CASE STUDY
SIMULATION GAMES - PHASE 1
Iterative Development of a Dynamic World Model

Part of the
Generative Cultural Anthropology [GCA] Theory
*

Gerd Doeben-Henisch
doeben@fb2.fra-uas.de
Frankfurt University of Applied Sciences
Nibelungenplatz 1
D-60318 Frankfurt am Main

June 18, 2020

Abstract

To work within the *Generative Cultural Anthropology [GCA]* Theory one needs a practical tool which allows the construction of dynamic world models, the storage of these models, their usage within a simulation game environment together with an evaluation tool. Basic requirements for such a tool will be described here with the example called a *Hybrid Simulation Game Environment [HSGE]*. To prepare a simulation game one needs an iterative development process which is described below.

1 Overview of the Whole Development Process

Figure 1 gives You an overview about the whole process from the first perceptions of some experts until their simulations of a constructed dynamic model and eventually some modifications to the model afterwards.

GUIDING IDEA: The guiding idea for the whole process is given in a situation, where some experts are *triggered* by a question or a problem or a task or something like this, which stimulates the experts to *start a process* in which they *activate* their experience, to *share* their experience, and to try to *built an appropriate dynamic model*. A dynamic model is a combination of a static model

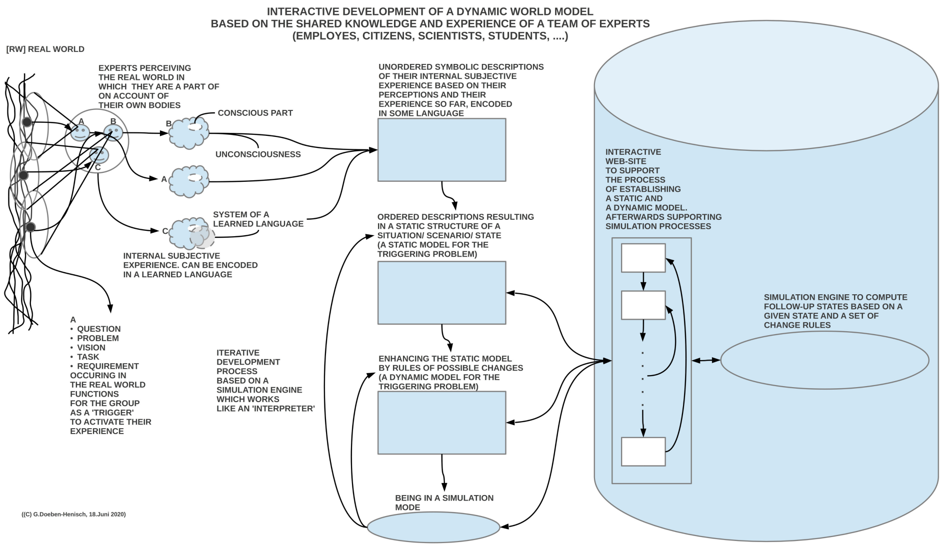

Figure 1: Overview for the Iterative Development Process to Construct a Dynamic Model

and some rules of change, which describe, how the static model can change. If the dynamic model can be compared with the real world in a way that the model appears to be *in agreement* with the real world then the dynamic model is a *sound* model, which can be said to be a *true* model. Otherwise the dynamic model is either *undefined* or even *false*, if there exists some direct *contradiction* to the real world.

COGNITION: Following science we know that a *homo sapiens actor* has a *brain* hidden inside in his body. This brain is responsible for the management of all experience, for all knowledge, for all emotions. This brain has no direct connection with the real world *as it is* but with the real world as it is *sensed* by different sensors on the surface of the body and hidden inside the body. These different sensed data are computed by the brain – mainly unconscious for the actor – and representing that what is called *cognition*. The so-called *consciousness* is part of the cognition and is continuously interacting with the *unconscious* part. The *symbolic language structure*, which has developed to a *meaningful* language by *learning* can be associated with parts of the cognition. With a developed and commonly learned language the experts can try to share parts of their conscious as well as unconscious cognition.

In the context of *simulation-game development* usually the following basic steps are assumed to be helpful:

1. **Brainstorming:** The experts trigger their experience to get some answers, which can be shared with each other in an unstructured way. With the method of brainstorming the content of the *unconscious* part of the cognition can become *activated* to reveal *what is there*.
2. **Structuring Objects in Space:** Assuming some ordering criteria the experts will try to apply these to the unordered data so far and to arrange them according to these ordering criteria.
3. **Adding Possible Changes:** After having gained some static structure they continue with looking for possible and sound rules of change for this statically ordered state of facts.
4. **Testing by Simulation:** To compute a follow-up state of a given state by applying some defined rules of change to this given state is called a step in a simulation. Repeating such a computation generates possibly a whole sequence of follow-up states leading to a final state of the sequence which represents a possible state in the future. The whole sequence represents a simulation of length n. Repeating simulations with varying assumptions can show a subset of possible future states.
5. **Evaluation:** A set of simulations can be used to evaluate whether these computed states appear to be in a defined sense *sound* states or not. Many more questions can arise from such simulations and can trigger new insights into the complex behavior of the dynamic model.
6. **Iterate:** If the evaluation points to critical points regarding the assumptions of the modeling then one can modify these assumptions and thereby modify the model.

SIMULATOR: As one can see in figure 1 the *simulator* is assumed to be an entity located in the server, which can be reached by defined interactions through an *interactive web-page*. The entity is assumed as a *software program (an algorithm)*. The experts have to represent their ideas about static and dynamic structures in *documents* written in some defined *symbolic language* for to enable the simulator to *simulate*. Thus, after (i) recognizing the written input as fulfilling some formal criteria for a *well written document* describing a dynamic model, the simulator (ii) has to compute the *follow-up state*. The follow-up state has then to be presented as an output to the web-page, which can be read by the experts.

2 Cognitive Aspects of Symbolic Expressions

Before we dig into the details of the construction of the static and dynamic models it is helpful to consider for a moment the different kinds of languages

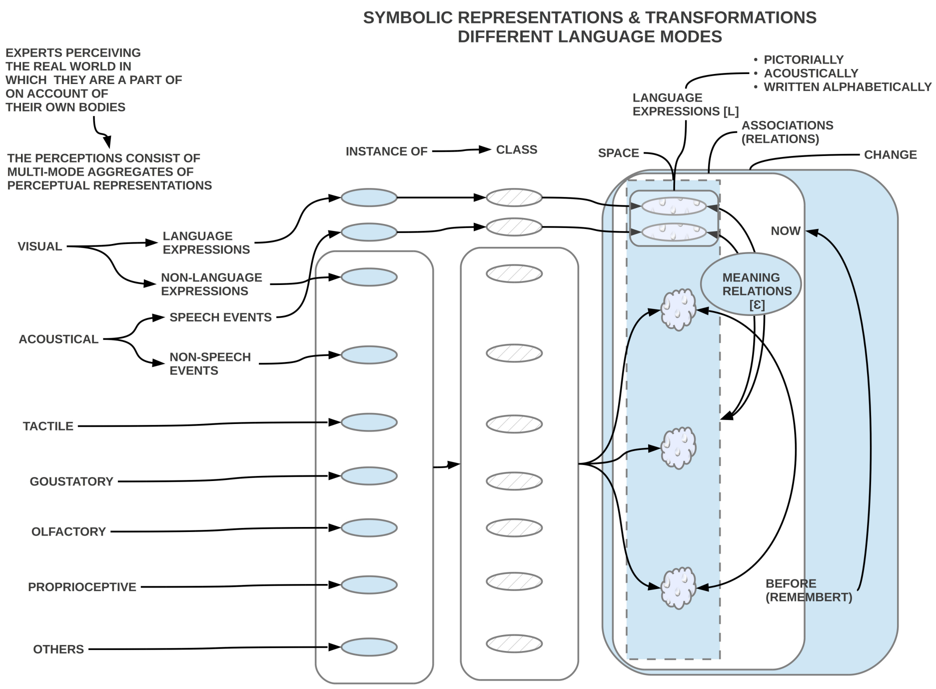

Figure 2: Some Cognitivre Aspects of Perceived Experience and their Relations zu Symbolic Expressions

and their cognitive context.

Figure 2 points out that the perception of an expert is a multi-threaded process with different modes of perceptions which are organized into one compound perception including visual, acoustic, tactile, and many more different perceptions which can be constitutive for a certain perception of an object or an event.

And these compound perceptions can still be differentiated even more by distinguishing between *concrete instances* which can be mapped into some more *general clusters* representing objects, which are mapped into some *three-dimensional space*, with numerous possible associations (relations) between these objects, and furthermore there can be *changes* with regard to an *assembly of objects*.

The set of all these different kinds of perceptions, relations, and changes constitute the *cognitive space [C]* of an expert. To that extend that an expert has learned a *language [L] common with other experts* the expressions of this language can be *mapped* ϵ into subsets of his cognitive space, even onto the language expressions themselves. Those parts of the cognitive space which are mapped to language expressions are usually called the *meaning* of the language

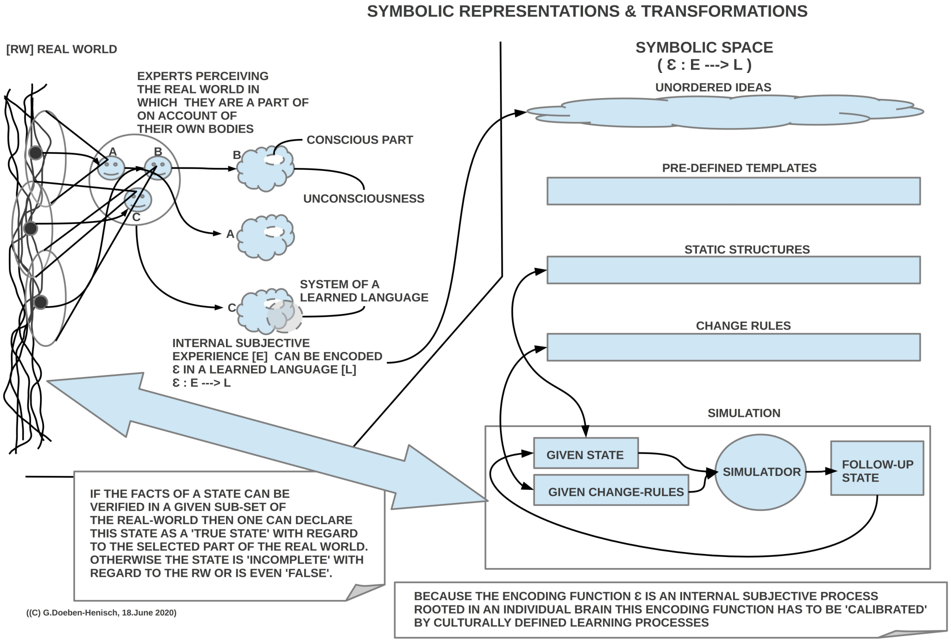

Figure 3: Symbolic Representation of Experience and its Transformations

expressions. If we write the mapping as $\epsilon : L \longmapsto C \cup L$, then the meaning $m \in L \cup C$ of some expression $l \in L$ would be $\epsilon(l) = m$.

3 Symbolic Representations and Transformations

Figure 3 repeats the idea of the different phases leading from a trigger-event to the brainstorming, to the ordering as well as to the possible changes of the assumed ordered static structure.

But this figure does not yet decide with which kinds of languages the documents should be realized. In a paper together with my colleague Matthias Wagner (2007)[DHW07] I had pointed out within the realm of Safety-Critical Systems [SCS] that in the realm of requirements analysis it could be of great help to use at least the following three modes in parallel: everyday language, pictorial language (not only diagrams, rather like Comics), as well as a mathematical formalization.

In the years following 2007 I have experimented with many versions of this idea.[1]. It became clear that the following combination seems to be helpful:

[1] See some of these in the DAAI-document from 2018: `https://www.uffmm.org/`

1. Start with an *everyday language* description in parallel to a comics-like *pictorial story*.

2. Generate a *formal description* based on the to other documents.

The logic behind this approach is rooted in two facts. First: words of a language expression as such do not point directly to their intended meaning. You have to *know explicitly* the meaning relation ϵ of the language in use, otherwise you do know nothing about the intended meaning. If you know the meaning relation, words are fine and can be quite accurate. Second: Because the visual structures in the perceptions of human actors with regard to triggering situations seem to be highly similar, it is possible to use a pictorial language whose elements are common to all human actors. Famous examples for this are comics which have gained an international attention. One example is the comic called *Asterix*[2], which has been translated into more than 100 different spoken languages without changing the pictures.

In the follow up papers of this case study I will demonstrate with so-called toy examples how such different languages will look like, and then I will give a complete formalization of this approach followed by a real example which we will test with real people in the real world.

References

[DHW07] G. Doeben-Henisch and M. Wagner. Validation within safety critical systems engineering from a computational semiotics point of view. *Proceedings of the IEEE Africon2007 Conference*, pages Pages: 1 – 7, 2007.

wp-content/uploads/2019/05/aaicourse-15-06-07.pdf

[2]See: https://en.wikipedia.org/wiki/Asterix