

KOMEGA REQUIREMENTS: Basic Version with optional on-demand Computations by the Computer

*

Gerd Doeben-Henisch (gerd@doeben-henisch)

Oct 27-28, 2020

Abstract

Introducing a new general interface to transfer messages to the simulator as computer to compute explicitly some functions.

1 Human Actors only

Until now any kind of computation associated with the expressions of the used everyday language is done by the communicating human actors themselves. The computer does only manage sets of expressions S, S', X without any deeper understanding. But as one can experience there are many circumstances where the human actors have to manage some bookkeeping (e.g. amount of money earned or spend during the last month), some math computations (e.g. the consumption of energy of a house during the winter time) or to clarify certain facts or complex issues (e.g. the diffusion of some fluids in other fluids). In those and other similar cases it makes sense to use the computer to do this tedious work. To enable this the computer needs all the necessary basic facts and operations applicable to these facts. Because these facts and operations are not available by the expressions of the everyday language as such but only by the implicit *meaning function* μ located in every human speaker-hearer one has to transmit *on demand* exactly those *facts* and *operations*, which are needed for computations by the computer.

Until now no general interface for such information has been defined in the komega-SW! What one needs here is a *general interface for all kinds of information*. Thus we have to generalize the whole picture.

*Copyright 2020 by eJournal uffmm.org, ISSN 2567-6458, Email: info@uffmm.org

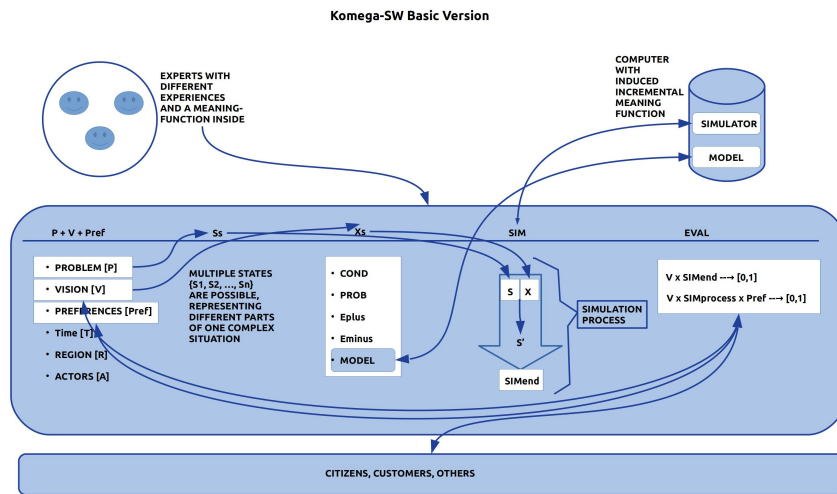


Figure 1: The new general interface as part of the komega-SW in the Basic Version

2 Including the Computer

Figure 1 gives an outline of the whole scenario with the *Basic Version of the komega-SW*. If you would compare this figure with those figure from the last version¹, then you can detect that the *structure of the change rules X* has changed. Besides the standard elements {COND, PROB, Eplus, Eminus} you find now *only one new element* labeled *MODEL*. The MODEL-element is now the required *general interface GI* which allows the user to *transfer* all that kinds of information, that the user wants intentionally to be computed.

In the general case of the general interface *GI* it is difficult to decide, what kind of stuff shall be transferred to the computer. For the basic version we have the following experimental requirements for this general interface:

1. In the problem-vision document D_{PV} you define a minimal *space R* and a *standard clock C₀* which shall be valid throughout your whole simulation. The **required space [R]** is basically given as a *basic coordinate* $[0, 0, 0]$ and all other points or regions are in some distance to the basic coordinate. Usually a 2-dimensional plane will suffice, but a 3-dimensional space is possible too. To relate this to *real world coordinates* one can define a mapping from the basic coordinate $0, 0, 0$ to a real coordinate r, r, r extended with a measure to translate the *unit distance* with a *real distance*. Associated with the required region *R* is a **required standard clock C₀** having the format 0. The standard clock has a *basic tick c₀* which can be counted: $t' = t + c_0$, reading that a point of time t' is the

¹See: <https://www.uffmm.org/2020/10/17/komega-requirements-from-the-minimal-to-the-basic-version/>

result of the last point of time t incremented by one basic tick c_0 . One can *relate* the standard clock C with a *real clock* C_{real} by mapping the standard time root 0_C with some real time stamp $t_{C_{real}}$ and translate a real time time-unit like *minute* with an amount of standard ticks.

2. To allow the user to stay with the used everyday language it is possible to use expressions from the everyday language to describe the wanted computations. The only minimal requirements for user-messages are the following ones:
 - (a) There is an *overall schema* with the sections: {Actors, Action/ Relation, Property, Operation, Unit, Value} For examples look to the section after this enumeration.
 - (b) All used properties are assumed to have the initial value '0'. If this does not fit you have to begin with an initial operation to set the value accordingly.

Examples for General Interface Messages: Given is a simple *state description* S followed by a series of simplified change rules by mentioning only the Eplus mode and the MODEL mode, thereby allowing to change the initial state description:

1. **State:** Gerd is hungry. He wants to go for lunch to the Greek restaurant around the corner. At the same time are two students A and B on the campus wanting to ask Gerd some questions. They know that Gerd often is going to lunch to the Greek restaurant around the corner.
2. **Change:** At 1 pm Gerd is going to the Greek restaurant. MODEL: 'Gerd', 'is going', 'time needed', 'add', 'minutes', '5'.
3. **Change:** Gerd is in the Greek restaurant. Students A and B take a walk to the Greek restaurant. MODEL: 'students A and B', 'walking', 'time needed', 'add', 'minutes', '10'.
4. **Change:** Gerd orders the small menu from the day. It costs 7 Euro. MODEL: 'Gerd', 'is ordering', 'expenses', 'add', 'Euro', '7'. The students A and B arrive at the Greek restaurant.
5. **Change:** Gerd receives the menu and eats. MODEL: 'Gerd', 'eats', 'nutrition', 'add', 'calories', '450'. The two students are ordering. Student A orders a coke. It costs 1.6 Euro. MODEL: 'Student A', 'ordering', 'expenses', 'add', 'Euro', '1.7'. Student B orders a coffee. It costs 2 Euro. MODEL: 'Student B', 'ordering', 'expenses', 'add', 'Euro', '2'.

6. **Change:** Gerd's colleague Peter is entering the restaurant. Peter is taller than Gerd. MODEL: 'Peter, Gerd', 'occurring relation', 'is taller', 'compare', 'logic', 'true'.