SharpLudus revisited: From ad hoc and monolithic digital game DSLs to effectively customized DSM approaches

Andre W B Furtado, Andre L M Santos, Geber L Ramalho
{awbf, alms, glr}@cin.ufpe.br
Federal University of Pernambuco

DSM Workshop, October 2011
Peculiarities of Game Development

- Non-traditional Requirements Engineering
- Genres: popular yet treacherous concept
- Multi-discipline scope
- Sample availability
- Incipient study on SPL applicability
- Game engines
The road to abstraction and automation [Roberts & Johnson, 1996]:

1. A number of systems is created for a given domain
   - Abstractions are identified
   - Patterns are documented

2. A runtime (framework/servers) enables reusing patterns and abstractions

3. Languages and tools are created to support the runtime

Game engines are here!
First attempt

- SharpLudus Adventure Games SPL, 2006

SharpLudus: improvement areas

- It reported an ad hoc experience, not a systematic process
- Constrained to adventure games
- One macro DSL conceived, no sub-domains
SharpLudus: improvement areas

- No guidelines for domain analysis
- Limited input to DSL design (product line description and domain vocabulary)
- Validation through case study only
Domain-Specific Game Development

5y later...

Envisioning the Game Domain

racing domain?

shoot’em up domain?
Envisioning the Game Domain
Set expectations for Core Game Dimensions

- Player
- Graphics
- Flow
- Entities
- Events
- Input
- Audio
- Physics
- AI
- Networking
- Custom
Analyzing the Game Domain

Partition Game Domain into Sub-Domains

- Ex.: Arcade ➔ Bottom-up Shooter + Maze + ...
Analyzing the Game Domain
Assess Game Domain Automation Potential

- **Identify** sub-domain candidates for automation
  - Those are grainier and sub-atomic (scene/screen transition, collision relationship between entities, heads-up display, etc.)

- **Inputs:** feature model, domain expert knowledge, core game dimensions

- **Investigate** modularization of types in reference architecture and implementations

- **Investigate** code repetition in reference architecture and implementations
Analyzing the Game Domain
Assess Game Domain Automation Potential

**Prioritize** sub-domain candidates for automation, considering:

- Previous automation evidence (tools/languages)
- Feature coverage
- Development productivity if the sub-domain is automated (size of generated artifacts)
- Development abstraction if the sub-domain is automated (complexity / error-proneness of generated artifacts)
Game Domain Analysis

Application Core Assets

- Domain-Specific Game Architecture
  - Centerpiece: game engines
  - Peripheral pieces: reusable game components

- game engine promoted to domain framework
- generated or manual code
- adapter
Game Domain Analysis

- Development Core Assets
  - Characterize sub-domain variability

```
routine configuration

domain, sub-domains

creative construction

wizards
features tool, configurator
visual DSLs, tool integration
```
Game Domain Analysis

➡ Development Core Assets

Define DSLs, transformations and supporting assets

Design and implement IDE integration
Case Study: ArcadEx

- Game SPL for arcade games
Case Study: ArcadEx

- 150+ domain features
Case Study: ArcadEx

- Domain-Specific Game Architecture
ArcadEx sub-domains and DSLs

- GameDefinitionDSL

screen transition, game properties, screen background, background music
ArcadEx sub-domains and DSLs

- EntityDSL

entity definition, collision interest, input handlers, event reactions, entity-based timer events
ArcadEx sub-domains and DSLs

- ScreenDSL

heads-up displays, placement of entity instances, screen-based timer events
ArcadEx sub-domains and DSLs

- InputMappingDSL

keyboard ↔ gamepad
input mapping
Evaluation and future work

4-5x faster development
- Inline with DSM in other domains

Reduced flexibility at first, then:
- Extensibility hooks
- Enhanced built-in support by new SPL versions

For the future
- Controlled experiments
- Evaluate applicability for other domains
Conclusions

- Reduced complexity for consuming game engines
- Breakdown of game development tasks into more granular and automatable chunks
- Incremental delivery of value for prioritized game sub-domains
- Flexibility and extensibility for unforeseen behaviors
SharpLudus revisited: From ad hoc and monolithic digital game DSLs to effectively customized DSM approaches

Andre W B Furtado, Andre L M Santos, Geber L Ramalho
{awbf, alms, glr}@cin.ufpe.br
Federal University of Pernambuco

DSM Workshop, October 2011