

# KOMEGA REQUIREMENTS No.4, Version 1

## Basic Application Scenario

\*

Gerd Doeben-Henisch  
gerd@doeben-henisch  
in cooperation with the INM KOMeGA-Teams

August 26, 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 1 of the basic requirements No.4 which continues No.1-v3, No.2-v1, and No.3-v1

## 1 Actor Story [AS] Overview

Figure 1 shows the general requirements for the whole application indicating a necessary *system interface [SI]* as well as a system behind the system interface, the *simulator  $\sigma$* , which is a *learning artificial actor [LAA]* realized an *universal Turing machine [UTM]*.

In figure 2 one sees an overview of the main states of the actor story [AS]. As one can recognize directly is the overall format very modular and flexible. All different states are directly accessible and within the processing of a state one can activate some correction mode.

## 2 Start State

**Task:** Overview about all possible states.

**Actors:** Human experts.

---

\*Copyright 2020 by eJournal uffmm.org, ISSN 2567-6458, Email: info@uffmm.org, Publication date: August 26, 2020

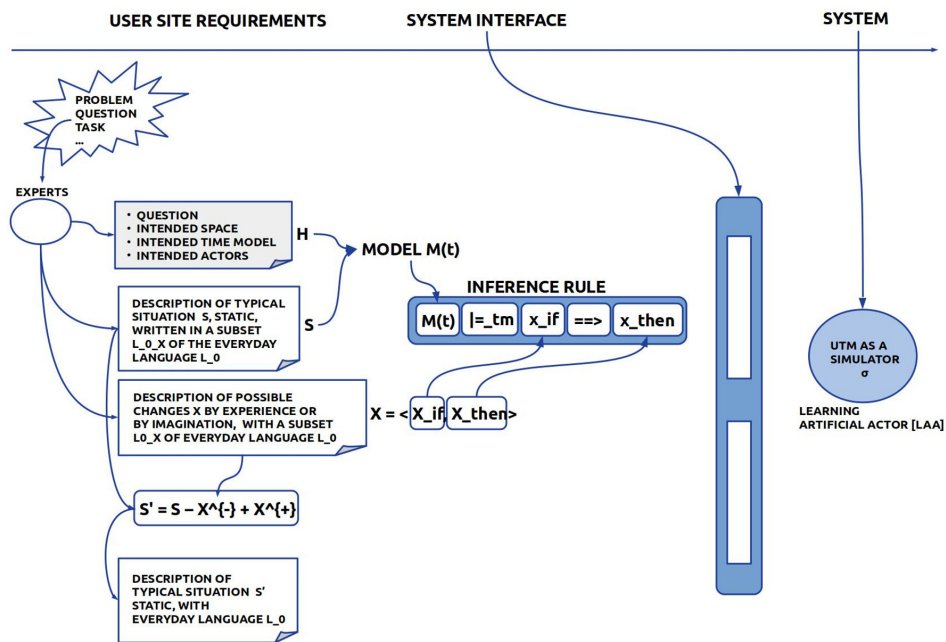


Figure 1: General Requirements with System Overview

**SI:** A main window W1 with a menu showing all possible options to select. A second window W2 showing the history since start so far: which states have been served so far and what is the actual result.

**Actions:** Select an option.

**General Behavior:** If one of the states which are reachable from the start state will be finished then the user is automatically brought back to the start state.

### 3 Editing Q State

**Task:** Input all data which are necessary for the Q-state.

**Actors:** Human experts.

**SI:** A main window W1 with a menu showing all possible questions to be answered.

**Actions:** Select every question and write an answer.

### 4 Editing S State

**Task:** Input all data which are necessary for the S-state.

**ACTOR STORY: SHARED SIMULATION GAME DEVELOPMENT  
AIDED BY HUMAN CENTERED AI**

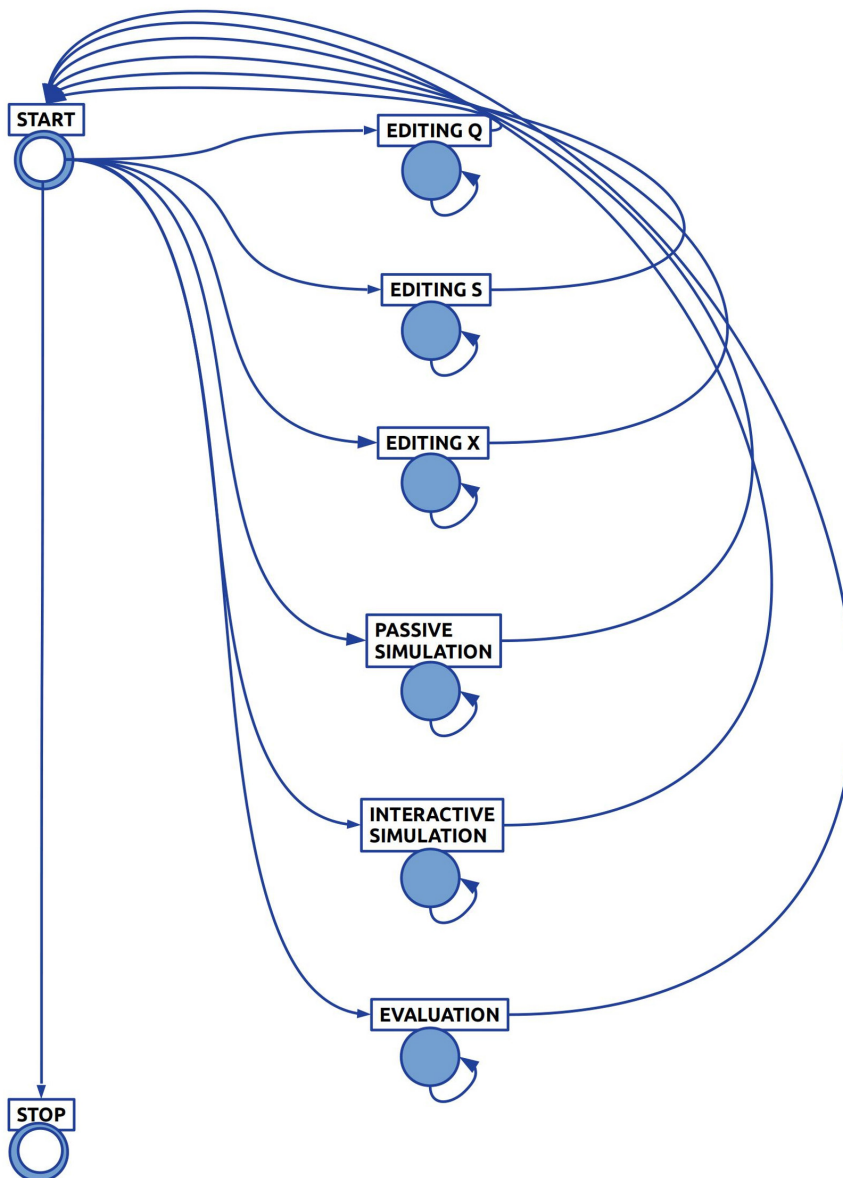


Figure 2: Actor Story Overview - v2

**Actors:** Human experts.

**SI:** A main window W1 offering the editing of a text consisting of individual statements. Every statement can be edited separately and repeatedly.

**Actions:** Select either a given statement for editing or edit a new statement.

## 5 Editing X State

**Task:** Input all data which are necessary for the X-state.

**Actors:** Human experts.

**SI:** A main window W1 offering the editing of a text consisting of individual statements. Every statement can be edited separately and repeatedly. Every statement has the format 'IF ... THEN ...' according to the theory.

**Actions:** Select either a given statement for editing or edit a new statement.

## 6 Passive Simulation State

**Task:** The simulator  $\sigma$  shall compute a series of states starting with the state  $S_0$ . The simulation will stop either after a given number of loops or by an user caused interrupt.

**Actors:** Human experts.

**SI:** After starting the simulation one sees two windows: W1 shows the actual state and W2 shows the rules which will be applied.

**Actions:** The simulator computes with an appropriate subset of the change rules  $X^* \subseteq X$  a new state  $S'$  which then becomes the new actual state  $S$ . The velocity of the simulation depends either from a pre-set parameter or of the interrupt by the user. If the simulator is unable to determine whether a certain change rule  $\xi \in X$  can be applied to the actual state  $S$  then the simulator asks the human experts for a judgment. In this case the simulator  $\sigma$  is said to act in *training's mode*.

## 7 Interactive Simulation State

**Task:** The simulator  $\sigma$  shall compute a series of states starting with the state  $S_0$ . The simulation will stop either after a given number of loops or by an user caused interrupt.

**Actors:** Human experts.

**SI:** After starting the simulation one sees two windows: W1 shows the actual state and W2 asks the different players for their input.

**Actions:** The simulator *unifies* all the different inputs and given rules to the *now actual set of selected change rules*  $X^*$  and computes a new state  $S'$  which then becomes the new actual state  $S$ . The velocity of the simulation depends either from a pre-set parameter or of the interrupt by the user. If the simulator is unable to determine whether a certain change rule  $\xi \in X$  can be applied to the actual state  $S$  then the simulator asks the human experts for a judgment. In this case the simulator  $\sigma$  is said to act in *training's mode*.

## 8 Evaluation State

**Task:** After a simulation the experts have the possibility to analyze the simulated process by different criteria.

**Actors:** Human experts.

**SI:** After starting the simulation one sees two windows: W1 shows the possible criteria which can become activated for an evaluation and W2 shows the results with regard to the criteria.

**Actions:** The human experts select those criteria which should be commented by the system and read then the output.

## 9 Stop State

**Task:** End the process.

**Actors:** Human experts.

**SI:** Bye Bye window

**Actions:** Quit.