

KOMEGA REQUIREMENTS No.1, Version 2

Basic Application Scenario

*

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

August 6, 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 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

*Copyright 2020 by eJournal uffmm.org, ISSN 2567-6458, Email: info@uffmm.org, Publication date: August 6, 2020

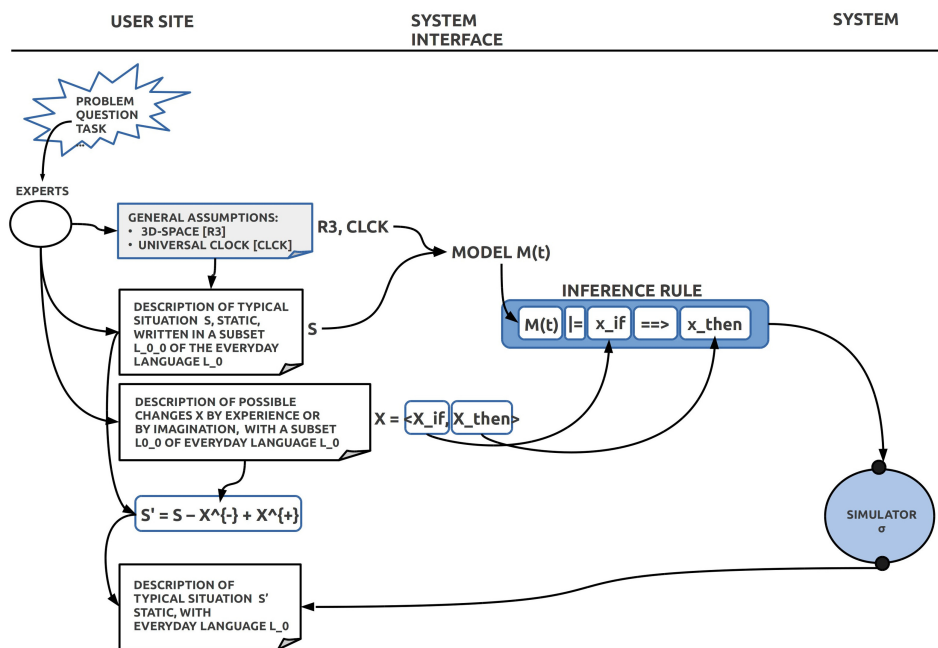


Figure 1: Basic requirements for the komega SW project, version 2

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: In this text it is assumed¹ that the *meaning* of language expressions is located *inside* an actor as his individual *meaning function* π which

¹Based on the new knowledge of empirical sciences especially in biology, psychology and brain sciences.

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 E-structures. The M- and E-structures are mostly unconscious and are very dynamic in nature; they can continuously change. An *actor* α is in this perspective a kind of a *translator* from properties of the *real world* RW into some *inner world* and then from the inner world again to the *outside world*.

$$\alpha_{perc} : RW \mapsto M_{nn} \times E_{nn.L_0} \quad (1)$$

$$\pi_\alpha : E_{nn.L_0} \longleftrightarrow M_{nn} \quad (2)$$

$$\alpha_{com} : M_{nn} \times E_{nn.L_0} \mapsto E_{L_0} \quad (3)$$

These basic facts are important for to understand that the actors are the only *carrier of the knowledge* encoded in the written text. Thus they can decided what these texts mean, and they are able to decide, whether these expressions in the texts do really *refer* so something in the intended scenario, in the state 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 is the case. 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 $X \Rightarrow Y$. The condition X represents expressions of the used language $L_{0,0}$ which have to be *true* in the assumed static state S , and if this is the case, then the action Y 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:

$$X = X_{if} + X_E \quad (4)$$

$$X_E = E^-, E^+ \quad (5)$$

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

$$S' = S - E^- + E^+ \quad (6)$$

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

$$S' = S - \bigcup E_i^- + \bigcup E_i^+ \quad (7)$$

Semantic Dependency: While the processing of the action part of a change rule X_i is *meaning free* does the test of the condition part of a change rule have some semantic flavor. The structure here resembles the case of *model theory*,² written as

$$M_t(x) \text{ iff } x = \langle R3, T, S \rangle \quad (8)$$

$$x \models X_{if} \quad (9)$$

' 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 π_j of the writing actors. Thus while every participating actor 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.

Inference Rule: Putting all pieces together we have the following *inference rule* allowing to *infer new true situations* having already a true situation and having change rules:

$$\text{Given} : M_t(x), S, X \quad (10)$$

$$x \models X_{if} \Rightarrow S' = S - \bigcup E_i^- + \bigcup E_i^+ \quad (11)$$

Simulator σ : The intended simulator σ must be able to realize the inference rule purely automatically. Thus we assume:

$$\sigma : S \times X \times INF \mapsto S' \quad (12)$$

²See e.g. Chang & Keisler (1978) [CK78]

References

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