# 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 18-30, 2020, Version 3

**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.

## Contents

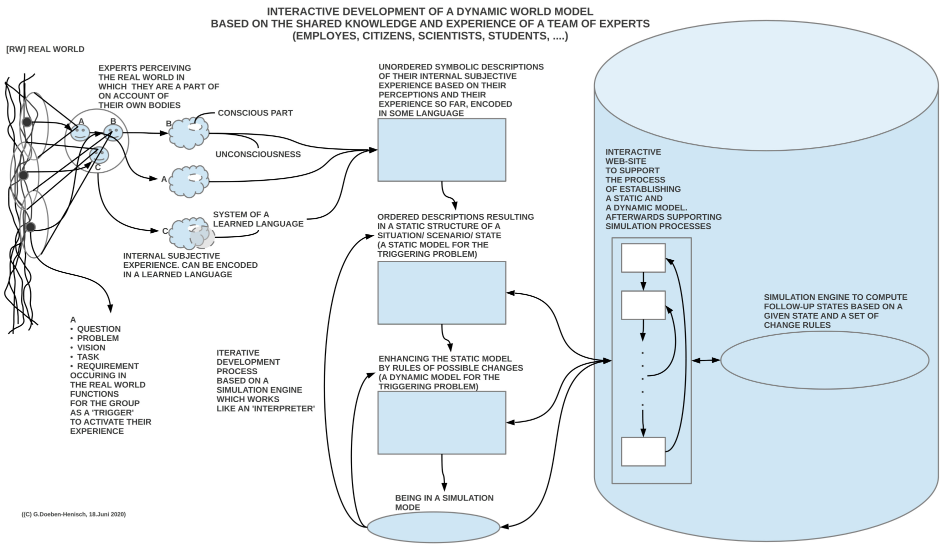


Figure 1: Overview for the Iterative Development Process to Construct a Dynamic Model

# 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 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*

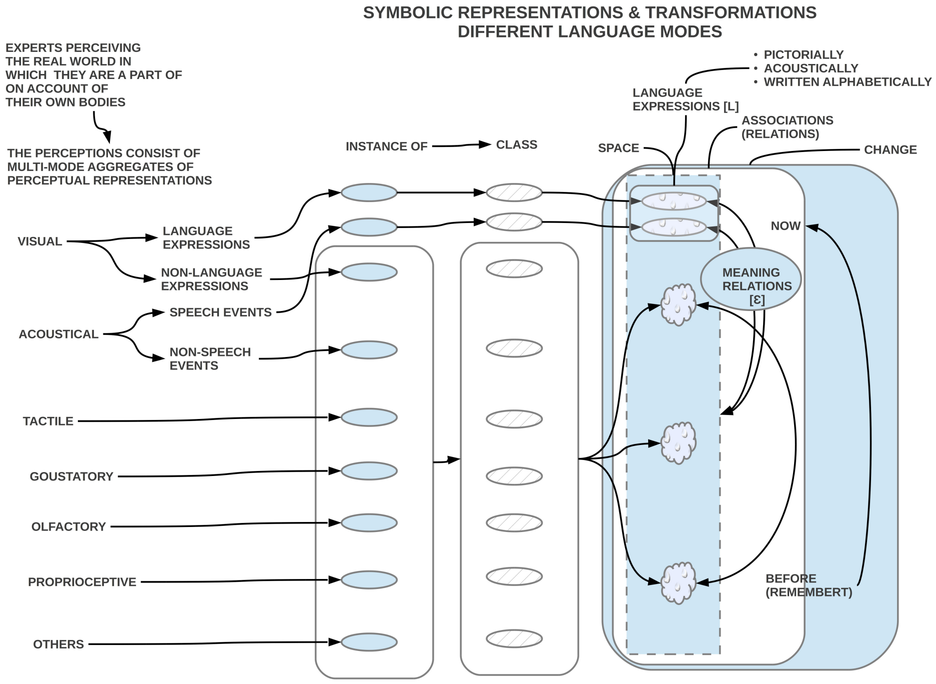


Figure 2: Some Cognitivre Aspects of Perceived Experience and their Relations zu Symbolic Expressions

*(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 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.

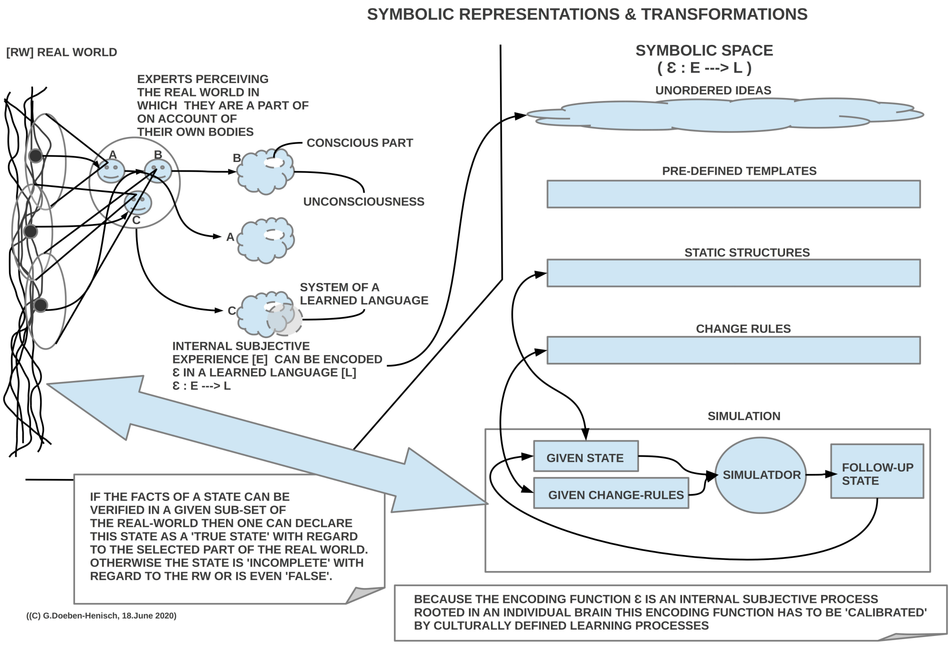


Figure 3: Symbolic Representation of Experience and its Transformations

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* $\epsilon$ 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 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. 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 $\epsilon$ 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.

## 4 Abstract-Concrete Concepts

During the first phase of the development of a simulation game dynamic model it is complete open in the beginning which kinds of concepts will be activated by the participants of the group. But it is possible to forecast that there are only two cases of concepts possible: Either (i) the concepts can be related to a

[1]See some of these in the DAAI-document from 2018: `https://www.uffmm.org/wp-content/uploads/2019/05/aaicourse-15-06-07.pdf`

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

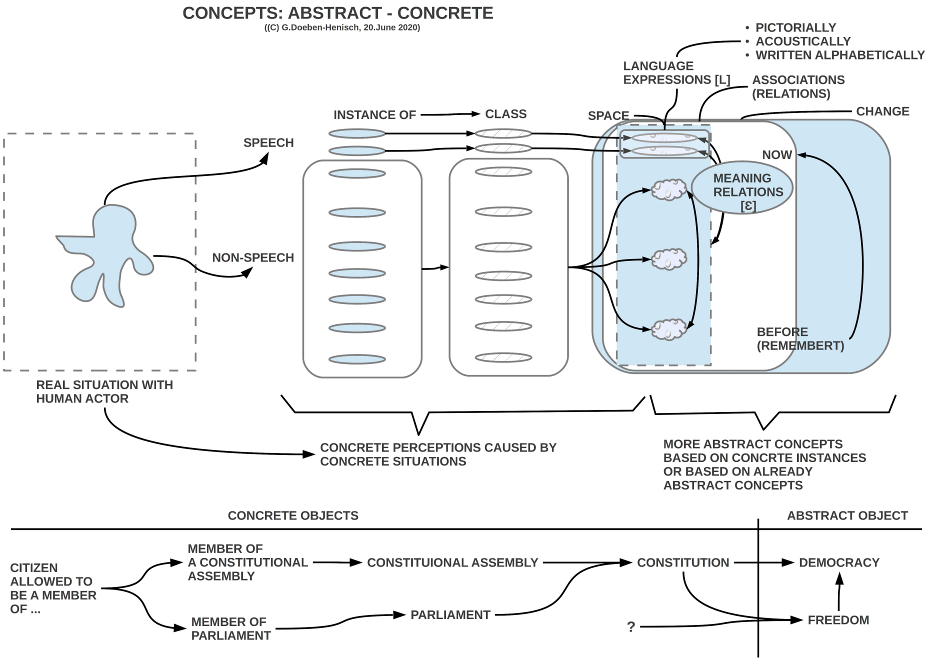


Figure 4: Explaining Abstract-Concrete Concepts within the Cognitive Space. How to Deal with Them

*concrete Object* or *event* related to a *concrete situation* where the participants are part of or have been part of in the past or (ii) the concepts cannot be related so some concrete object or event. In the last case the meaning of the concept is somehow *fuzzy* or *undefined*. If it is possible to name more concepts which are by their *abstract meaning* hierarchically related to each other in a way which finally leads to a concrete concept with a concrete meaning, then one has a path of relations enabling some *grounding* of the abstract concepts into a concrete concept. This kind of *derived conceptual grounding* does not automatically enable a more concrete meaning to the abstract concepts, but one has a least an *anchorage in the real world* from which some more concrete meaning eventually can arise.

Figure 4 shows in its lower part a simple example with the concepts *democracy* and *freedom*. While the abstract concept 'freedom' is only linked to the other abstract concept 'democracy', the abstract concept 'democracy' has a link to the concrete concept *constitution*. The 'constitution' is in a nation usually a concrete document with a text describing the basic ideas of a democracy (if the nation is intended to be a democracy). This concrete document 'constitution' is usually related to two different concrete objects: either the *constitutional assembly* which compiled the text for the constitution and voted for it or – after the constitutional act – that constitutional organ – usually the elected *parliament*

– which can change the text of the constitution if necessary. Both mentioned organs are collections of concrete *members* which are elected and which do the work. The elected members are taken from the *citizens of the state* which fulfill the requirements for being allowed to become an elected member. Thus while the concept of democracy as such is not a single concrete object but represents a complex meaning corresponding to a complex set of objects, actions, and relations between them this abstract concept can be related to more concrete concepts which are corresponding directly to concrete objects of the everyday world.

## 5 Implicit Structures Embedded in Experience

The section about 'concrete - abstract' concepts reveals, that a brainstorming looking only to associated words is highly *underspecified* compared to the everyday real world experience. As already discussed in the paper Doeben-Henisch (2020)[DH20] our everyday experience is the product of processes which are embedded in the real world with many given structures.

As pointed out in the figures 2 and 4 of this paper the cognitive machinery of a human person as an actor absorbs reality and thereby the structures which are embedded in this reality. From this follows that every object or change or relation can be reconstructed as part of these implicit structures.

Applied to the process of developing a *dynamic model* based on the shared experiences of the participating experts (cf. figure 5) one can establish the working hypothesis that the *triggering* of free associations within a group of experts does activate individual, concrete parts of the experience hidden in the unconscious memory, but these associations are usually not context-free. Thus if one modifies the triggering event in a way that one enriches the triggering event with information about intended contexts like space characteristics (a region, a city, a house, a company...), about time windows, about typical situations, etc. then the associated elements will be selected according to these general contexts. Thus the activated elements by the experts will automatically be framed by a possible situation-like state or eventually by some finite set of situation-like states. Thus naming individual, single facts appears like a process of revealing step wise aspects of a situation containing all these facts.

Such a process and its formal layout explains how a brainstorming can show a situation with its embedded structures.

Such a set of facts representing a possible situation can be written either in some everyday language or with a simplified everyday language like

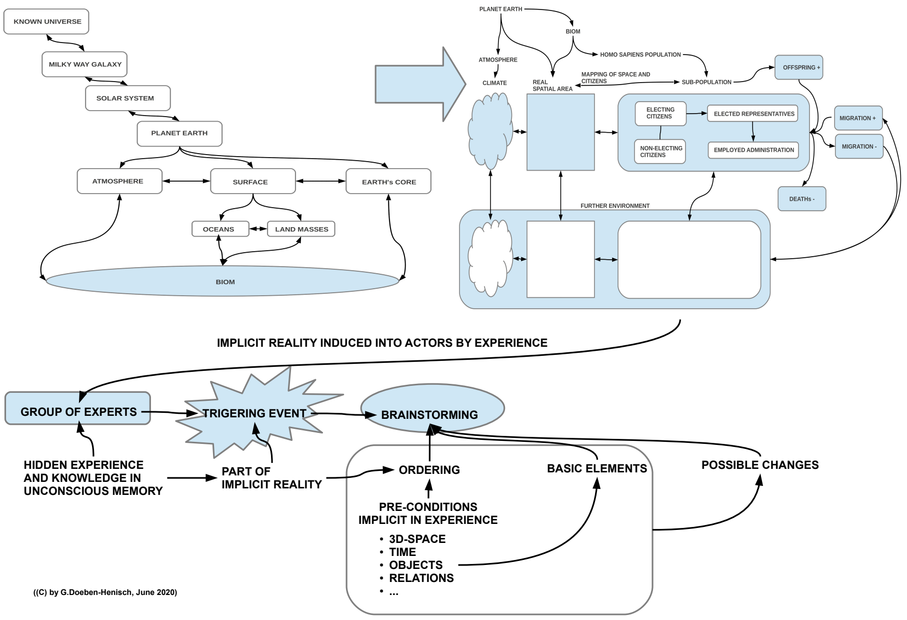


Figure 5: Structures of the real world are implicitly encoded in the individual experience of the world

the predicate-logic without quantifiers. One can do this in two steps or writing the facts directly with a predicate-logic like language.

## 5.1 Example 1

Let us take an example. Figure 6 shows a sheet of paper with some drawings, a wristwatch, and a pen; this on a table. From our real-world experience we know that a table is *usually* part of a room, a room is part of some flat, a flat is ... the set of possible embeddings can become very long.

Thus if we have *perceived* such a situation in some moment of time in the *past* then we can possibly remember this situation depending from the way how our brain has processed this experience.

A simple protocol of such a remembering process could look like this:

1. There is a sheet of paper.
2. There is the drawing in the left upper corner of the sheet of paper.
3. There is the drawing in the right upper corner of the sheet of paper.

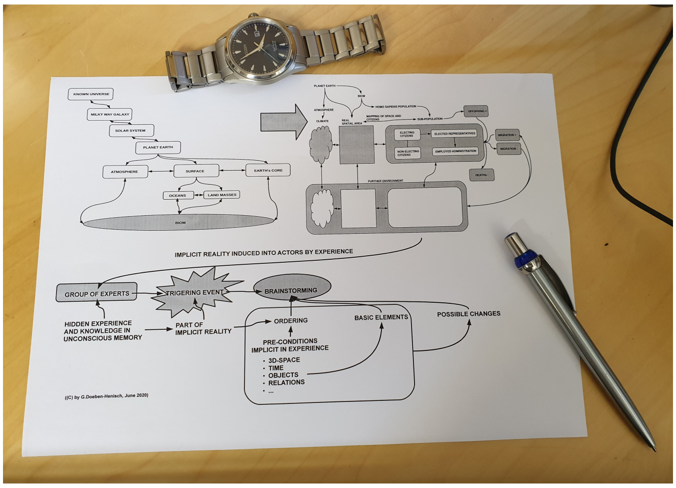


Figure 6: Picture of a real situation with a sheet of paper, a pencil, and a wristwatch

4. There is the drawing in the lower part of the sheet of paper.
5. There is a wristwatch on the upper border of the sheet of paper.
6. There is a pen lying partially on the right border of the sheet of paper.
7. Wristwatch, sheet of paper as well as the pen are on a table.

Transforming this in a more condensed way one could write:

1. sheet-of-paper(p).
2. drawing(d1), located-left-upper-corner(d1,p).
3. drawing(d2), located-right-upper-corner(d2,p)
4. drawing(d3), located-lower-part(d3,p).
5. wristwatch(w), located-upper-border(w,p), located-on(w,t).
6. pen(pn) , located-right-border(pn,p), located-on(pn,t),
7. located-on({w, p, pn},t), table(t).

This more condense kind of writing is using the basic grammar of so-called quantifier-free *predicate logic*[3] using simple enhancements. Thus one can see the following elements:

[3]For a first description of predicate logic, also called first-order logic, see `https://en.wikipedia.org/wiki/First-order_logic`

1. **Object Names:** There are *names* assumed to names of objects given in the memory, like 'd', 'p' etc.

2. **Properties:** Named objects are assumed having possible *properties* like 'sheet-of-paper()', 'drawing()'.

3. **Relations:** Named objects can be embedded in *relations* like 'located-right-border(x.y)', 'located-on(x,y)'.

4. **Multiple Objects:** Multiple named objects can be written as finite sets like 'x,y,z'. In this case such a finite set is handled like one object. This allows writings like 'located-on(w, p, pn,t)' saying, that the three named objects 'w, p, pn' are all on the table.

5. **Fact:** every expression like 'property(named object)', 'relation(list of named objects)' is called a *fact*.

6. **State:** Every finite set of facts is called a *state* or a *situation*.

From this it fallows that the we can write the above lists of facts as a *state S*:

*S* = {*sheet-of-paper(p), drawing(d1), located-left-upper-corner(d1,p), drawing(d2), located-right-upper-corner(d2,p), drawing(d3), located-lower-part(d3,p), wristwatch(w), located-upper-border(w,p), located-on(w,t), pen(pn) , located-right-border(pn,p), located-on(pn,t), located-on(w, p, pn,t), table(t)*}

Such a state has a structure like S=$\langle ON, REL \rangle$ consisting of *object names* [ON] and *relations* [REL].

## References


[DH20] Gerd Doeben-Henisch. Case study 1. part 1: Bootstrapping some citizens. A case study for the daai paradigm. *Integrating Engineering and the Human Factor, uffmm.org*, April 2020. URL: https://www.uffmm.org/wp-content/uploads/2020/04/case1-population-start-v1-1.pdf.

[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.