

DSL Project

Niklas de Bruyn

Malte Schulze Balhorn

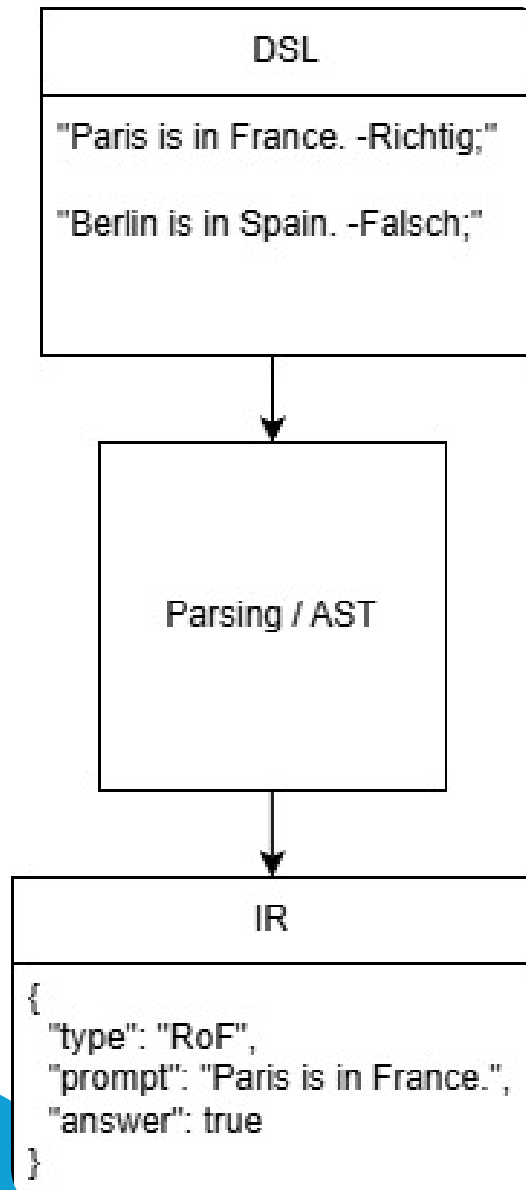
Robin Thielking

Motivation

- E-learning has gained significant prominence in recent years
- Didactic content can be delivered more efficiently in serious games or other areas
- Manually creating tasks is time-consuming and yields limited reusability

⇒ simple DSL which requires zero technical knowledge and can be used to generate worksheets and be interpreted for various digital formats





Why is it fun?

- Brings compiler construction concepts to life
- **DSL**: simple and intuitive for users
- **IR**: captures the exact meaning of tasks
- Forces abstraction of diverse task types
- A creative yet precise design challenge

Our Goal: DSL and IR

- DSL
 - Develop a DSL to be used by teachers to describe test tasks or exams
 - Easy and intuitive in design
 - Capable of describing the most important task types
- IR
 - Create IR to be used by e.g. ILIAS or our dungeon game to generate tasks
 - Ensure the IR contains all required information

Domain Specific Language

- Primary language is German
- Multiple task types:
 - Single-/Multiple-Choice
 - True or False
 - Matching
 - Sort
 - Marking
 - Cloze

Intermediate Representation

- General structure to be determined
- Needs to contain required information depending on use case to remain compatible with systems such as e.g. ILIAS
 - points
 - maximum points
 - metadata

How can we evaluate our success empirically?

Robustness Tests

- How effective are the static checks?
- How stable are the grammar and the translator?
- Test with unexpected inputs:
 - wrong types
 - missing IDs
 - different element order
 - syntax errors

Example: *How many of 100 mutated tasks are correctly detected by the pipeline?*

Parser & IR Performance

- Measure parsing time
- Measure IR generation time