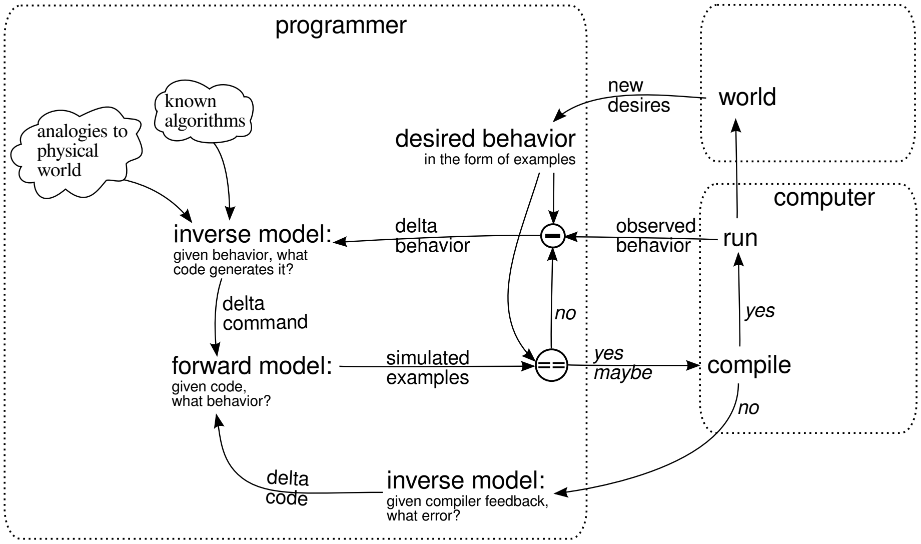
At NCSU this past spring Ras Bodik talked of program synthesis that was guided and trained using examples, something which I found to be totally brilliant, not just because it works, but also because examples are the lingua franca of humans. Examples seem like a good way of communicating desires with computers: rather than requiring a specialized computer language, you could potentially make anyone able to impose new desires & structure upon a computer. Not sure if that was the message, but that's what I saw[1].

Now, from what little I understand from papers and briefly reading Armando Solar-Lezama's thesis, the bulk of the work of this bootstrapping from examples uses a SAT solver. This means re-projecting the programming problem into a boolean satisfiability problem, which engages an active area of research for which there are powerful algorithms. Now, this makes good sense to me, but it is not how I see humans programming. Below is a flowchart of how I think of my (rather organic) development cycle.

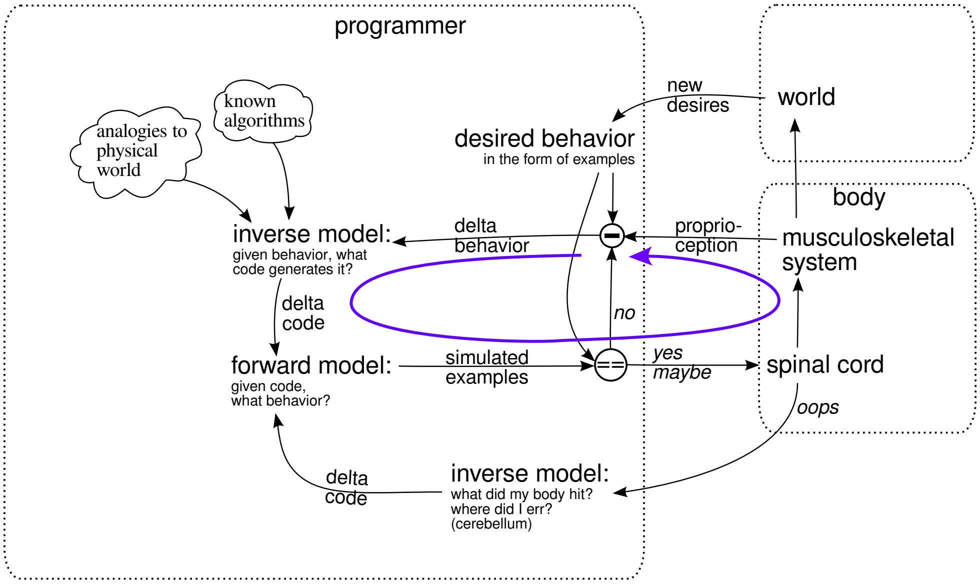


The key step is the 'inverse model' which (roughly) indexes algorithms based on behavior. When put into a feedback loop involving both internal comparison and comparison of desired and computer behavior, you can both debug and create new structure. The world, of course, also plays a role in defining and redefining desired behavior; the programmer is the translator for the world.

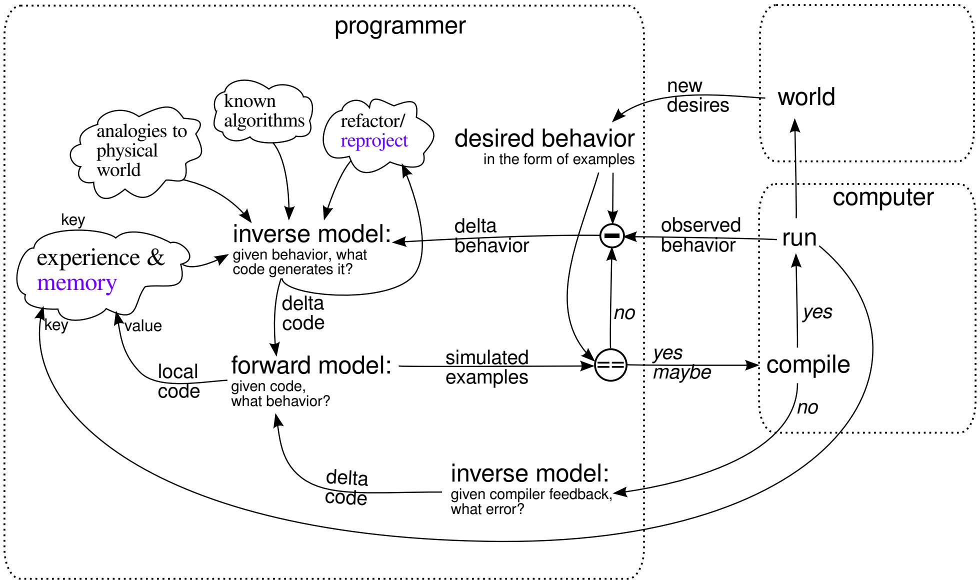
[1] As an aside, I've grown disillusioned with GAs, as they do not solve the communication problem - you still have to describe to the computer what you want, which requires language. Unless you're into artificial life, information (structural and otherwise), does not come from nowhere.

Last update: October 3, 2011.

This idea of an inverse model is not mine, indeed it's not even new – it happens to be a present hypothesis of how we generate *all* motor behavior. The flowchart below has this analogous corticothalamic loop highlighted in purple, with the output and feedback names changed appropriately. It seems to me that programming, like all other volitional behavior, makes use of the general-purpose motor planning and execution hardware that we have [2].



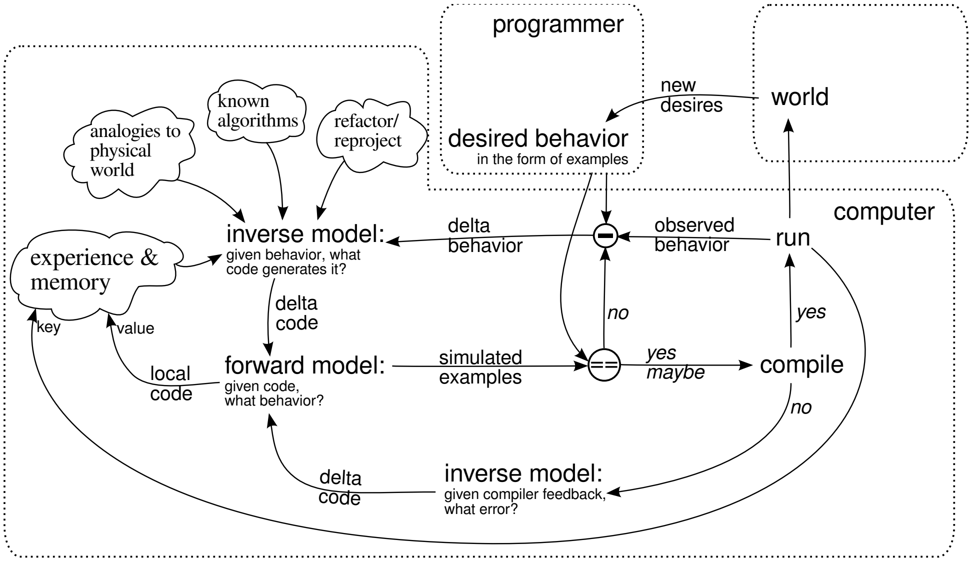
Where we get power over other animals, like the monkeys used in my present lab, is that we are capable of using analogies to re-project problems into new spaces, and more importantly that the forward and inverse models continually bootstrap from experience (see below). It seems to me that the present programming synthesis techniques could benefit from this sort of model building to exploit memory as well as processing power; I understand that some research has been done into 'jungloids', but much more could be done.



[2] Of course, my Phd thesis is on neuroprosthetics and motor control, so I'm biased to see this!

Last update: October 3, 2011.

Irregardless, it should be possible to model most of the motor loops presently in a human programmer's brain on a computer, with the same example-based specification of desired behavior. Indeed, the computer doesn't need a forward model of itself, it can run so much more quickly, and theoretically remember so much more. How exactly such loops should be modeled/implemented is a difficult question, and open to some interesting, challenging research.



Hypothetical future programming schema: let the computer do more work!

Last update: October 3, 2011.