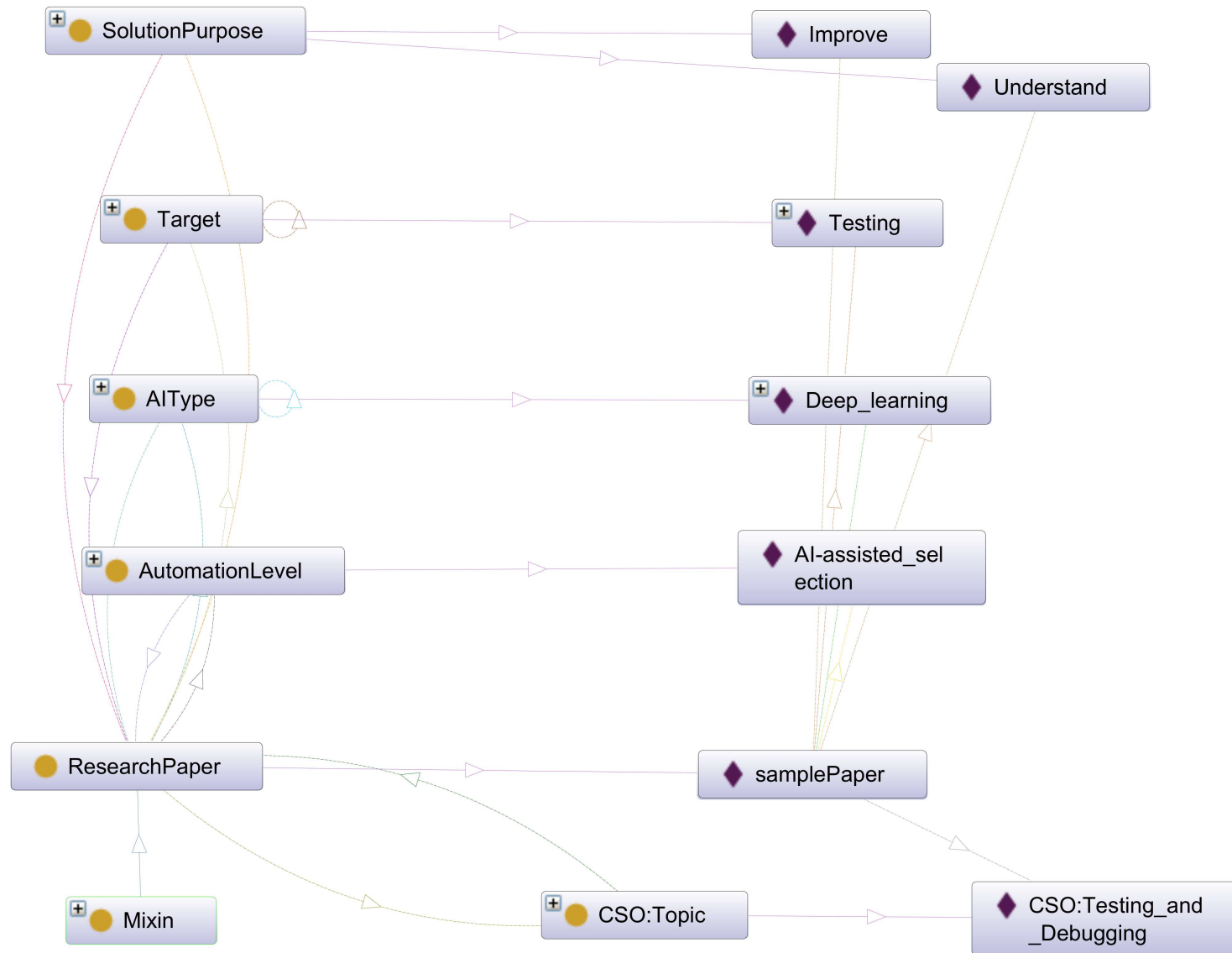
The AI4ST Ontology – AI Type Dimension



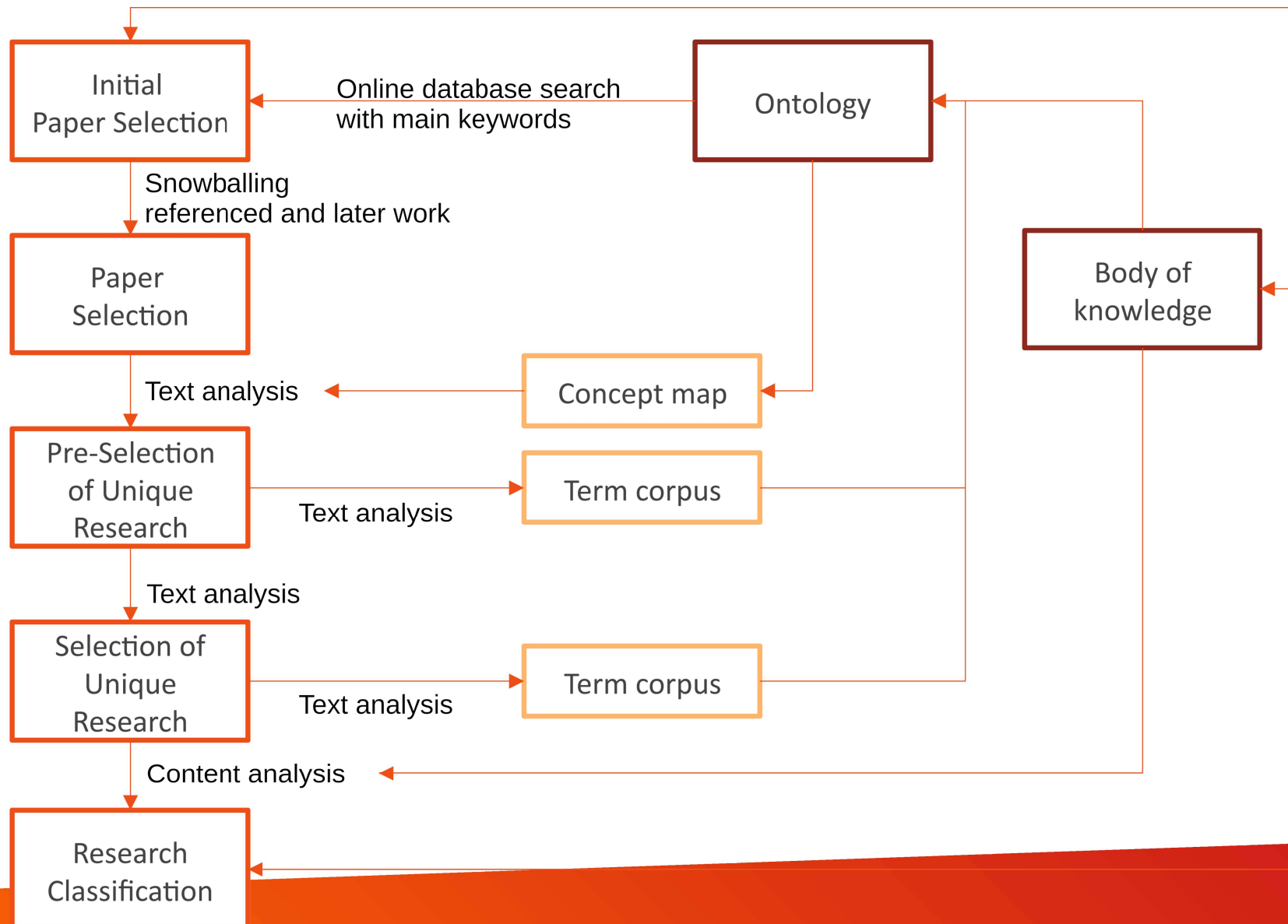
The AI4ST Ontology – Automation Level Dimension



The AI4ST Ontology – Sample Paper



AI4ST Systematic Literature Review



SRL Results

- Starting with ICSE 2025 + co-located conferences/workshop and backward snowballing
- **1643** papers identified
- **1150** referred to a term in the stc ontology in their abstracts and/or titles, but 735 of these referred to only one term
- Another **1337** papers used variations of the terms in the stc ontology, such as 'unit test' instead of 'unit-level test', but 460 papers referred to only one variation.
- **949** papers using two or more original or alternative terms.
- Of these 949 papers, **656** contained terms from the ai4st ontology related to AI.
- Within the 656 papers, **38** relevant **original** research papers for ai4st were identified.

SRL Results

Text analysis revealed

- 40 new stc term candidates:
 - terms such as 'test result' and 'fuzz testing' - included in the ISTQB Glossary, but not yet in stc
 - terms such as 'flaky test' and 'genetic testing' - not included in the ISTQB Glossary, but extensively discussed in research
 - Including terms such as 'test bot' and 'bias testing' related to AI for software testing or testing of AI
- 53 new stc synonym candidates and
- 26 terms that can be treated as either new term or synonym candidates.

(stc: over 200 terms, ISTQB glossary over 600 terms)

38 papers classified in AI4ST

Table 1: List of Research Papers

Ref	Title
[12]	A Method and Experiment to evaluate Deep Neural Networks as <u>Test Oracles</u> for Scientific Software
[40]	A new approach for automatic test case generation from use case diagram using LLMs and prompt engineering
[17]	<u>Acceptance Test</u> Generation with Large Language Models: An Industrial Case Study
[11]	AI in Service of Software Quality: How ChatGPT and Personas Are Transforming Exploratory Testing
[33]	AI-Assisted Test Script Generation for GUI Applications
[39]	AI-Based Enhancement of <u>Test Models</u> in an Industrial Model-Based Testing Tool
[36]	AI-Driven Acceptance Testing: first insights exploring the educational potential for test analysts
[29]	AI-Driven Testing: Unleashing Autonomous Systems for Superior Software Quality Using Generative AI
[37]	AI-driven web API testing
[23]	AI-Powered Multi-Agent Framework for Automated Unit Test Case Generation: Enhancing Software Quality through LLMs
[26]	AI's Understanding of Software <u>Test Architecture</u>
[43]	An Agent-based Architecture for AI-Enhanced Automated Testing for <u>XR Systems</u>
[34]	An Approach To Extract <u>Optimal Test Cases</u> Using AI
[21]	An Approach to GUI Test Scenario Generation Using Machine Learning
[44]	AssertT5: Test Assertion Generation Using a Fine-Tuned Code Language Model
[41]	Automation of Test Skeletons Within Test-Driven Development Projects
[69]	BugBlitz-AI: An Intelligent QA Assistant
[20]	ClozeMaster: Fuzzing Rust Compiler by Harnessing LLMs for Infilling Masked Real Programs
[66]	Deep Multiple Assertions Generation

38 papers classified in AI4ST

...continued from previous page

- [6] Development of Cloud and Artificial Intelligence based Software Testing Platform (ChArIoT)
- [22] Generative AI for Software Test Modelling with a focus on ERP Software
- [60] Ethical AI-Powered Regression Test Selection
- [28] Getting pwn.d by AI: Penetration Testing with Large Language Models
- [72] GUI-Based Software Testing: An Automated Approach Using GPT-4 and Selenium WebDriver
- [8] Leveraging Large Language Models for Usability Testing: a Preliminary Study
- [18] LLM-Based Labelling of Recorded Automated GUI-Based Test Cases
- [70] New Approaches to Automated Software Testing Based on Artificial Intelligence
- [42] On the Effectiveness of LLMs for Manual Test Verifications
- [19] Owl Eye: An AI-Driven Visual Testing Tool
- [35] Reducing Workload in Using AI-based API REST Test Generation
- [10] Reinforcing Penetration Testing Using AI
- [24] RIVER 2.0: an open-source testing framework using AI techniques
- [1] TCP-Net++: Test Case Prioritization Using End-to-End Deep Neural Networks Deployment Analysis and Enhancements
- [30] TOGLL: Correct and Strong Test Oracle Generation with LLMs
- [54] Using an agent-based approach for robust automated testing of computer games
- [12] Using Large Language Models to Generate Concise and Understandable Test Case Summaries
- [45] Visual Test Framework: Enhancing Software Test Automation with Visual Artificial Intelligence and Behavioral Driven Development
- [27] WIP: Assessing the Effectiveness of ChatGPT in Preparatory Testing Activities

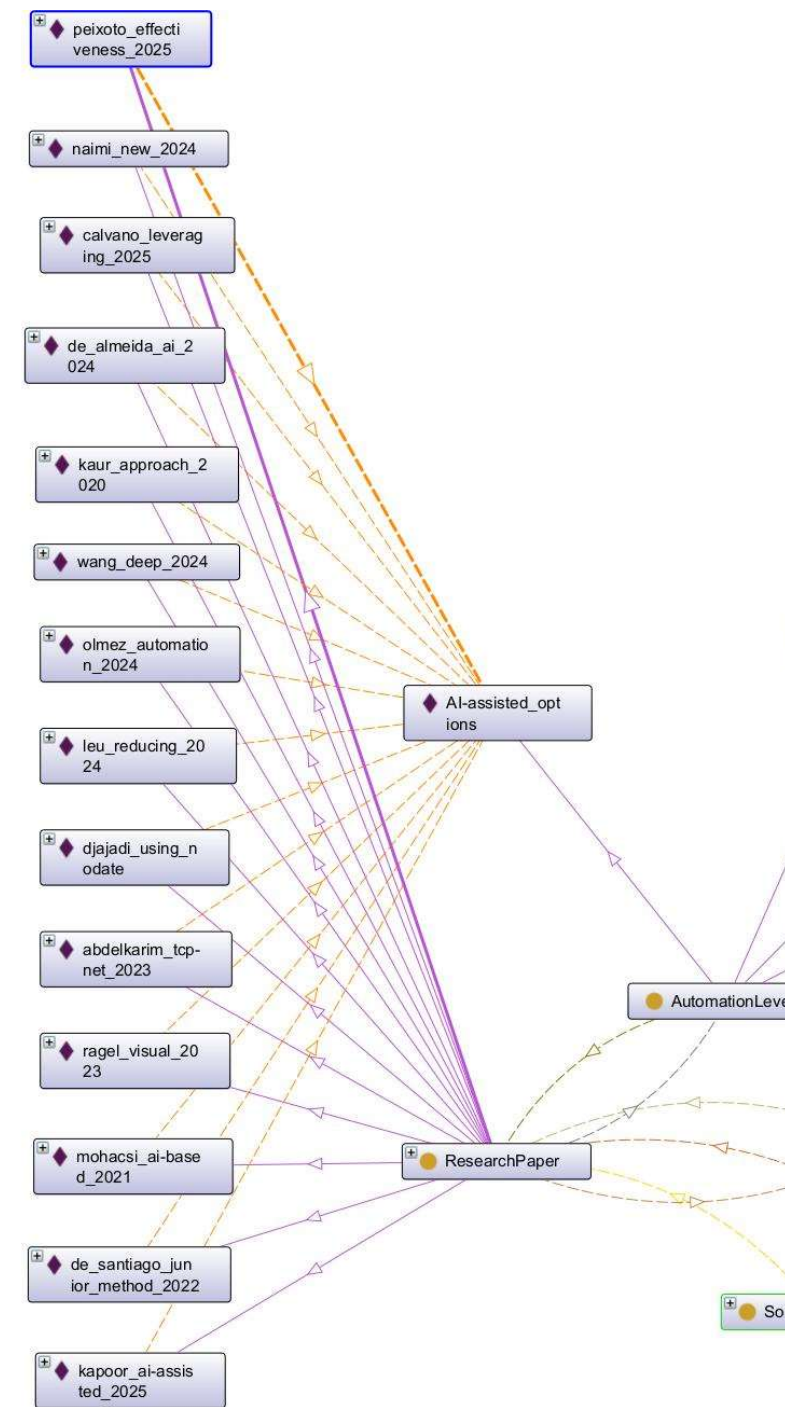
Interactive paper search

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX ai4st: <http://purl.org/ai4st/ontology#>
3 SELECT ?paper
4 WHERE {
5   ?paper ai4st:hasLevel ai4st:AI-assisted_options .
6 }
```

Listing 1.1: All papers on AI-assisted options automation.

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX ai4st: <http://purl.org/ai4st/ontology#>
3 SELECT DISTINCT ?target
4 WHERE {
5   ?paper rdf:type ai4st:ResearchPaper .
6   ?paper ai4st:hasTarget ?target .
7 }
```

Listing 1.2: All software testing targets addressed by research papers.



Thank you for your attention.

If you want to follow up:

Navigating the growing field of research on AI for software testing

<https://github.com/schieferdecker/ai4st>

I used beside other the following tools :



zotero



citation gecko

