FUNCTIONAL GRAPH SYSTEM MODELLING

(Andreas Scheidegger, Konolfingen, Switzerland)

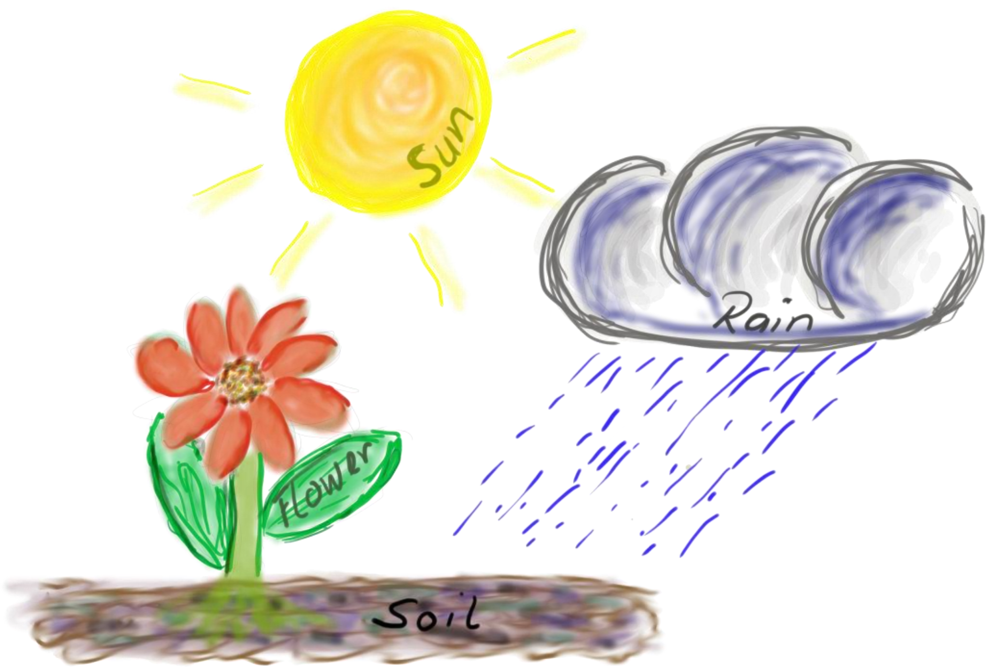
1. INTRODUCTION

Whenever we look at the world, we are applying models, mostly to understand and quite often to extrapolate. A tree is a tree for me, because what I think it is, my model of it, fits with what I see in front of me. Science is mostly about finding the model which fits our observation or extrapolating a model and check for matching observations. The problems we are facing today have always several connected causes. Interconnection-based thinking, living and feeling is therefore fundamental to safe our planet.

There are many general approaches for system modelling. Some methods are listed in (Marchese, 2013).

In the nineties I was facing the problem of modelling interacting machine systems (machine tools and robots) to better understand and control their safety aspects. So, I was looking for a modelling method which can handle physical properties as well as fuzzy aspects, linguistic descriptions, colors or even feelings. I found a method (Vester, 1983), which was used in environmental system modelling. I improved this method to more general use cases and published it under (Scheidegger, 1996). By applying this Functional Graph (I use FG to simplify) method to generate a model of a simple environmental system, I will explain its usage, properties and advantages.

2. FG MODEL

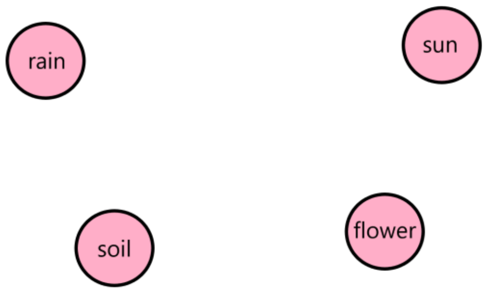


The system we are looking at consists of the objects[1] “soil”, “flower”, “rain” and “sun”. Since we cannot model the whole universe, we need to define system borders. In our case we define these borders implicitly, only considering the objects drawn on the sketch to be part of the model.

In the next step, we try to define the properties of the objects and try to describe the states or conditions they can be in. As we keep the model simple, we use every day’s language for this description. The following table shows a possible definition:

Soil	Can be *wet* or *dry*
Flower	Can *bloom* if it gets sunshine and water or *dies* otherwise
Sun	Can *shine* or *disappear* (behind the clouds)
Rain	It *rains* (out of clouds) or *clears away*

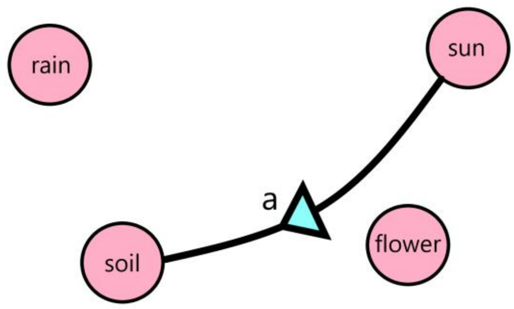
At this state the representation as a graph is introduced. Graphs were used the first time by Leonhard Euler ((Schubert, 2012) and (Wikipedia, https://en.wikipedia.org/wiki/Seven_Bridges_of_Königsberg, 2018)) when he tried to solve the problem of passing all bridges of Königsberg (Kaliningrad) but only once. For our system the graph representation of the objects looks like this:



I must admit, this looks quite boring, without any outstanding information. It’s just the representation of the objects observed in our system. It starts to get more interesting if we introduce the connections between the objects. In the FG the connections represent interactions or influences from one object to another. The easiest way to find these connections is by pairwise comparison and asking for instance “what influence has the sun shining to the soil”, - the soil dries out, “what influence has the soil to the sun”, - none. This influence is drawn as an arrow in the graph, adding a freely chosen identifier (a). Connections in the FG model are always unidirectional.

---

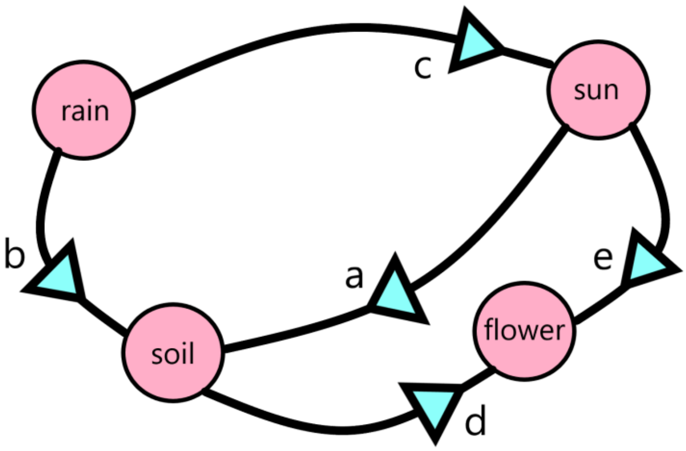
[1] In the original publication I used the word “parameter” to describe an entity of a system. I prefer to use the word “object” instead in this document.



Doing this for all the objects, defines all possible connections in the model. The connections can be drawn as arrows in the graph. Let's therefore list all the influences in our model and describe their function:

a	When the sun is shining, the soil dries out
b	When it's raining, the soil gets wet
c	When it's raining, the sun doesn't shine
d	When the soil is wet, the flower can bloom (only together with some sunshine)
e	When the sun is shining, the flower can bloom (only on a wet soil)

Introducing these influences in the graph, we get the following picture:



This is called a functional graph, which already offers a huge playground for discussion: Obviously, an object with arrows only pointing out to other objects is not influenced and can therefore only be changed from outside the system border. It's either a constant or an *input parameter,* like the "rain" in our model. An object with only incoming arrows has no influence on the rest of the system and its state is probably of some interest to the observer and therefore an *output parameter*, like the "flower", which we would like to see blooming.

We can start to play with an initial state for each object and let it rain, - for a while – and then let the sun shine by letting the rain disappear, - for a while – and so on. We can observe that without a delay of the states of the objects, the flower will never bloom and indirect influences over several objects are mixing up.

In a simulation, the time dependency is solved by adding inertia or time delays to the states of the objects according to the observation in the real system. In our model we probably would need to add a delay time to the soil, so that it's only drying out slowly when the sun is shining. The other problem mentioned earlier is indirect influences. We can learn more about them by introducing an additional representation called matrix of influence which corresponds to an adjacency matrix in the known graph theory.

The rows and columns of the matrix are both labelled with the identifiers of the objects. The labels of the influences are filled into the fields of the matrix following the rule that the row object influences the column object. We are only looking at unidirectional influences, therefore an arrow in the graph corresponds to one field in the matrix.

	Soil	**Flower**	**Sun**	**Rain**
Soil		d		
Flower				
Sun	a	e		
Rain	b		c	

The matrix of influence is a different representation of the same model and the same characteristics of the model can be seen: An empty row, like the "flower" row, means that this object has no influence on the rest of the model. An empty column, like the "rain" column, means that this object is not influenced by the rest of the model and is defined from outside.

This representation shows the advantage of the FG method. It properly separates the observable objects from the pure influences. All observable manifestations and their time dependency are bound in the objects. All connections and dependencies of these objects are bound in the timeless matrix of influence, which I call the "nexus".

As mentioned before, the matrix of influence allows to analyze the indirect influences. By applying the rules of matrix multiplication to itself, the resulting matrix of influence shows the dependencies over two objects, by applying the same rule again, influences over three objects are visible, and so on. Let's call the direct influences, influences of first degree, influences over two objects of second degree, and so on.

	Soil	Flower	Sun	Rain	Soil	Flower	Sun	Rain	Soil	Flower	Sun	Rain
						d				d		
					a	e			a	e		
					b		c		b		c	
Soil		d										
Flower												
Sun	a	e				ad						
Rain	b		c		ca	bd|ce				cad		
					second degree				third degree			

Now, let it rain and see the effect to the blooming of the flower:

There's no direct influence from the rain to the flower. The box where the rain row meets the flower column is empty. Let's move to the second-degree influences and we have the "bd" influence, which means the rain wets the soil and the wet soil makes the flower bloom but, unfortunately, "ce" is against us in the same box, because the rain cloud covers the sun and the lack of sunshine prevents the blooming of the flower. At least the third-degree influence "cad" where the rain cloud covers the sun which keeps the soil wet which lets the flower blooming could help us – but not without sunshine.

As in real life, there's no blooming without sunshine. Therefore, we need to stop the rain, let the sun shine via the second-degree influence "ce" and keep the soil wet by applying the delayed third-degree influence "cad". By the way, a closed loop influence (or feedback loop) would appear as a diagonal element in the matrix of influence.

It's visible now, that this sketch keeps quite a lot of secrets. But without squeezing out the model, where the FG method is very helpful, we would never have achieved this level of information, - actually, information which is a property of the system and the model and FG only helped to show up.

The model we have been analyzing was a rather simple one. Talking to a biologist, he would never agree to reduce the flower to its blooming only. Additional objects like the air need to be added and objects like the flower need to be divided in smaller units to achieve a more precise and detailed model. Here again the FG method can be helpful:

		Soil	Flower				Sun	Rain
		Soil	root	stem	leave	bloom	Sun	Rain
Soil	Soil		d					
Flower	root							
	stem							
	leave							
	bloom							
Sun	Sun	a			e			
Rain	Rain	b					c	

The flower itself is divided in a more precise four object model containing its root, stem, leaves and bloom. With pairwise comparison, the inner influences (green) as well as the incoming influences (blue) and the outgoing influences (red) can be found. I leave this more detailed modelling to the biologist.

Just as a thought experiment, this refinement could be repeated on and on to a model of subatomic level. In general, the objects consist of matter and energy and sensation, images, feelings and thoughts (SIFT and Qualia) [2] and the matrix of influence becomes a field of infinite interconnections, the nexus field.

3. FG PROPRTIES

Even if the discussed example is very simple, it already shows a lot of the properties and possibilities of the FG method. The simplicity of the method allows an unlimited range of applications with various levels of precision without limitation on numbers or physical units, including linguistic, descriptive and even sentimental models. The advantage of the method is the representation of a model as objects made of all forms of qualia and connections from the nexus field.

Without going too much into detail objects and connections must follow some general rules, which I like to describe:

Objects properties

As mentioned before, objects have a variable characteristic or quality. The state of this characteristic is influenced from outside the object. The object itself defines the rule on how it manages cumulated or conflicting influences and on how a delayed or

[2] These objects of generalized quality can be defined as qualia. More detailed information is found in (Wikipedia, https://en.wikipedia.org/wiki/Qualia, 2018) or (chopra & kafatos, 2017), also including the definition of SIFT (sensation, images, feelings, thoughts)

continuous change of state in time is applied. It's also in full consistency with relativity, so that time delays and inertia are only applied locally to the states of the observed object. Objects are not limited to observable items only, but can be sensations, images, feelings or thoughts (SIFT) as well.

Connection properties

As shown before, the connections can be found by pairwise comparison of the objects. The only rule applied is rather mathematical: a connection must transform the state of the characteristic of its source object to a state of the characteristic of its sink object. In the FG model, influencing connections are always unidirectional. The limitation of the transformation to the characteristics of source and sink objects determines the independence from space and time. In addition, the nexus is not observable or measurable. Only its manifestation in qualia can be observed.

Application

The above described rules and properties are not so far from object-oriented programming or UML (Unified Modeling Language). Therefore, an implementation to a state-of-the-art application can easily be done (and was already realized for a specific use case in (Scheidegger, 1996)). Similar applications using the method described in (Vester, 1983), are used for the simulation of environmental models. The much bigger challenge for humans will be to learn more interconnection-based thinking in daily life and to get familiar with the nexus realm. As we saw in our example, already a small number of connected objects is straining our mind.

4. CONCLUSION

Compared to the standard models in physics, where the focus is on the behavior of matter in space and time, the FG method supports models of all kinds of objects – called qualia. These objects are embedded in a field of interconnections – called nexus. Therefore, the FG model is mixing physical units, sensations, images, feelings, thoughts or other imaginable items and let them interact without limitations. The future is not about computer-based models only. Our mind can improve interconnection-based thinking in a way, that we feel the joy of the flower blooming in the garden or the pain of the plastic poisoned sea turtle when we waste a wrapping.

5. REFERENCES

chopra, d., & kafatos, m. (2017). *you are the universe.* New York: Harmony Books.

Marchese, F. T. (2013). *http://csis.pace.edu/~marchese/CS389/L5/Chap5_summary.pdf.*

Scheidegger, A. (1996). *Modellierung von Maschinensystemen mit Hilfe funktionaler Graphen.* Zürich: ETH Zürich.

Schubert, M. (2012). Mathematik für Informatiker. © Vieweg+Teubner Verlag | Springer Fachmedien Wiesbaden GmbH 2012.

Vester, F. (1983). *Ballungsgebiete in der Krise : vom Verstehen und Planen menschlicher Lebensräume.* München: dtv-Sachbuch ; 10080.

Wikipedia. (2018). *https://en.wikipedia.org/wiki/Qualia*. Von Qualia: https://en.wikipedia.org/wiki/Qualia abgerufen

Wikipedia. (2018). *https://en.wikipedia.org/wiki/Seven_Bridges_of_Königsberg*. Von Seven Bridges of Königsberg. abgerufen