# KOMEGA REQUIREMENTS No.1, Version 3
# Basic Application Scenario
*

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

August 6-11, 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 due to intensive discussions in the INM project team.

## 1 Basic Application Scenario

Before starting any kind of programming one has to consider, which *application scenario* is the context of the software and what are the detailed *functional and non-functional requirements* which have to be fulfilled to match the intended case. Figure 1 gives a first starting point for the intended application scenario.

**Main Actors:** The main actors in the intended application scenario are some *experts* working as a *group* with the *common intention* to solve a *given problem* $P$ (a task, a question, ...).

**How to Proceed:** At the *beginning* of the process every *expert* $A_i$ (with 'i' as an index in the range of the number 'n' of experts) has its own *experience*, therein embedded the individual *knowledge*. These experiences are usually mostly *unconscious*. Thus it is needed to start a *process of common communication* to *activate* as much experience as possible from the unconsciousness

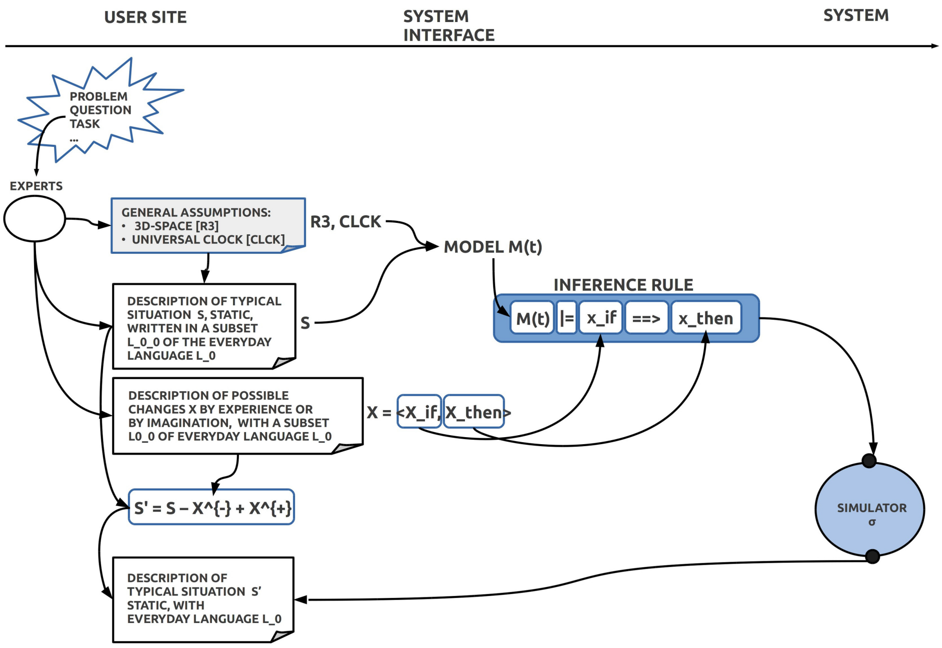


Figure 1: General framework for experts to share their experience in the format of a computer aided simulation-game

which is *related* to the problem $P$ in question. Clearly, this 'hosted' knowledge can be extended with *actual knowledge* from external sources.

**Target:** This activation process of available knowledge *reaches* its first *end* if the experts could *write down* two *texts* in a subset $L_{0.0}$ of the used *everyday language* $L_0$:

1. A description of at least one *static state* $S$ which is a typical part of the intended *problem* $P$.
2. A collection of *change-rules* $X$ representing a set of *known* possible changes related to this static state as well as – in some cases – a collection of rules representing a set of *new possible* changes enabling new static states related to the problem $P$.

These two texts are close to the understanding of the experts measured in the light of the used everyday language $L_{0.0}$.

**Implicit Knowledge: Meaning** In this text it is assumed[1] that the *meaning* of language expressions is located *inside* an actor as his individual *meaning function* $\pi$ which maps internal neural structures $M_{nn}$ – which are representing in some sense experience from the outside world and the body – to other neural structures $E_{nn}$ – which are representing structures of a language – in a way that the M-structures are functioning as the *meaning* for the E-structures, and E-structures are possible pointers to M-structures. The M- and E-structures are mostly unconscious and are very dynamic in nature; they can continuously change. An *actor* $\alpha$ is in this perspective a kind of a *translator* from properties of the *real world* $RW$ into some *internal states*$IS$ of his body and then from the internal states back to the *outside world*.

$$\alpha_{perc} \quad : \quad RW \longmapsto IS \tag{1}$$
$$M_{nn} \cup E_{nn.L_0} \quad \subseteq \quad IS \tag{2}$$
$$\pi_{\alpha} \quad : \quad E_{nn.L_0} \longleftrightarrow M_{nn} \tag{3}$$
$$\alpha_{com} \quad : \quad IS \longmapsto E_{L_0} \tag{4}$$
$$\sigma_{\alpha} \quad : \quad RW \longmapsto L_{0.0} \tag{5}$$

This *actor based translation* $\sigma$ from properties of the real world into the expressions of a subset of the natural everyday language $L_{0.0}$ is not a 1-to-1 mapping but highly modulated by the actor involved in this translation. The actors are the only *carrier of the knowledge* encoded in the written text. Thus the actors can decide what these texts mean, and they are able to decide, whether these expressions in the text do really *refer* to something in the state $S$ assumed as a *static* state. If expressions *do refer* then these expressions are assumed to be *true* in this context, otherwise it is unclear, what the case is. If an expression refers to the scenario and that, what the expression *means* in the light of the knowledge of the actor, is somehow the *complete opposite* to that what is *given*, then such an expression can be classified as *false* in this context.

It is further assumed with the sciences that the internal knowledge of the actors is *biased* by the fact that all experience is organized in a way which resembles a *3D-like space* as well as an implicit timely ordering. All actors are able to organize parts of their internal knowledge as if these can be *ordered in a time-line* and they can perceive events as happening before, simultaneously or after some other events.

**Change Rules X:** Change rules X have the basic structure $X_{if} \Rightarrow X_{then}$. The condition $X_{if}$ represents expressions of the used language $L_{0.0}$ which have to be

[1]Based on knowledge from the empirical sciences especially biology, psychology and brain sciences.

*true* in the assumed static state S, and if this is the case, then the action $X_{then}$ has to be executed. The action is very straight forward: there are either some expressions $E^{-}$ which are part of S which have to be *deleted* in the follow-up state S', and/ or there are some expressions $E^{+}$ which have to be *newly introduced* in the follow-up state S', written as:

$$X = X_{if} + X_{then} \quad (6)$$

$$X_{then} = E^{-}, E^{+} \quad (7)$$

Thus to generate a follow-up state S' from a given state S one has to proceed according to the following schema:

$$S' = S - E^{-} + E^{+} \quad (8)$$

Because there can be more than one change rule $X_i$ which can be activated one has to consider more than one action part:

$$S' = S - \bigcup E_i^{-} + \bigcup E_i^{+} \quad (9)$$

**Semantic Dependency:** While the processing of the then-part of a change rule $X_i$ is *meaning free* the test of the condition part of a change rule has nevertheless some semantic 'flavor'. The structure here resembles the case of *model theory*,[2] written as:

$$M_t(x) \quad iff \quad x = \langle R3, T, S \rangle \quad (10)$$

$$x \models X_{if} \quad (11)$$

'x' is here a model which can *satisfy* the $X_{if}$ part of a change rules. Different to classical model theory the model in this case is not a mathematical structure but the state description S enhanced with the assumptions about $R3, T$ , encoded by the meaning functions $\pi_{\alpha}$ of the writing actors. Thus while every participating actor $\alpha$ is assumed to be able to decide by himself whether the model $x$ satisfies the if-parts of the change rules $X$, it is not yet completely clarified how one can make the *implicit* knowledge in a way *explicit*, that a purely *mechanical* decision would be possible. Because we want to use a computer to do these mechanical decisions automatically we will need this clarification.

[2] See e.g. Chang & Keisler (1978) [CK78], or more fundamentally Tarski (1936) [Tar36]. See my discussion of Tarski comparing syntactical and semantic logical concequences as well as a meaning device [DH20]

**Simulation Inference Rule:** Putting everything together we have the following *inference rule* for our *simulation* allowing to *infer new true situations* having already a true situation and having change rules:

$$\begin{aligned} Given \quad &: \quad M_t(x) && (12) \\ x \quad &\models \quad X_{if} \Rightarrow S' = S - \bigcup E_i^- + \bigcup E_i^+ && (13) \end{aligned}$$

**Simulator $\sigma$:** The intended simulator $\sigma$ must be able to realize the inference rule purely automatically. Thus we assume:

$$\sigma : \quad S \times X \times INF \longmapsto S' \tag{14}$$

From *formal logic* we know[3] that formal logic alone cannot transcend the realm of expressions because formal logic is bound the the space of expressions only.

From a *human actor* we know that he can bee characterized as a kind of a *meaning device* $\alpha$[4], which explains that he can *look beyond expressions* and he can use its implicit knowledge of the meaning of expressions and through these meanings he can activate models of the real world representing important properties of the real world.

If a simulator $\sigma$ would be a purely logical device then the task would not be solvable. But from history we know that there exists a *mathematical model* of a machine which can do logical proofs. This concept has been invented by Alan Mathew Turing 1936[5] while he was writing a paper where he showed that it is possible to re-write the Goedel-Proof from 1931[6] in a different way by using a mathematical concept which is mimicking the behavior of an accountant writing symbols on a sheet of paper. Today this *symbol-writing-accountant-abstraction* is called a *Turing Machine* and in computer science the Turing-machine concept is the basic concept to define and to compute everything which is related to a computer.[7]

From a *Turing machine [TM]* it is known that a TM can handle arbitrary expressions $E_{tm}$. Furthermore we know that the *neural encoding* of world experience inside human actors can be simulated by a TM using *artificial neural*

[3] See my discussion of the syntactical and semantic logical consequences Doeben-Henisch (2020)[DH20].

[4] Well knowing, that a human actor is far more than only a meaning device!

[5] See Turing 1936/7 [Tur 7]

[6] See Goedel (1931) [Goe31]

[7] Until now it is not clear whether and to which extend the so-called quantum-computers can be understood as computers in the sense of a Turing Machine!

*networks [ANNs]* as $ANN_{tm}$ simulating the behavior of the brain. Thus the expressions used by a TM can not only be the expressions of some natural language $L$ but also expressions representing the neural code of *neural correlates of expressions* $E_{tm.nn}$ as well as *neural correlates of properties* $M_{tm.nn}$ of the world. Therefore a TM is in principle capable of setting up a function $\pi_{tm}$ which can mimic the meaning function $\pi_{\alpha}$ of a human actor as $\pi_{tm} : E_{tm.nn} \longleftrightarrow M_{tm.nn}$. Because of these capabilities in principle a TM can be understood as a possible *artificial meaning device [AMD]*.

**An Empty Solution ...:** Knowing that a computer realizing a TM can *in principle* behave like an artificial meaning device [AMD] *can* become a solution *if* we would be able to write down all the needed functions to do the job. From human actors we know that these gifted with a highly developed body and brain can learn any kind of language, but they nevertheless need many years until they can master a natural everyday language well.

A TM compared to a human actor with body, brain and a *built-in* capability to *learn continuously* is an *empty sheet of paper*. To enable a computer to act like a human actor would require to *translate* all this *structural knowledge* which is encoded in the body, in the brain and in the necessary learning functions in a way that it would fit as functions inside a TM. But, even if this some day would have been realized as a TM-body-brain-learning capable machine this machine had to learn as every human child too.[8]

From this we can infer for our project that we should proceed as follows:

1. Organize a process with *multiple phases*.
2. Every phase has its own *subset* $L_{0.i}$ of a natural language $L_0$ as reference set and it has to be clarified *which functions are needed* to be implemented in a TM functioning as a simulator for these texts.
3. The theory of *formal languages* associated with the *theory of automata* can perhaps be of some help, as well as some parts of *machine learning [ML]*.
4. But the main paradigm to be followed is the paradigm of the *automatic meaning device* $AMD$.

[8]Turing has considered these questions and has written some papers about this. One paper was a report written in 1949Turing (1949)[MM]. A German translation can be found in Dotzler & Kittler (1987)[DK87] .

**Turing's Return:** The early ideas of Turing to enrich the TM concept with those parts of the real world which are necessary for intelligence, learning, and symbolic communications have been – and are – the headlines of a *research program* which can only be realized in a *close connection* with the *human actor* who tries to *understand himself* by looking into the *artificial companion* like into a *mirror* showing *similarities* but also *differences*.

## References


[CK78] C.C. Chang and Jerome H. Keisler. *Model Theory*. Noth-Holland, Amsterdam - Oxford - New York, 2 edition, 1978. 1st editioin 1973.

[DH20] Gerd Doeben-Henisch. Reviewing tarski's semantic and model concept. 85 years later .... *eJournal uffmm.org, ISSN 2567-6458*, pages 1–11, 2020. https://www.uffmm.org/wp-content/uploads/2020/08/review-tarski-semantics-models-v1-printed.pdf.

[DK87] Bernhard Dotzler and Friedrich Kittler, editors. *Alan Turing. Intelligence Service*. Brinkmann & Bose, Berlin, 1987. Collection of selected papers of A.M.Turing in a German Translation.

[Goe31] Kurt Goedel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, i. *Monatshefte fuer Mathematik und Physik*, 38:173–98, 1931.

[MM] Bernard Meltzer and Donald Michie, editors. *Intelligent Machinery*, number 5, Edinburg. Edinburgh University Press. The contribution 'INtelligent Machinery' from A.M.Turing is a report from 1949. The whole content can be find here: https://www.doc.ic.ac.uk/ shm/MI/mi5.html.

[Tar36] A. Tarski. Über den Begriff der logischen Folgerung. *Actes du Congrès de Philosophie Scientifique*, VII, ASI 394:1–11, 1936. The Congress took place 1935.

[Tur 7] Alan M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(2):230–265, 1936-7. http://www.turingarchive.org/browse.php/B/12.