

PharmAsia and the Evolution of the Code.

There are dynamic and static variables. For Disease and Physician's cover of the Disease we have dynamic variables. For Work and Injury (that is a cover of Work) the variables are also dynamic. These dynamic variables have further covers that make the Video, or correct the Video.

Class declaration introduce further dynamic variables. Most other variables (as symptoms) are static. They rely on definitions into classes. The reading assignement for static variables are suppressed or correct Video. For the dynamic variables, we do not have suppression (only when tha video has been improved).

The dynamic variables have no type before use. They are initialized at the time following declaring static variables. They are linked.

Prolog uses the Unification: (defined as variables also representing functions, that are allowed in an expression, the process beeing called higher-order unification.). The Japanese produced Actor Prolog that is good for prototyping, but management of exceptions of Python or Ruby uses many (multi) Objects, and have memory erases.

Use Case of PharmAsia as Images and Video: a Design to relieve Pain and Improve Health. Disease is seen as Body Experience and Body Feedback.

1. Physician possesses Videos of Affliction (the Separation Theorem)
Here k_l is a right superior Class:

$$\text{sense of information } \cup [k_l, x_1, x_2, \dots, x_n] \leftrightarrow [t, x_1, x_2, \dots, x_n]$$

2. The Sick corrects Video to enhance properties of his/her Condition. This is determining Classes k_l .
3. Pain is located. (The Sick relates to the Video)
4. Physician corrects Video, enhances movement and determines Regularity.
5. The Sick corrects Video as a selection of Images relating to Pain.
6. Physician orders Images and corrects Video.
7. Pharmacist moves from Regularity to Intake.
8. Physician corrects Video and determines Body Member as Success.
9. The Sick changes final Video