Visually representing data-driven analysis using state diagrams

Simon Delisle
Michel Dagenais

Progress Report Meeting
May 2015
Presentation plan

- Introduction
- Related Work
- Architecture
- Tool presentation
- Applications and use cases
- Future work
Introduction

- Trace analysis to find performance issues
- Different tools available (Jumpshot, Trace Compass)
- Force users to use specific analysis and trace type
Introduction - Goal

- Trace Compass → Data-driven analysis with XML specification
- User-friendly capture of all the data-driven trace analysis information
Related Work

- Trace analysis
- Languages
  - DSL
- Modeling
  - Papyrus
  - Frameworks
Trace analysis

- **State system method**
  - Build a state machine from trace event
  - Often uses an hard-coded state provider

- **States inside traces**
  - SLOG2 and Jumpshot
  - Good technique to show information in Gantt views
  - Limited to a trace type
Languages

- **Declarative language**
  - Snort
  - Easily modeled

- **Imperative language**
  - DTrace and SystemTap
  - No graphical representation

- **Automata-based language**
  - STATL
• **Domain specific language (DSL)**
  - Declarative language
  - Solve domain specific problems
  - **Benefits**
    - Domain expert can understand, validate and develop DSL programs
  - **Disadvantages**
    - Need to maintain the languages
    - User must learn a new language
Papyrus

- Modeling tool
- UML support
- Eclipse based
- Create DSL
Modeling Frameworks

- GMF (Graphical Modeling Framework)
  - Used to build other graphical tools

- Graphiti
  - Relatively new
  - GMF alternative
  - Easy to use and learn

- Sirius
  - Built on top of GMF
  - No programming to build our tool
Architecture

- Old fashioned

- New way
- Based on UML
- Pseudostate
  - Initial
  - Condition
- States
- Transition
  - State changes
Attribute Tree

- Tree based editor
- Used in the state machine editor
  - Specify the attribute that will be changed with the transition
- Can be reused
Applications and Use Cases

- Kernel state machine
Applications and Use Cases

- Analyse Trace Compass itself
  - Bad requests
Future Work

- Synchronisation between views
- Define filter and pattern
- Better handling of all files that are used to define analysis
Conclusion

- We have a complete editor to create analysis
- A more efficient way to create a state provider