# Designer's Notes: GURPS Steam-Tech

**by William H. Stoddard**

***GURPS Steam-Tech*** was put together from the works of many contributors. In checking their submissions before accepting them for the playtest draft, I noticed several points where the contributors disagreed on how to apply the existing rules -- with each other or with me -- and realized that on some points the implications of the rules weren't fully clear to me. Nowhere did this happen more often than in the creation of clockworks, automata, and analytical engines. By the time the final draft went in, the playtesters and I had made many things clearer. For the benefit of other steampunk GMs and players, or of anyone who uses the ***GURPS*** rules for computers, here are fuller guidelines for designing such systems.

## Complexity and IQ

Complexity is an abstract measure of a computer's information processing power, with higher scores indicating more power. For most computers, IQ is Complexity + 3; for sentient computers, IQ is Complexity + 5. The original Complexity scale started at 1, and thus at IQ 4. ***GURPS Steampunk*** added lower Complexity scores, down to -2, corresponding to IQ 1, the lowest score possible in ***GURPS,*** to represent devices simpler than computers, but still possessing some sort of data processing capability and perhaps behavioral response.

## Action Sequences

The simplest thing an automatic device can do is carry out a fixed series of movements or actions. A device carrying out one series, such as a clock, is Complexity -2; a device carrying out several coordinated series in a repeating cycle, such as a music box, is Complexity -1; if the device does not precisely repeat, as in an orrery, where several celestial bodies move with different orbital periods, the device is Complexity 0.

## Self-Regulation

Another class of very simple devices is automatic control systems, which monitor and compensate for environmental changes. Devices such as a thermostat, which keeps a room at a fixed temperature, or a governor, which keeps a steam engine running at constant speed, are Complexity -1. Autopilots, which keep a vehicle moving in a fixed direction at a fixed speed, are Complexity 0. Note that a vehicle that moves on tracks needs a steering mechanism less sophisticated than an actual autopilot; a course control that can start, stop, and make a sequence of right and left turns at specific points is Complexity -1.

## Computation

Complexity scores can be defined for calculating devices. Adding machines are Complexity -2. Four-function calculators are Complexity -1. Calculators with preprogrammed functions, such as scientific calculators, are Complexity 0. Programmable calculators are Complexity 1 or higher.

## Codes and Cyphers

Encoding or decoding messages with simple substitution codes is a function much like simple calculation; machines for this purpose are Complexity -1. Such coded messages can be broken without a penalty to Cryptanalysis. More complex encryption schemes require actual programs, discussed below. Breaking any code, even a simple substitution code, also requires an actual program.

# Programs

Only a machine of Complexity 1 or higher can be programmed. A procedure counts as a program if it contains branch points, where the machine performs one or another action depending on the internally stored results of previous actions. If there are no branch points, or if the branch depends on something external (such as a human operator throwing a switch, or a thermostat sensing the temperature of a room), the procedure is not a program and usually requires Complexity 0 or less.

Programs carry out a variety of functions; major categories are personality simulation programs, advantage programs, skill programs, and utility programs. See ***GURPS Robots*** for several examples of each type.

A standard machine can run two programs at its own Complexity.

# Languages

A program amount to a set of instructions. But the form taken by the instructions depends on the complexity of the machine. Complexity 1 machines must be programmed by actually specifying the internal state of the mechanism, using what is called "machine language." The very first computers during World War II were actually programmed by physically rewiring the circuits, and analogous methods involving Mechanic skill can be required in a steampunk setting.

Machines of higher Complexity can be programmed or instructed through "languages" that bear a closer resemblance to actual human languages. The instructions may take the form of alphanumeric punched cards, or lines of text on a screen, or speech, or sign language, depending on the machine's sensors. At Complexity 2, a machine can understand a restricted instruction set -- something like Fortran for a computer, or a set of code phrases for an early mechanical man. At Complexity 4, it can cope with actual human languages.

# Personality Simulation

A machine at Complexity 4 can achieve limited personality simulation, which includes a single disadvantage and up to five Quirks. Over time, the machine's lack of depth and of actual human experience will become apparent. A machine at Complexity 5 can achieve full personality simulation with a multilayered personality. Either type of program can be either voluntary (a role the machine is playing) or obligatory; obligatory personality programs are worth character points, but voluntary ones are not. A machine at Complexity 6 has the potentiality for sentience; a sentient machine does not simulate, but actually has a personality.

A simpler option is a pet program, which simulates the behavior of an animal. Pet programs are possible at any Complexity; the maximum IQ a program can simulate is its Complexity + 2. Note that the beginnings of real trainability at IQ 4 correspond to the ability to understand a limited range of spoken commands at Complexity 2. A pet program can include psychological disadvantages corresponding to those of the animal it simulates, such as Hidebound or Incurious.

# Skills

A skill program grants a machine a number of points dedicated to a specific skill. Note that points

spent on mental skills count x4 because of the machine's Eidetic Memory. The maximum number of points in a skill program depends on Complexity: 1/2 for Complexity 1, 1 for Complexity 2, 2 for Complexity 3, 4 for Complexity 4, 8 for Complexity 5, and 16 for Complexity 6 (counted as 2, 4, 8, 16, 32, or 64 for mental skills).

How does this translate into skill levels? Machine IQ also depends on Complexity. So for a Mental/Average skill such as Cartography, Complexity 1 would allow IQ 4 and 2 points of skill, for Cartography-4; Complexity 2 would allow IQ 5 and 4 points of skill, for Cartography-6; Complexity 3 would allow Cartography-9, Complexity 4 would allow Cartography-14, Complexity 5 would allow Cartography-23, and Complexity 6 would allow Cartography-40.

But someone has to write the program, and it's not possible to write a program above one's own skill. In a realistic campaign, the best person in a given field is probably skill 20 -- skills above 20 are getting into the "best in history" range and their possessors will not be writing programs! So a machine of Complexity 5 or 6 probably should not have skill higher than 20, even if it could run a more complex program. If there are sentient machines in a campaign, they can write their own programs and get up to skill levels no human being can match.

Machines of low Complexity can't have enough skill to be worth much. But a machine can also provide assistance to a human user's skill. Allow +2 to skill at the Complexity where such assistance first becomes possible, and an added +1 for each added level of Complexity.

# The Automaton Option

An analytical engine can have the Automaton option, which enables it to control a vehicle, automaton, or mechanical man. The code that does this occupies one of its two program slots and enables it to perceive, communicate, manipulate, and move about.

An automaton has DX as well as IQ. This is equal to half its Complexity (rounded down), plus 8. Thus, a Complexity 4 machine has DX 4/2 + 8 = 10. Physical skills, based on DX, are not enhanced by Eidetic Memory; for a Physical/Average skill such as Boxing, Complexity 1 would allow DX 8 and Boxing-6, Complexity 2 would allow Boxing-8, Complexity 3 would allow Boxing-9, Complexity 4 would allow Boxing-11, Complexity 5 would allow Boxing-12, and Complexity 6 would allow Boxing-14.

In a steampunk setting with mechanical men, "programming" may take place through physical drill in a series of actions, and a "programming language" may be a spoken language with a small number of standard signals. The whole process may look much like the military training of the period.

# The Dedicated Option

A machine can have the Dedicated option, meaning that its programs are built into its gears or circuits. This inflexibility makes it cheaper but less flexible. A normal machine has the Rote Learning [-25] disadvantage, enabling it to learn maneuvers but not new skills. A Dedicated machine has Cannot Learn [-30] and cannot even learn maneuvers.

A Dedicated machine can still have more than one program; it simply can't change which ones it is running, or store programs without running them. For example, a Complexity 4 mechanical man with Juggling-12 as a Dedicated program needs one program slot for the skill program, and another for its basic body operation program, so clearly it can run two programs at the same time.

Since machines of complexity 0 or less cannot be programmed, they all have the Dedicated option.

# The High-Capacity Option

The High-Capacity option gives a machine three slots at its Complexity, instead of two. If it has the Automaton option, its body operation program occupies one slot, so it has two left, instead of one.

For machines that do not run programs, High-Capacity can be interpreted in other ways. For a music box, for example, it could mean having several different stored melodies that can be played alternatively; for a calculator, it could mean being able to retain a number in memory, over and above the two it was immediately operating on, like a four-function calculator with a Memory key.

With these interpretative rules, it's possible to build automata for a variety of purposes. For example, consider the control device for the bomb-carrying ornithopter automaton. This needs to direct a series of flapping movements that keep the device flying and also to execute a series of turns and drop bombs at specified points, but it's not able to sense if the wind blows it off course, so it's Complexity -1. At TL(4+1) this is achievable with a Genius small device, a standard miniengine, or a Dumb microframe; since weight is at a premium in a flying machine, make it a Genius small device. In addition, give it the Automaton, Dedicated, and Mechanical options.

Accounting for the effects of these choices gives a device that weighs 15 lbs., occupies 0.3 cf, costs $5,000, and consumes negligible power. With Complexity -1, it has IQ 2 and DX 7; its flight will probably be somewhat awkward.

To make it a true autopilot, capable of correcting for the wind blowing it off course, it would need Complexity 0. This might just barely be reconciled with the need for low weight by making it a Genius miniengine with the Automaton, Dedicated, Mechanical, and Supercompact options. The result would weigh 60 lbs., occupy 1.2 cf, cost $105,000, and consume 0.2 kW. Complexity 0 would give it IQ 3 and DX 8. Raising the cost of the ornithopter from $40,000 to $140,000 might be justified by increased reliability; an experimental advanced design might be a useful target for foreign spies in a clockpunk setting.

With a little ingenuity, a GM can contrive many other clockworks, calculating engines, and automata to aid, baffle, or tempt players. ***GURPS Steam-Tech*** offers a number of examples based on these rules.

Article publication date: December 21, 2001

---



---



---