# Substrate-targeted programming (STP)

Philip Thrift

8 hours ago

*Philip Thrift*

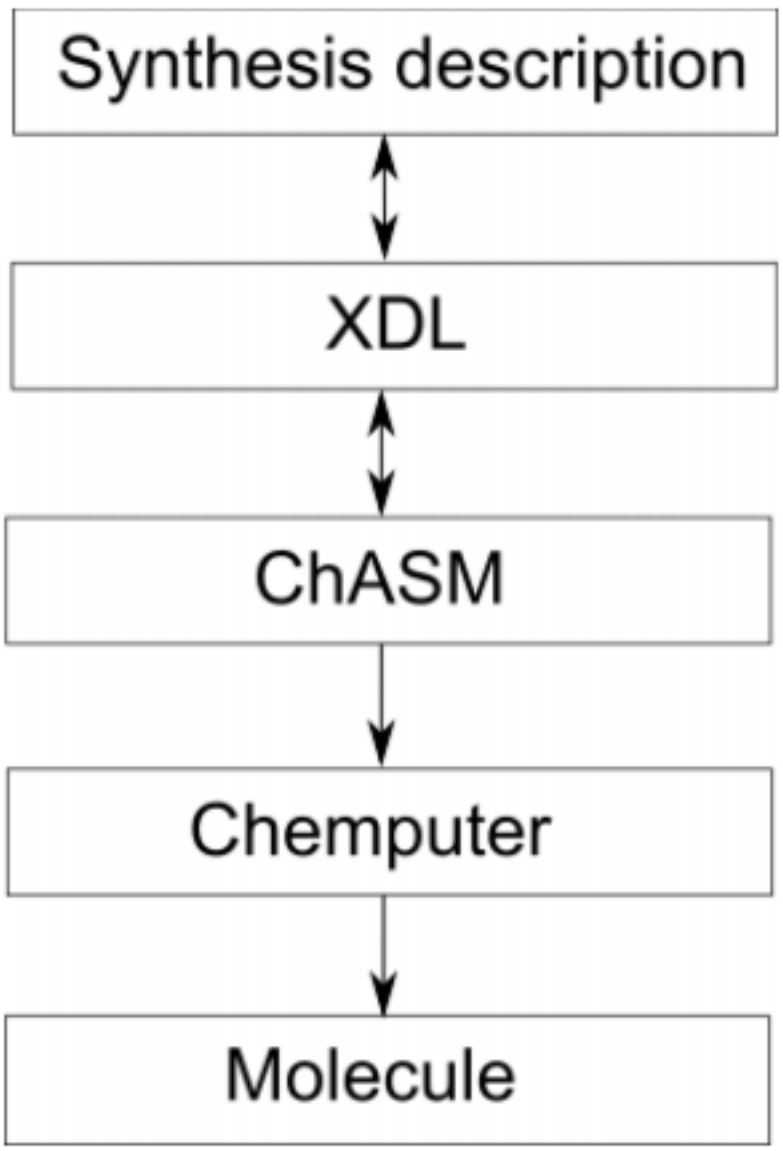
Substrate-targeted programing (STP) is a programming paradigm [2] that consists of a class of programming languages, frameworks, and methodologies where the programs are compiled – or translated – into objects of a specific substrate. STP can be considered as the programming languages for programable matter [1]. The compiled objects – which are not machine code objects of a conventional computer – may have uncomputable features that are outside the standard Turing machine model. The vocabulary of an STP language is built around terms of a particular substrate, or domain. Thus an STP language is a domain-specific language (DSL). Its intended target is produce objects to be executing in "real world" vs. executing as conventional-computer machine code objects.

STP includes programming languages for synthetic biology, chemistry, and various types of materials (physical media). For example, in ChASM (Chemical Assembly Language [3][4], a program can appear as [5]:

```
...
# stirring and waiting
    S SET_STIR_RPM(reactor_reactor, grignard_stirring_rate);
    S SET_TEMP(reactor_reactor, grignard_temperature);
    S START_STIR(reactor_reactor);
    S START_HEAT(reactor_reactor);
    S SET_RECORDING_SPEED(150);
    S WAIT(wait_time_grignard_initiation);
    S STOP_HEAT(reactor_reactor);
    S SET_TEMP(reactor_reactor, room_temperature);
```

. . .

The translation of program to chemical object proceeds as shown:




**The PTLOS framework**

PTLOS(π,λ,τ,o,Σ)

| | |
|---|---|
| π | program |
| λ | language |
| τ | transformer |
| o | object |
| Σ | substrate |

A configuration PTLOS(π,λ,τ,o,Σ) — lower case Greek letters π, λ, τ, o, and capital Greek letter Σ

are variables that take on concrete (particular) values — is defined:

PLTOS(π,λ,τ,o,Σ) designates a program π that is written in a language λ that is transformed via a compiler/assembler τ into an output object o that executes in a computing substrate Σ.

“Material PLTOS Thesis”:

> Every material (*alt*. physical) phenomenon can be effectively represented by some PLTOS(π,λ,τ,o,Σ).

$\tau^{-1}$ is a decompiler/disassembler: it takes an object o and produces a program π,in some language λ.

**Σ = synthetic biological**

τ is a biocompiler / biomolecular assembler (from the developing field of synthetic biology).

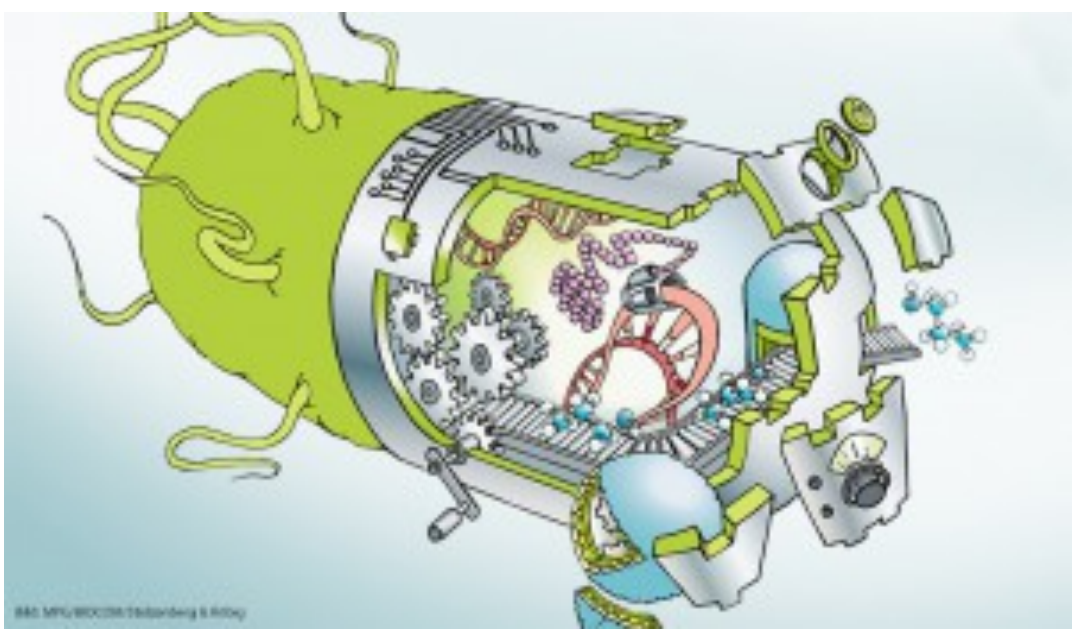
Example: A biochemical molecular program (π) written in a synthetic-biological language (λ) that is biocompiled (τ) into a life form (o) that is injected into a person (Σ) to cure a disease.

If o is effective (n carrying out its programmed task of attacking the disease), this PLTOS is an effective representative of a life form. (In fact the representation is the life form itself.)

Languages for synthetic biology, cf. [10].

<

**Semantics for STP**



from [6]

*The material semantics of a program derives from the properties of the material substrate of its execution.*

Unconventional computing (or computation) may become more "conventional" with progress in synthetic biology and nanotechnology where new kinds of "computers" are being made. In the PLTOS) framework, the *objects* could be lifeforms or microbots (or even macrobots) that go out into the "real world" to perform tasks versus being conventional (object) code running within a conventional computer. [7] points to a *physical semantics* in addition to the conventional programming language semantics (denotational, operational, axiomatic, etc.) Physical semantics would define the semantics of certain types of unconventional programs (like bioprograms of synthetic biology) in terms of their physical, chemical, and biological properties.

In addition to conventional [8] and physical semantics, some might expect (assuming an experiential nature of matter. This (A *panpsychist* theory posits "psychical" states in matter in addition to the physical – like charge, mass, … – ones:)

> **φ-states** (physical [including chemical and biological])
> **ψ-states** (psychical [or experiential])
>
> material semantics =
>
> physical (*incl.* chemical+biological)
> +
> psychical (or experiential) semantics

The exploration of physical and psychical semantics (towards a conscious synthetic agent) is a next phase of unconventional programming.

A toy example for material semantics: A Turing-type computer, but instead of operating with symbols, it is operating with emojis – but the emojis have actual (material) realization as (elements of) experience.

**Future**

1. The development of STP languages and compilers.
2. An exploration of the nature of computability of STP objects.

[1] https://en.wikipedia.org/wiki/Programmable_matter

[2] https://en.wikipedia.org/wiki/Programming_paradigm

[3] https://science.sciencemag.org/content/sci/suppl/2018/11/28/science.aav2211.DC1/aav2211-Steiner-SM.pdf

[4] https://github.com/croningp/ChemputerSoftware

[5] https://github.com/croningp/ChemputerSoftware/blob/0.1.1/experiments/ChASM/nytol.chasm

[6] http://news.bio-based.eu/maxsynbio-max-planck-research-network-in-synthetic-biology/

[7] https://www-users.cs.york.ac.uk/susan/bib/ss/nonstd/index.htm”

[8] https://en.wikipedia.org/wiki/Semantics_(computer_science)

[9] https://www.cs.york.ac.uk/nature/SpInspired/workshops/TEMC-2019-Tokyo/programme.html

[10] https://interestingengineering.com/synthetic-biology-gets-a-new-molecular-programming-language-called-crn