Gerd Doeben-Henisch, member PES

The Planet Earth Simulator Project – First Considerations to the Visual Programming Interface of the Formal Systems Language

***Abstract*—This are some basic considerations to the idea of a visual programming Interface for the FORMAL SYSTEMS LANGUAGE (FSL) of the PLANET EARTH SIMULATOR (PES) Project. It considers the problem from the point of view of Epistemoogy, Theory of Science, and Computational Semiotics.**

***Index Terms*—Computational Science, Computational Semiotics, Epistemology, Theory of Science**

INTRODUCTION

The PLANET EARTH SIMULATOR (PES)-Project [1] is an open source project which has been started January 2003 by Gerd Doeben-Henisch and Jens Heise at the Institute for New Media [2] in collaboration with the University of Applied Sciences [3] in Frankfurt am Main (Germany). The intention of this project is to offer everybody who has a webbrowser at his disposal to generate knowledge or to use knowledge which others have already delivered to the system.

To get a rough idea of the project we will have a short look to it from the point of usage.

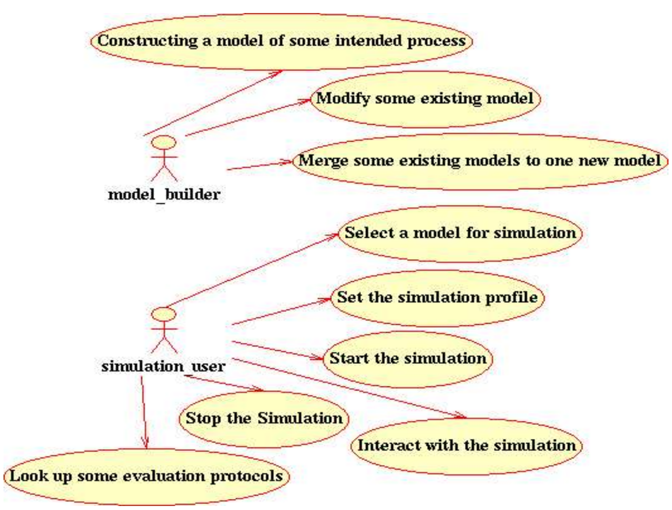

Fig. 1: The Use Case of the Planet Earth Simulator. The user can act either as a model builder or as a simulation user.

The PES-Project is supported by the Institute for New Media e.V. in Frankfurt am Main (Germany)

From the point of USAGE (cf. Fig. 1) you can construct arbitrary intended models of parts of the world. For this you have a VISUAL LANGUAGE *VisualFSL* at your disposal to built VISUAL ARTIFACTS --diagrams, graphs-- as representations of your intended parts of the world. A complete description is called an LMODEL as short for *Logical* Model (for the definition of an LMODEL see [6]). And, as far as other pre-built LModels are already available, you can SELECT LMODELS for simulation and then you can RUN THE LMODELS for to simulate some intended processes. During such a simulation you can also INTERACT with the simulated processes, individually or in groups. SYNCHRONOUSLY or ASYNCHRONOUSLY you can also activate some EVALUATION PROTOCOL for to collect and to show selected data of the process activities.

VISUAL EDITING OF LMODELS

To allow the simulation of processes one has to feed the system with models of processes. In version 1.0 of the PES this should be done by a visual programming interface. This requires the definition of a visual dialect of the FSL. We will call this VISUAL LANGUAGE *VisualFSL* (cf. Fig.2).

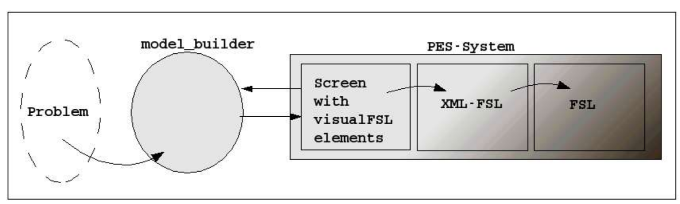

Fig.2: A user as model builder tries to translate a real world problem into a visual representation with the visual Formal Systems Language FSL. This will in turn be converted into FSL

The representation in *visualFSL* will automatically be converted into an internal XML-Format of FSL. This format then can be further converted into a 'pure' FSL-text.

ADEQUACY AND COMPUTABILITY OF LANGUAGES?

At this stage one can ask the general question, which language is *most adequate* for the description of reality? This leads to the further question, how one can use *reality* as a clearly defined reference point for the evaluation of language expressions.

A lesson learned from modern physics is, that there is no such *plain unquestionable reality* as undisputed fixpoint for everybody at any place and time. 'Reality' is bound to demonstrable examples, to measurements and to related language expressions, which can become rather complex. This shows to some degree a kind of vicious circle in that sense that the reality intended from some person can only be communicated by some *language based communication.*

The following diagram (cf. Fig.3) shows the heirarchy of languages leading from natural languages to the visual Language visualFSL.

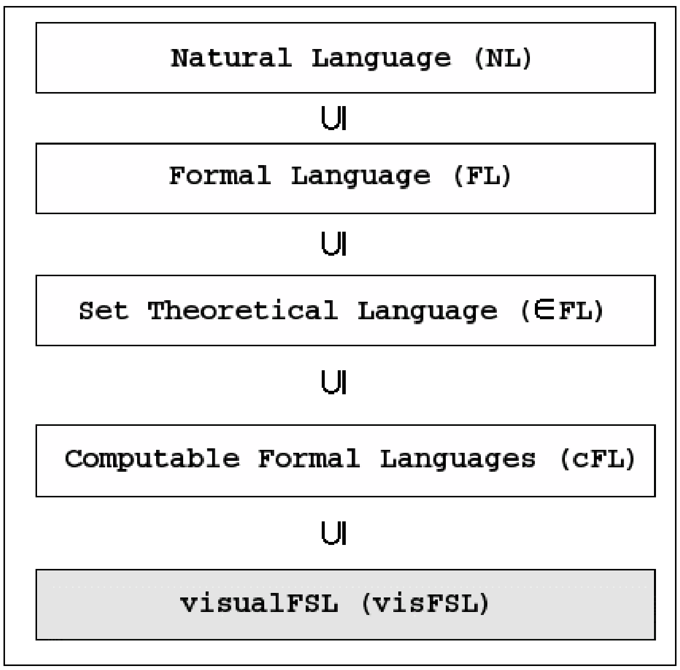

Fig. 3: Language Hierarchy leading from Natural Language to visualFSL.

A natural language is classified as the *strongest possible format* of a language, because here is everything allowed. The problem with this is, that this language is in many situations very vague and uncertain.

This his lead to the project of *formal languages* which are completely defined on a formal basis; every sign used within a formal language can only be used by explicitely defined rules of operations. One strong example of such a formal language is te *set theoretical language*.

Although formal languages seem to imply some limits for the usage it has been shown, that even these languages are too *rich* for the question of decidability. As Kurt Gödel could show 1931 are even 'simple' systems built up with a set theoretical language such that they either are *complete* and *not consistent* or they are *consistenst* or *not complete.* Furthermore could Alan Matthew Turing 1936 show, that formal languages equivalent in power to the set theoretical languages are *not computable*. This means that the usage of those languages for tasks where the computer should come to a defined endpoint is not possible.

These results generated the concept of *computable languages.* Computable languages are those where a computer as powerful as a turing machine is able to reach a definite result after finitely many steps. Formal tools in the realm of computable functions are e.g. certain types of recursive functions, formal grammars and automata.

The *visualFSL* belongs to this class of computable formalisms. Thus this language can be used on a computer to solve finite tasks.

SEMANTIC ADEQUACY OF VISUALFSL?

Having positioned the visualFSL as a computable language one can and should askt the question, to which extend the visualFSL is able to describe processes of the real world which are intended by a user.

As mentioned in the paragraph before is it probably impossible to give an exhaustive answer to the question whether visualFSL is *completely* adequate or not. The best we can do at the moment is to give a *relative* answer comparing the power of visualFSL with the power of other languages.

With regard to *Mathematics* combined with modern *Physics* we now that the mathematical language is extremely powerful; at least is is the formal (set theoretical) language of modern mathematics the most powerful tool we know today.

The key-concepts of mathematics are *set*, *relation* and *function,* whereby *function* is a special case of *relation*. This together allows the formulation of *structures* and *systems*.

Thus if we use the concept of *system* although as key concept in *visualFSL* we know that visualFSL in principle should have the same expressive power as Mathematics.

The *difference* between a plain set theoretical language and visualFSL is computability: unrestricted set theoretical descriptions are not computable. But the intruiging philosophical question is whether this *more* in expressive power of the unrestricted set theoretical languae is *real* or only *virtual* in the sense that one can form expressions without limitations, but the human brain as the basic processor to use these expressions is a finite machine, probably not more powerful than a turing machine, and this brain in turn will therefore probably not be able to exploit the un-compuable parts ot he language really.

As long as one adheres to the hypothesis of the *Turing-Equivalency-of-the-Human-Brain* the usage of visualFSL is not a substantial reduction compared to the ordinary mathematical language.

A WORLD OF SYSTEMS

If the key concept of the visualFSL is the concept *system* then, clearly, every description of the world will describe the world as a network of systems where every individual system can again be a collection of systems (cf. Fig.4).

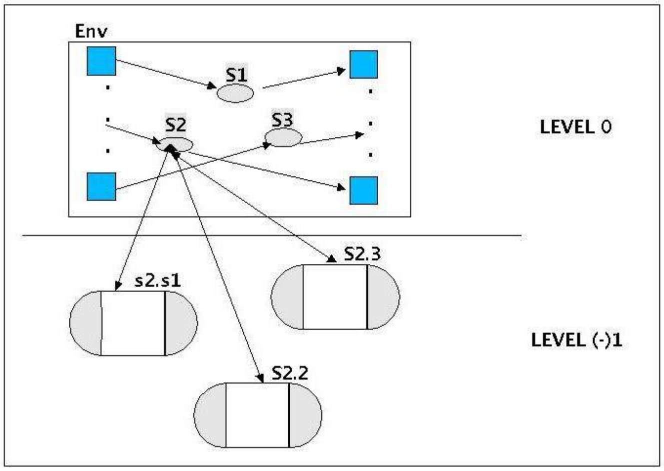

Fig. 4: Starting with a root-system one can introduce more systems which belong to a level below that system which is the 'container' of those additional systems

References

[1] Planet Earth Simulator (PES) Project: http://www.planetearthsimulator.org (This is the project website)
[2] Institute for New Media e.V.: http://www.inm.de (This is the web site of the sponsoring institute)
[3] University of Applied Sciences - Department of Computer Science and Engineering, http://www.fh-frankfurt.de/2_studium/introseiten/index_2fb2.html (The main site of the faculty)
[4 George J.Klir , *Facets of Systems Science*, New York - London: Plenum Press, 1991
[5] Helmut HEROLD, "lex und yacc. Lexikalische und syntaktische Analyse", Addison-Wesley Publishing Company, Bonn - München - Paris et al , 1992
[6] Gerd Döben-Henisch, "The Planet Earth Simulator Project – Description of Formal Systems Language (FSL) Vers.1.0", MS, Institue for new Media e.V., Frankfurt am Main, 2004
[7] Kurt Gödel
[8] Alan Matthew Turing
[9] Examples of computable formalisms