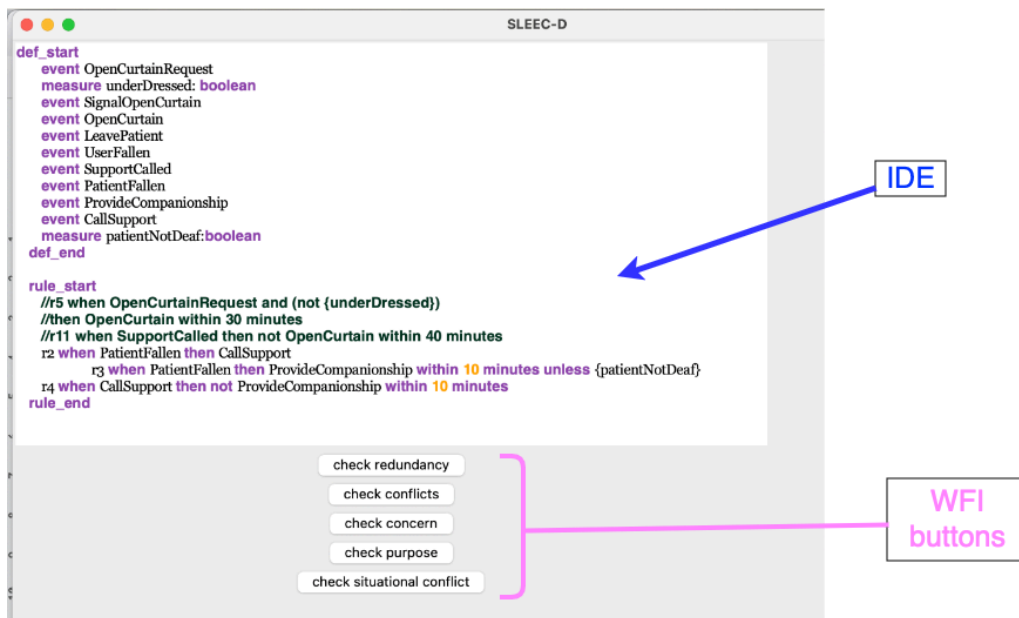


Demo 1:

1. Launch the tool: `python3 sleecFrontEnd.py`

2. **LEGOS-SLEEC overview:**

You will see LEGOS-SLEEC, which consists of two parts: the IDE and the buttons to check well-formedness issues.



3. **Using SLEEC rules:**

Let's use the SLEEC rules examples we discussed in our classes. Copy and paste the content of `demo1.sleec` into the IDE.

4. **Maintaining your own copy:**

It is always better to keep an updated copy of your modifications to `demo1.sleec` in your IDE of choice (e.g., Emacs, Geany, or Vim). **Note that LEGOS-SLEEC does not allow saving modifications.**

5. **SLEEC definitions and rules:**

You will see definitions, rules, concerns, and purposes in the IDE. **Measures used in the rules are always enclosed in curly brackets { }**. This helps LEGOS-SLEEC differentiate them from events.

6. Syntax errors:

- Please keep a terminal window open. Syntax errors are shown there rather than in the IDE.

- Let's misspell an event and run the tool again. Observe that the error is detected and message displayed in the terminal. Let's fix the error.

7. Checking for redundant rules:

- Let's check if there is any redundancy. Here it is, as seen this morning.
- We can resolve redundancy by simply commenting out rule 5.

8. Checking for vacuous conflicting rules:

- Now, let us check if there are any vacuous conflicting rules.
- We can resolve these conflicts similarly.

9. Checking whether the rules are insufficient (button `check concern`):

Next, let us check if there are any concerns that could arise while respecting our rules. Here it is, as seen this morning.

10. Resolution of the insufficiency detected:

To resolve this, we can add the following new rule to ensure the system calls emergency services as soon as possible (e.g., within 20 seconds):

```
Rc1 when SmokeDetecorAlarm and (not {userDisablesAlarm}) then  
CallEmergencyServices within 20 seconds
```