KOMEGA REQUIREMENTS No.4, Version 3
Basic Application Scenario
*

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

September 4, 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 3 of the basic requirements No.4 which replaces No.4-v1 and No.4-v2. The actual theoretical background is given with the posts with the title 'Extended Concept for Meaning Based Inferences – Part 2. Version 2'[1] as well as the post with the title 'Actor Epistemology and Semiotics. Version 1'.[2]

1 Actor Epistemology and Semiotics

The task of this text is to describe the requirements which are framing the coding of the theory. Figure 1 delineates the theoretical point of view behind the whole project. The main actor is not the computer but the *human actor* HA. The reason behind this setting is the fact that the human actor is the only real bearer of knowledge on this planet. And this human knowledge has the specialty that it is organized in at least two main dimensions: the knowledge of the real structures of the world – including the own system – F which is completely encoded in neural states NN of the brain, and as part of this basic dimension there is a special subset of knowledge functioning as expressions E of a symbolic language, which are related by an *adaptive meaning function* μ to parts of the knowledge. Communication between different brains is only possible by the expressions E which are transformed into real world events E_{RW} which can be exchanged between different bodies. Without the human actor with its

[1]See: https://www.uffmm.org/2020/09/02/extended-concept-for-meaning-based-inferences-part-2-version

[2]https://www.uffmm.org/2020/09/03/actor-epistemology-and-semiotics-version-1/

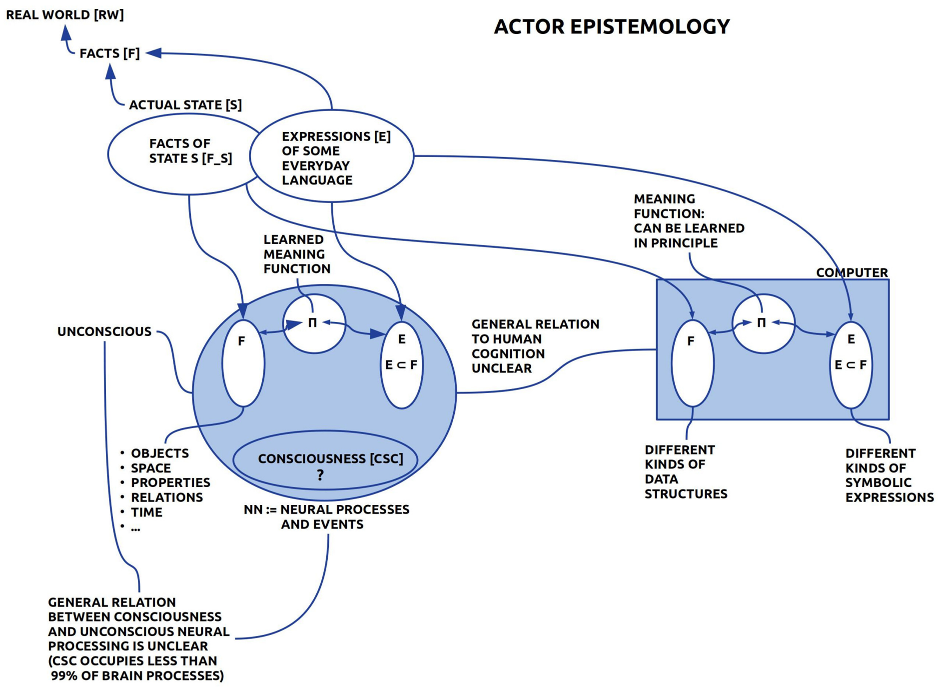

Figure 1: Actor epistemology as main point of view behind the project

built-in and *individually learned* meaning function μ these expressions are without meaning. Therefore any kind of meaningful communication about the real world has to put the human actor in the center of the discourse. Computer can be used to support such a meaningful communication but only in a very limited and specialized way. Nevertheless *as real part of real human communication* computer can be of central importance in the future.

If two different brains want to communicate with each other they have to use expressions E_{RW} embedded in a shared meaning function μ. These expressions can be organized as *texts* TXT which represent a finite set of expressions organized in a linear sequence.

In this project only two kinds of texts will be assumed: First a text which is describing a *state* S of the *real world* RW, either as *actually given* or as *assumed to be possible in the future*. It is assumed that the expressions of such a state description S describe so-called *facts* F_{RW} of the real world, which can be observed and which can be decided by the authors of the text as *being the case*. 'Being the case' is either inferred from these expressions *directly* by exploiting the meaning function μ or these expressions can be *transformed* by finitely many operations again by exploiting the meaning function μ that one can reach expressions which directly can be decided as 'being the case' or not.

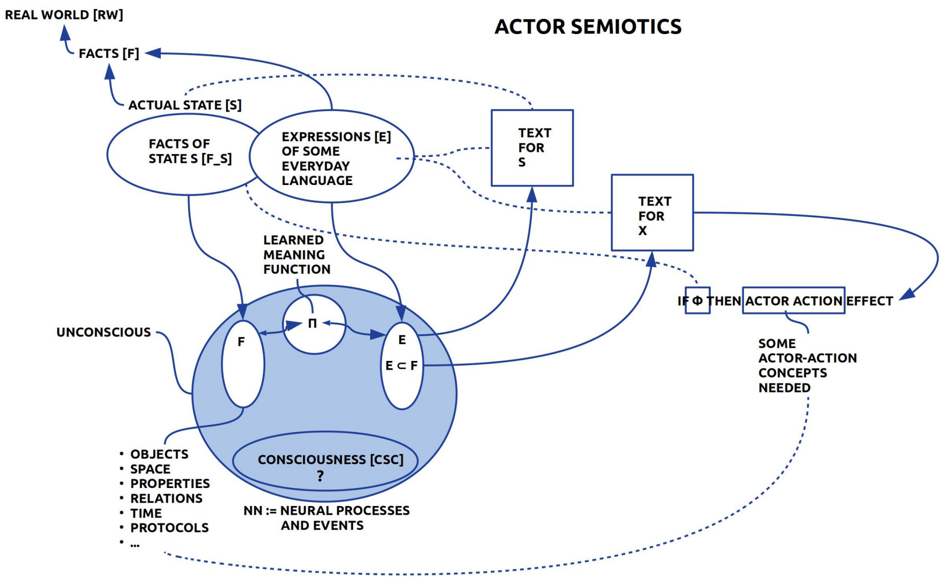

Figure 2: Actor semiotics points to the expressions used by human actors in their communications which are functioning as meaningful signs enabled by a working meaning function

In this context it is assumed that a text S is assumed to be *true* because all its expressions are assumed to be *true*. *Being true* means, that the uttering human actors can relate these expressions E_{RW} by their meaning functions to internal neural correlates F_{NN} of the facts which are perceived and learned from the real world facts F_{RW}. Only human actors can manage this kind of *truth*.

Second there is another text which is describing the assumed *dynamics* of the state S which is done by a finite set of *change rules* X. The general assumption behind the concept of change rules is given in the idea that a state description S represents a certain finite *time slot* T_{Δ} in an assumed time model T and that parts of the state S can *change* thus that a *new state* S' occurs. To that extend one can *imagine* such *possible* changes one is able to have a vague *look into some possible future state*. Considering this case of possible changes one has to determine the possible *causes of change*. Besides those cases where one *does not know* what are the causes for observable changes it is assumed in this text that *every identified actor* can be a possible cause for change.

2 Actors

In figure 3 three main types of actors are distinguished. This list is possibly not complete. If other types of actors will be identified in the future then the list

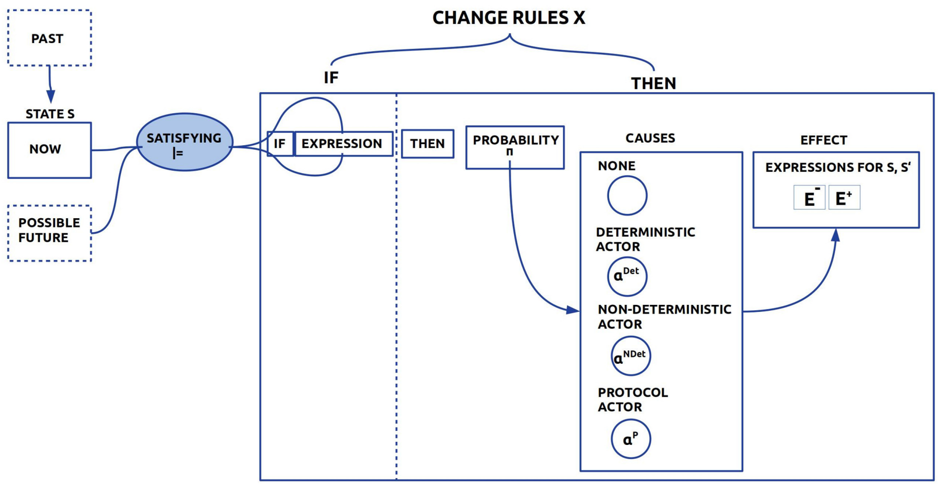

Figure 3: General format of a change rule $\xi \in X$

has to be extended.

Deterministic Actor: A *deterministic actor* α^{Det} is assumed to be an input-output system with a *behavior function* ϕ_{Det} which determines which kind of *output* O will occur if a certain kind of *input* I happens, written as

$$\phi_{Det} \quad : \quad I \longmapsto O \tag{1}$$

If one knows the behavior function ϕ_{Det} then one can predict the output of the system if one knows which kind of input is actually given. This can be written as a change rule like 'IF expression Φ is true THEN actor α^{Det} will act with ϕ_{Det} and thereby the actual state S will change ba removing some fact and/or by adding some fact. Example: Look to a light-device in your flat: pressing the light-button will either turn the light on of off.

Non-Deterministic Actor: A *non-deterministic actor* α^{NDet} is like a deterministic actor but with a difference: the output is not only depending from the input but also from the actual *internal states* IS. Thus the behavior function ϕ_{NDet} is written as

$$\phi_{NDet} \quad : \quad I \times IS \longmapsto O \tag{2}$$

Because one usually can not exactly know which internal states IS are actually active it is not possible to predict the output of the system clearly.

Examples of non-deterministic actors are all machines whose parts can change during live-time. These changes can cause a change in the overall behavior function because individual parts will change their role in the whole system such, that the internal states are changing. Example: Your car has a damaged cable; you are out of gasoline. The material of a bridge is changed by age. Engineers are speaking here of the *reliability* of the system which is depending from the complexity of the machine as well as from the time-depending changes of the material.

Learning Actor: A sub-case of non-deterministic actors are the *learning actors*. These non-deterministic actors can change their internal states in the light of their experience, written as:

$$\phi_{NLDet} \quad : \quad I \times IS \longmapsto IS \times O \tag{3}$$

Thus the internal changes can happen all the time and therefor it is difficult if not impossible to predict the behavior of such an actor precisely. All kinds of biological system have this learning ability. That it is possible to predict the behavior of biological systems nevertheless is due to the fact that biological systems are using their learning ability not to become always different but to *optimize* their behavior for certain *goals*. Such a goal-directed behavior can be quite stable as long as the goals are stable and the result of this behavior for the actors.

Protocol Actor: A social type of actor is the *protocol actor* α^P. A protocol actor is a *virtual actor* consisting of at least two real actors which have *agreed* to follow a certain *protocol* P for some time T_{Delta}. A protocol is a text with a finite list of change rules which have to be served in the linear order of the protocol. The acting actors are still the individual real actors, but because they *know* about the protocol and they have *agreed* to follow the protocol the pattern of their actions *contains a subset of actions which conform to the protocol*, written as:

$$\phi_P \quad : \quad A \times P \times S \longmapsto ACT \times EF \tag{4}$$

Thus the participating actors $\alpha^P \subseteq A$, knowing an agreed protocol $p \in P$ and being part of the state S will do some action $a \in ACT$ and thereby they will cause some *effect* EF which can change the state S. Examples are situations where you enter some store to buy something. *Buying* in a store something follows certain unwritten conventions functioning like a protocol. More officially you will marry another person, then you have to proceed according certain rules (depending from the state you are in). Or you will get a degree from a school or university. In all these cases you have to follow some – unwritten or written – rules as valid between you and some other persons.

3 Application Scenario

As application scenario the following is assumed in this text: It is assumed that there exist a *group of experts* – this can be everybody – which want to *share their knowledge* to built up some *common view* related to a given *problem* D_P associated with some *part of the real world* and with a *time model* (which period of time, which cycles). Further it is clear which kinds of actors shall be involved in the problem. The group of experts will collect their common view into a text describing an *actual state* S as part of the problem as well as a text with *change rules* X which describe which kinds of changes are assumed to be *possible* and which are acknowledged to be necessary. These two texts S and X should be given in a format which allows the participating experts – or probably other persons which have not been part of the discussion – to apply the change rules X to the state description S in a way that it can be decided what is a follow-up state S' of the actual state S. The follow-up state S' then will become the new actual state S and the change rules can be applied again to S. This process of applying the change rules X to the actual state description S will be repeated until there is some stop criterion which is fulfilled.

Such a process of a repeated application of the change rules to the state description is in this text called a *simulation*. To apply the change rules X to a state description S requires that it can be *decided* whether the *conditions* of the change rules are *fulfilled/ satisfied* $\models$ by the state description S, written as

$$S \models condition\ \Phi \tag{5}$$

For this decision whether a satisfaction is possible or not it needs in the general case in varying degrees *knowledge about the meaning function* μ associated with the expressions. The only bearer of real meaning functions on the planet earth are human actors. Thus in the general case it is not possible to delegate this decision of being satisfiable or not to a computer. Therefore two main cases seem to be applicable:

1. **Bookkeeping by a computer:** Because no computer is available to do the general job sufficiently well one can use a computer nevertheless – like in the case of a program to support writing and drawing – to support the editing of the two texts S and X as well as different support functions to make a simulation easier.

2. **Artificial Actors:** To that extend as some *artificial actors* are available which can *mimic* a human actor in some respect it seems conceivable to run a simplified simulation which can show cases which are nevertheless of interest for the human actors.

In the following text both cases will be described: (i) Support Editing; (ii) Offer artificial Actors.

4 Actor Story [AS]

In an *actor story* [AS] all the possible situations will be described in which the user of the system can interact with the system. The user has here the role of an *executing* actor and the technical system has the role of an *assisting* actor. That a user can interact with the system it has to be assumed that there exists a *system interface* SI which allows the user to make some *inputs* to the system as well as to receive some *outputs* from the system. Strictly speaking the user itself has also an interface called the *user interface* UI and which kind of actions the user is capable of or which kinds of inputs the user is capable to receive depends from this user interface.[3]

4.1 Start

In the start state it is assumed that there is at least one user before the system and the system interface SI invites to start the process.

Task: Start the process

Actors: Human experts.

SI: A main window W1 showing all possible options:

(a) Edit Q
(b) Edit S and Edit X in parallel
(c) Simulate
(d) Evaluate
(e) Stop

Actions: Select an option.

[3]Because people can have great differences in their user interfaces as well as in their cognitive capabilities a full analysis of the user-system interaction had to address all these cases. In the context of the actual project we have still a strongly experimental setting and it will be considered only the case of no special handicaps.

4.2 Editing Q

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.

(a) Describe the *problem* P

(b) Describe the intended real part of the world (*space*).

(c) Describe the *time* model T: which time period, which cycles.

(d) Which *actors* A are participating in the scenario.

(e) Some other assumptions.

Actions: Select every question and write an answer.

4.3 Editing S and X State in Parallel

This state allows the editing of the texts S and X in parallel, but one must not. Additionally one can call from within this state the simulation mode to test whether the actual texts are working.

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

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

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 or test the simulation or stop.

4.4 Simulation

General: The simulation mode depends highly from the kinds of actors which are involved. In a *human only* simulation all decisions will be made by human actors. In an *artificial actor only* simulation the whole simulation can be done completely automatically. In a *mixed simulation* real humans as well as artificial actors can interact.

Task: The Program which manages the simulation is called a *simulator* σ. According to the situation it has to manage the application of the rules. The simulator σ computes 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 actors as well as – optionally – artificial actors

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 a new state S' by applying the change rules X. This follow-up state S' then becomes the new actual state S. The process can be repeated. 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.

4.5 Evaluation

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.

4.6 Stop

Task: End the process.

Actors: Human experts.

SI: Bye Bye window

Actions: Quit.