Using an inference engine to manage UI complexity

Iulia Dobai
Philips Applied Technologies
Alcor Scenario Engine
June 19, 2007

...about the speaker...

• Iulia Dobai
  – Born in Brasov, Romania
  – 2000 – 2004 - Bachelors in Computer Science from “Transylvania University” in Brasov
  – 2005 – 2007 – postmaster in Software Technology (OOTI), Stan Ackerman’s Institute, TU/e
  – Passionate about dancing (salsa), sports (squash, tennis), mountains.
...why this presentation?...

- OOTI Program:
  - assignment at Philips Applied Technologies in the ALCOR project.

- Supervisors:
  - Gerrit-Jan Bloem & Njin-zu Chen: AppTech
  - Alexander Serebrenik: TU/e

- Role of this presentation:
  - Give you a practical example of when and how a rule-based engine can be applied.
  - Underline some of the advantages and disadvantages

---

**PHILIPS**

**Agenda**

- Introduction into the assignment (5 min)
- Inference engine based solution (10 min)
- Assessment of the solution and comparison with traditional approach (7 min)
- Conclusions (3 min)
...Alcor...???

• What?
  – uses innovative interaction devices and new interaction concepts
  – develops experience demonstrators

• Why?
  – to prove the feasibility of using innovative interactions in future applications

**Alcor – Problems**

• Development speed (managing complexity)

• Extending & changing an existing demonstrator

• Verification & testing
Alcor Scenario Engine assignment

• Develop a high level definition language to express application behavior (interactions).

• Develop a Scenario Engine component that automates the application behavior execution.

• Develop proof of concept applications

• Why? (key drivers)
  – Better support for rapid application prototyping
  – Increased maintainability (changeability and testability)
  – Usability (software developers, designers need to use this solution)

• Constraints
  – Latency
  – Interoperability

Agenda

• Introduction into the assignment (5 min)
• Inference engine based solution (10 min)
• Assessment of the solution and comparison with traditional approach (7 min)
• Conclusions (3 min)
ALCOR - Initial Problem and Solution

ALCOR – expected outcome
Rule-based reasoning

- In a rule-based approach:
  - The problem-dependent set of data declarations (called the **knowledge base/rule base**) is expressed in the rule-based language.
  - The problem independent program uses an inference mechanism (called the **inference engine**) to derive conclusions from premises.
Rule-based System

• What is?
  – A rule-based system is a system that uses rules to derive conclusions (actions) from premises (conditions and events)

• What is it made out of?
  – An inference engine
  – A rule base/ knowledge base
  – A working memory

The inference engine

• Implements a pattern matching algorithm (most commonly: RETE) to determine which rules apply given the current status of the working memory.

• Implements an agenda, which stores the rules that apply and which uses a conflict resolution strategy to determine which rule should be fired first.

• Implements an execution engine that fires the rules.
Rule-based terminology and heuristics

Rule-based world

- Rule
- Fact
- Template
- Data-driven execution

Metaphor in OO programming

- A static if-then or switch construction
- Instance of a class of objects
- Class
- Sequential execution

Rule-based systems are natively non-deterministic.

On applying a rule-based solution

I chose CLIPS (C Language Integrated Production System) because
- It is developed in C (speed)
- It can be embedded in any C++/Java/Perl/C# application
- It has support for rule-based, procedural and OO programming
- It is mature (20 years), documented, supported
- It is free (LGPL license), approved by IP&S

The ScenarioEngine is the “borderline” component that ensures the communication between CLIPS and the rest of the C# code.
**PHILIPS**

**CLIPS and the ScenarioEngine**

**PHILIPS**

**Example Simple Application**

```
RotatingScreen
HD = "screen"
+isimgSmall(); bool
+rotateX(in degrees : int : void)

rotation.clp
```

```
declare screen
(slot mode
 (type STRING))
)

declare v_evt
(slot command
 (type STRING)
 (default ?NONE))
)

declare rotationRule
(screen (mode "rotation_mode")
(v_evt (command ?v : (eq (str-compare ?c "rotate") 0))
 (test (= \condition (isimgSmall ?f))
 (action "screen" rotate 90)
))
```
Agenda

- Introduction into the assignment (5 min)
- Inference engine based solution (10 min)
- Assessment of the solution and comparison with traditional approach (7 min)
- Conclusions (3 min)

Redesigned & Implemented an existing Demo

- Proved that the proposed scenario engine solution can be used for complex real ALCOR Demonstrators.
- Offered a reference application to the ALCOR team for future ALCOR demonstrators’ development
- Performance
- Reusable & Extensible design of applications
- Maintainability and Rapid Application Prototyping
Scenario Engine Throughput

• In the first four tests, old approach is about 8 times faster than the new one.
• In the last two tests the two approaches are comparable
• The decision making in the scenario engine approach takes on average 1 ms/event.
• The bottleneck for execution time is the medical platform
• The decision time is about 10% of the entire execution time.
• We conclude that the scenario engine approach is not a hindrance in achieving the desired latency.

Maintainability (1)

• Criteria (what influences maintainability)
  – Size
  – Localization of change
  – Clear/concise mapping
• Analyzability
• Verifiability
• Forces explicitness
• Completeness
• Reliability
Maintainability (2) - Size

- Smaller size => higher maintainability
- The ScenarioEngine approach contains 7 times less lines of C# code; however it adds to the C# lines of code, 50% lines of CLIPS rules.

Rapid Application Prototyping

- Criteria (what influences RAP?)
  - Modularity
  - Scalability
    - High-level of abstraction
- Speed
- Reliability of what is developed
Usability?

- Very hard to assess – time and practice can prove it or usability tests (any volunteers?)

- Usability Drawbacks:
  - Developers need to perform mental switches between C# and CLIPS. The two languages are used concurrently.
  - For CLIPS development – mathematical logics awareness is required as well as insight in the heuristics of rule-based reasoning.

- Usability Advantages:
  - It is more straightforward, the level of abstraction (being higher) is closer to our mental models.
  - It can be easier accessible for people with other backgrounds: designers (with understanding of predicate logics)

Agenda

- Introduction into the assignment (5 min)
- Inference engine based solution (15 min)
- Assessment of the solution and comparison with traditional approach (7 min)
- Conclusions (3 min)
Scenario Engine - Conclusions

- Yes, a rule-based language is a valid option for specifying UI interactions (state transitions become rules)
- Yes, an inference engine is a valid option for managing complex UI behavior (especially if it is evolving and changing)
- The solution provides support for rapid application prototyping and better maintainable applications!
- Usability