

Software design document

This document provides an overview of the features that will be implemented in **2.2 Step-based assessment asset**. This asset consists of a parser and a reasoner.

Description

The following figure schematically describes an implementation framework for scenario based games. It shows the relation between the two assets from Utrecht University that will be delivered to RAGE, 3.3 Communication Scenario Editor & 2.2 Step-based assessment. It additionally shows how game developers may use the assets.

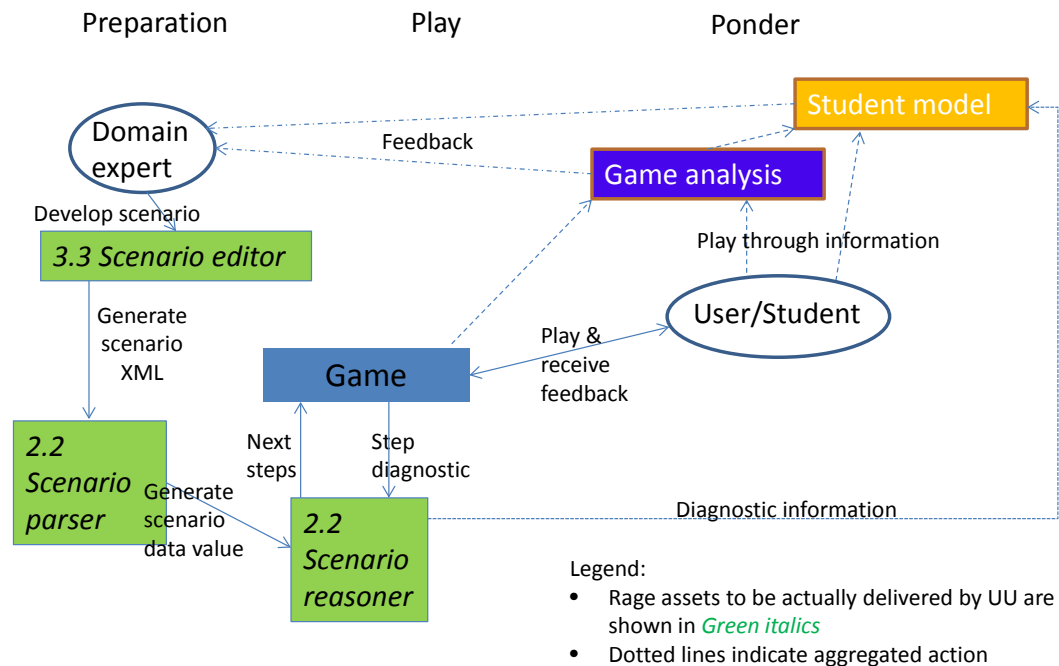


Figure 1: An implementation architecture of scenario-based simulations

A communication-skills teacher develops a scenario in the scenario editor as a graph of steps along with the respective scores and feedback per step. The editor generates valid scenarios in xml format.

How to use this asset

The scenarios in XML format generated by the editor are parsed; the resulting output is a scenario data-type value. This is required one-time per scenario.

The scenario reasoned is a web-service that offers its services through JSON-RPC or XML-RPC. A game (to be developed by a game company) interacts at run-time with the generic scenario reasoner (with the ID of the specific parsed scenario), which provides information about the possible following steps at each step in the series of interactions. Incremental scores are also fed-back by the scenario reasoner to the game.

The two assets 3.3 Communication Scenario Editor & 2.2 Step-based assessment are typically used together in a game. It is of course possible to use the XML output from the editor; however in that case the game-developer needs to develop their own “parser” and “reasoner” modules.

The asset is implemented in Haskell and the Ideas framework from Utrecht University. The Ideas framework has been tested extensively both functionally and non-functionally and used in diverse domain reasoners (DME secondary math education, Math-Bridge, MathDox, Logic tool, Ask-Elle, tutor for Haskell) in addition to the communication reasoner.

Timelines

See planning overview in a Google doc.

(https://docs.google.com/spreadsheets/d/1w13P0bUEvYlrKu0KRxli_HUHXFHTXITqhVbu9cCVaYs/edit#gid=398706816).

Additionally a Kan-Ban board is used to track progress on specific items, a snapshot is provided.

Initial evaluation release (end of Feb 2016)

- All the above mentioned features will be part of the initial release.
- Quality aspects WP2
 - Repository location: (<https://github.com/UURAGE/ScenarioReasoner>)
 - Running and building instructions in Git.
 - API and demonstration of use: An example of a C# and Javascript to call this asset will be provided. Separate Git location under construction with a sample scenario.
 - Deploy instructions in Git.
 - Documentation in Git.
 - Tutorials
 - API interaction <https://www.youtube.com/watch?v=t5V5POF1FUc>
 - Javascript sample <https://www.youtube.com/watch?v=tOc0P0tT1yE>
 - Platform requirement in Git.
 - Technical Report Ideas (UU-CS-2014-005) including test report available on request.

2nd Release (end of June 2016)

- Major redesign of the XML structure based on the RAGE user-requirements. The parser and reasoner need to be adapted for the changed scenario XML output.
- Specific features planned for implementation are the following:
 - Multiple computer statements (for INESC emotional appraisal assets).
 - Multiple virtual characters (for OKKAM).
 - We have a biweekly meeting with all people involved. New changes/features are evaluated and decided upon from a user-, development-, and architecture-perspective.
- Focus on additional WP2 quality aspects (system tests, profiling, load tests).
- Investigate A2 (T2.4A) UCM asset use.
- Develop a client-only based (i.e. no server side) application of the reasoner. (Rage game developer wish.)
- Make asset Rage-repository ready (in co-operation with OUNL).

Snapshot KanBan Flow (extracted 26 Feb 2016)

https://kanbanflow.com/board/e61fde83c7b1bcd97bbacfc96c37e43f

Apps ★ Bookmarks Archive Sports Travel Writing Care Acting Painting Global Nomad IBM PhD rlaia

KanbanFlow UURAGE Administration Raja Lala

| Backlog | Priority | In progress 1 / 5 | Done |
|--|--|--|--|
| <div>Unit tests for scenario editor</div> <div>editor test</div> <div>Document Unit Test Coverage for scenario editor</div> <div>editor test</div> <div>Game Integration Tutorial for scenario editor</div> <div>Johan Jeuring editor</div> <div>Configuration tutorial for scenario editor</div> <div>editor</div> <div>Document Unit Test Coverage for scenario reasoner</div> <div>reasoner test</div> <div>Unit tests for scenario reasoner</div> <div>reasoner test</div> <div>Reliability Tests for scenario reasoner</div> <div>reasoner test</div> <div>Profiling scenario reasoner</div> <div>reasoner</div> <div>Profiling scenario editor</div> <div>editor</div> <div>Use UCM Server-side sign-on / ... tool (project requirement)</div> <div>Raja Lala reasoner</div> <div>Inter Asset integration (3D characters and AI)</div> <div>Raja Lala Raja</div> | <div>Installation tutorial for scenario editor (README)</div> <div>Marcel van Geest editor</div> <div>Due: Monday 17:00 (Done)</div> <div>Installation tutorial for scenario reasoner (README)</div> <div>Marcel van Geest reasoner</div> <div>Due: Monday 17:00 (Done)</div> <div>Game Integration Tutorial for scenario reasoner</div> <div>reasoner</div> <div>Due: Monday 17:00 (Done)</div> | <div>Encoding defect in reasoner</div> <div>Marcel van Geest reasoner Raja</div> <div>The reasoner returns doubly encoded text for special characters (e.g. !). This defect is 'corrected' (in a rather inelegant fashion) in the game by decodeUTF8, which uses the now deprecated function escape.</div> | <div>Today</div> <div>Tuesday, 23 February</div> <div>ScenarioParser aan C# integration asset toevoegen</div> <div>Jordy van Dortmont player Jordy</div> <div>Monday, 22 February</div> <div>Deployment of UURAGE editor on production server</div> <div>Marcel van Geest editor</div> <div>Friday, 19 February</div> <div>Add UURAGE/ScenarioEditor remote to www repository and try to cherry pick commits and push them to UURAGE/ScenarioEditor/master</div> <div>Jordy van Dortmont editor Jordy van Dortmont</div> <div>Thursday, 18 February</div> <div>Installation package for scenario editor</div> <div>editor</div> <div>Installation package for scenario reasoner</div> <div>reasoner</div> <div>Create new instance of the scenario editor on test server</div> <div>Jordy van Dortmont editor</div> <div>Create ScenarioEditor repository</div> <div>Jordy van Dortmont editor</div> <div>Create new common branch for merging with both the UURAGE branch and the Communicate! branches</div> |

Timer Reports Filter

Get Premium Help & feedback

