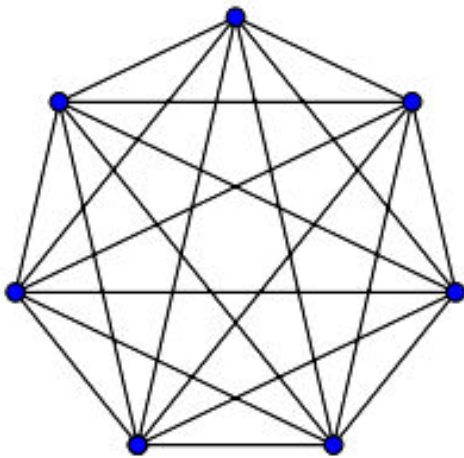
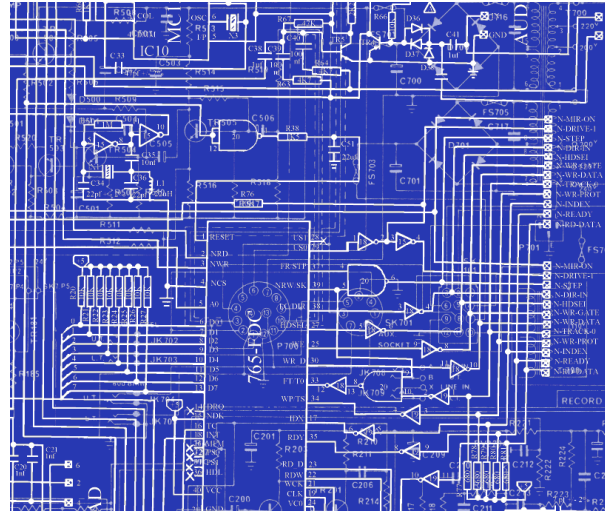


Technological complexity

One problem with the current tinkering system is that it seems too easy to make kitchen sink objects: why not add *all* bioware to a morph, *all* kinds of protection to armor, and so on? Sure, it will increase the price, but there is always someone who can pay the creds or rep. In the case of armor there are some obvious physical limitations, but these hardly apply to morphs.

One obvious approach is to make the Hardware or Programming harder for every extra feature. Designing something with 5 extras would have a -50 modifier, discouraging it.

However, I think a more fun and realistic approach is to allow it: people can just throw standard nanolibraries into their build however they want. The price is complexity, and in particular the risk that something goes wrong. As we know from software keeping everything modular and secure is *very* hard as projects grow (my favorite is the [Vista speech recognition security flaw](#)) If there are N subsystems there are $N(N-1)/2$ potential interactions. This means that the device with 4 extras will have $5*4/2=10$ possible interactions. Assume that there is 1% chance of a problem per interaction, then the total chance of at least one problem is 65%!



So my proposed house rule is that when somebody designs something, it gets assigned a complexity rating N based on the number of things going into it. Then the GM secretly rolls $N(N-1)/2$ dice, counting the number of 0's. An equal number of dice are rolled again, and the number of 0's here tells the number of hidden defects. They are either randomly assigned to pairs of subsystems or to the whole device.

For example, designing an enhanced spaceroach a biologist throws in chameleon skin, extra armor and eelware. That makes 4 systems (roach, skin, armor, electricity) and $4*3/2=6$ interactions. Rolling, he gets one bug. Selecting two systems at random it is between the skin and electricity - when the roach electrifies someone its chameleon skin flickers in a very visible way.

General bugs might be security flaws (whether software, nanotech or biotech): attackers might gain a +30 bonus, reduced durability, vulnerability to certain conditions, random misbehavior (the railgun plays local mesh entertainment when fired) or emergent effects: the spaceroaches start communicating electrically, the AI gets an extra, unintended motivation.

The probability 1% of a bug was mainly set for simplicity. It might be higher for certain kinds of designs (morphs, AIs). In this case, add bugs for 1's, 2's and so on in the second roll.

Careful designers of course test their creations (new skill roll, takes an equal time as the design). That gives them the chance to discover one bug if they are successful (and several on critical success; critical failure means a fix introduces a new bug or misfeature). This can be repeated. Really complex designs are

possible, but requires a lot of time-consuming testing. Sensible designers hire testers to try out their constructions.