Software design document

This document provides an overview of the features that will be implemented in **3.3 Communication Scenario Editor.**

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.

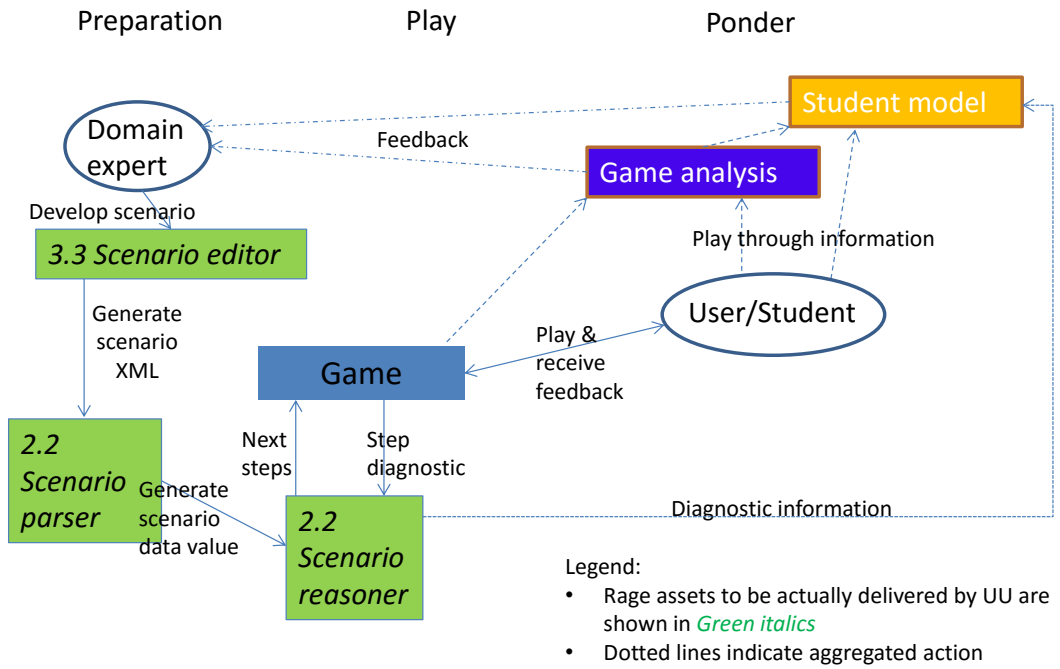
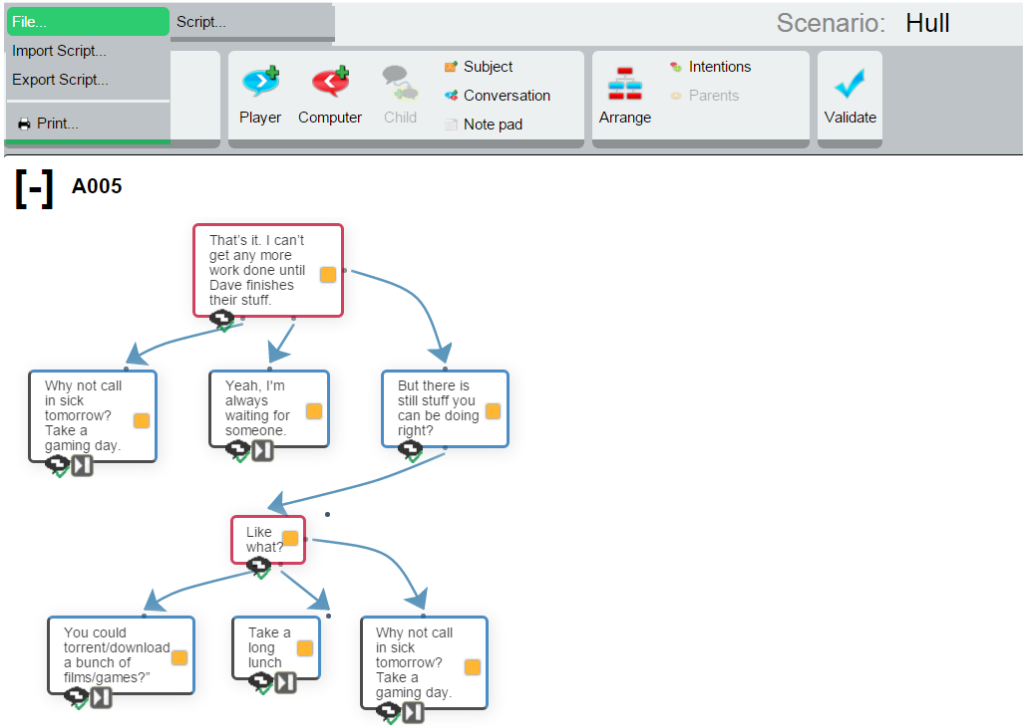


Figure 1: An implementation architecture of scenario-based simulations

How to use this asset

A domain expert develops a scenario in the scenario editor as a graph of steps along with the respective scores and feedback per step. See screen-shot below.



A domain expert can then export valid scenarios in XML format. Scripts may be saved locally and imported to modify and continue development.

The generated XML files can be parsed and interpreted (not part of this asset). See also 2.2 Step-based assessment, a UU asset that parses and 'reasons' per step. Typically, these assets are used together in a game; though Game developers may develop their own parser and reasoner.

The Communication Scenario Editor asset is implemented in Javascript and runs on most browsers, eliminating the need for a native client.

Timelines

See planning overview in a Google doc.
(https://docs.google.com/spreadsheets/d/1OoZUDLiFBoPYWcHcuK4poz81sq0gladvGawn6vlXc-M/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 WP3
    - Repository location: (https://github.com/UURAGE/ScenarioEditor)
    - Running and building instructions in Git.
    - Deploy instructions in Git.
    - Documentation in Git.
    - Tutorials
        - Scenario Properties (https://youtu.be/NIJwMnZssDw). This video explains the how to set the properties, e.g. starting screen, background, characters, and feedback type.
        - Scenario Learning Goals (https://youtu.be/Q3e79I3jJbs). This shows how to set the learning goals such as negotiation and empathy, and associated scoring.
        - First Scenario (https://youtu.be/eaLblvbmPOU). This video explains how to create a scenario.

        - Interleaving Scenarios (https://youtu.be/gyEjh86xrDo). Interleaving is an advanced feature of our scenarios/implementation. This feature is especially useful in the domain of communication skills where students may have to perform multiple (sub)tasks, but the order in which these tasks are performed is not important, and steps of the various topics may be interleaved. (For example: a doctor has to ask questions about the location of the pain, the duration of the pain, how the pain is experienced, etc, but usually it is not important that these questions are asked in a particular order).
        - Advanced Scenario Features (https://youtu.be/z15zNnHfFF4). This video explains how to set various conditions, which you can use to steer a conversation.
    - Platform requirements: works on most browsers.
    - Profiling.

2nd Release (end of June 2016)

- Redesign of the asset and XML structure based on the RAGE user-requirements.
- Specific planned features are:
    - 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 WP3 quality aspects (code coverage, system tests, profiling, load tests).
- Make asset Rage-repository ready (in co-operation with OUNL).

Snapshot KanBan Flow (extracted 26 Feb 2016)