KOMEGA REQUIREMENTS No.4, Version 2
Basic Application Scenario
[*]

Gerd Doeben-Henisch
gerd@doeben-henisch
in cooperation with the INM KOMeGA-Team and especially with Tobias Schmitt

August 2, 2020

Abstract

This text describes the basic requirements for the komega software project, which is part of a larger project in the domain of an applied cultural anthropology. This is version 2 of the basic requirements No.4 which continues No.1-v3, No.2-v1, No.3-v1, No4-v1.

1 General Requirements, Actor Story [AS], and System-Layout Overview

Figure 1 outlines the *general requirements*, figure 2 translates these requirements in an actor-actor interaction story – short: *Actor Story [AS]* –, and figure 3 correlates the actor story with the *system layout* distinguished according to two main cases: (i) in the lower part it is outlined how the *local development scenario* looks like and (ii) in the upper part you see the *real application situation* where the users are interacting with a web-browser which is connected to a web-server realized with a python based web-server django including the main python program, some java-libraries for the web-browser appearance as well a sql-database (MYSQL).

Basic Idea Developmental Programming: Following the vision of the actor story then the whole story consists of clearly distinguishable individual tasks (Q, S, X, PS, IS, E) which can freely being repeated as often as the user wants it. The interaction of the user with the system will be kept simple offering only a

[*] Copyright 2020 by eJournal uffmm.org, ISSN 2567-6458, Email: info@uffmm.org, Publication date: August 28, 2020

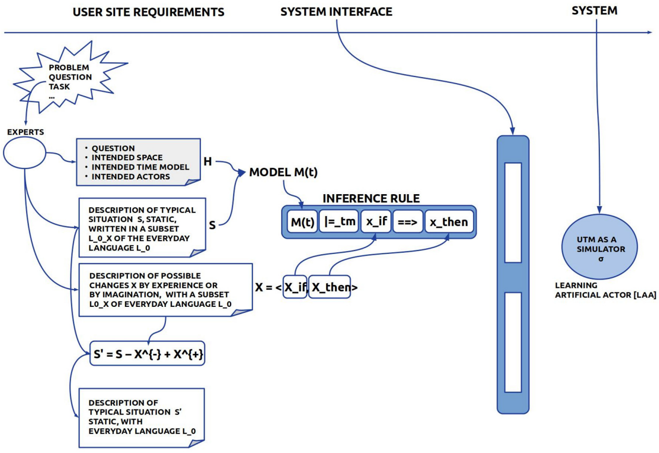
USER SITE REQUIREMENTS
SYSTEM INTERFACE
SYSTEM
PROBLEM
QUESTION
TASK
...
EXPERTS
• QUESTION
• INTENDED SPACE
• INTENDED TIME MODEL
• INTENDED ACTORS
H
MODEL M(t)
DESCRIPTION OF TYPICAL SITUATION S, STATIC, WRITTEN IN A SUBSET L_0_X OF THE EVERYDAY LANGUAGE L_0
S
INFERENCE RULE
M(t)
|=_tm
x_if
==>
x_then
DESCRIPTION OF POSSIBLE CHANGES X BY EXPERIENCE OR BY IMAGINATION, WITH A SUBSET L0_X OF EVERYDAY LANGUAGE L_0
X = <X_if, X_then>
S' = S – X^{-} + X^{+}
DESCRIPTION OF TYPICAL SITUATION S' STATIC, WITH EVERYDAY LANGUAGE L_0
UTM AS A SIMULATOR σ
LEARNING
ARTIFICIAL ACTOR [LAA]

Figure 1: General Requirements with System Overview

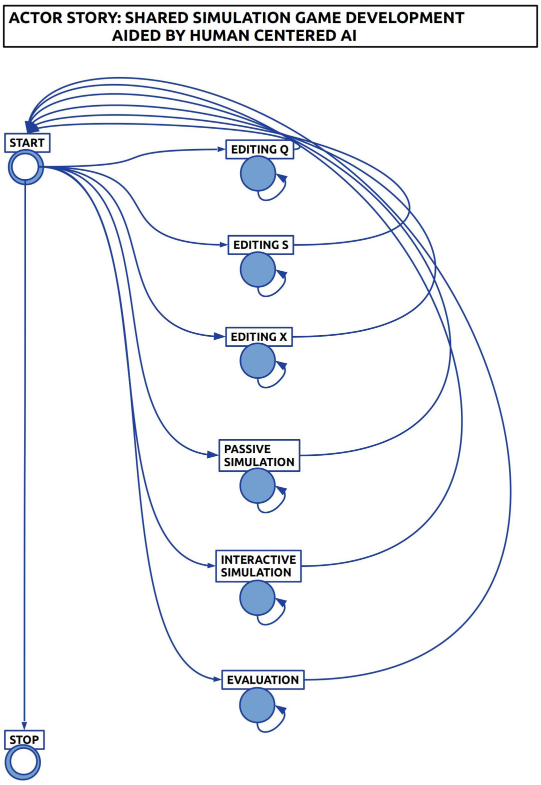

Figure 2: Actor Story Overview - v2

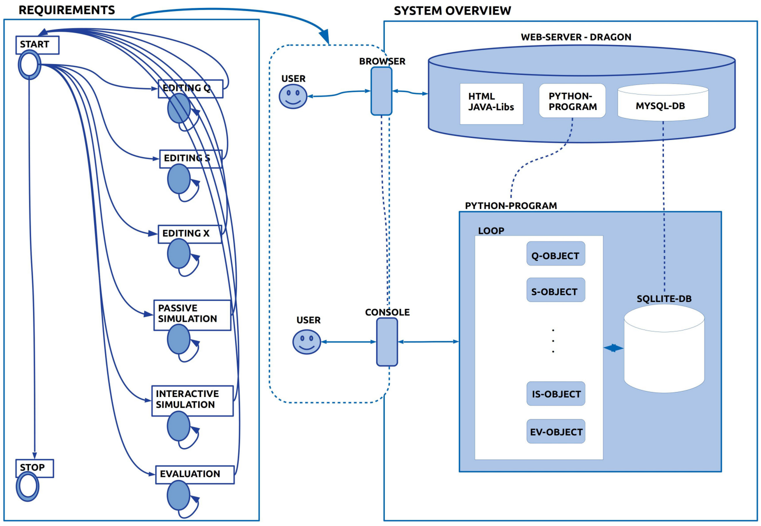

Figure 3: Overview: Requirements left and System Layout right

console input and output. The individual tasks will be served by corresponding objects which provide all the data and all the operations which are needed for a task. Every individual task mediated by its object is associated with a database – during development the sqllite database which is part of python3 – where results of the task can persistently be stored or can be reloaded on demand.

Basic Idea Real Application: In the real application the user will directly interact with a web-browser which mimics an advanced console interface (which easily can be extend with all kinds of graphic). As database the web-server – the python-based django web-server – will use a full sql-database (preferred mysql).

2 Milestones

The *first* milestone is the implementation of the tasks {Q, S, X, PS} until end of October 2020, the *second* milestone the extension with {IS,E} until end of December 2020.

It follows then a phase of intensive testing until March 2021 with accompanying improvements.

The first major release is reached if multiple groups can work in parallel with the server through a graphical system interface [GSI].[1]

[1] The widely spread usage of the wording *graphical user interface [GUI]* is highly misleading

because indeed the interface is the interface of the system. The user as such has also an interface, the *user interface [UI]*, but this is in most HMI applications only badly investigated and only rarely formally specified.