Theory of mechanical information

Sunday, 26 May 2019

Machinery

The first human-computer interface is a sequence of instructions. Assembly language is the system of representation of logical semantics prescribed by hardware. The computation of numerical processes employs an operable specification of process semantics, P_σ.

$$P_\sigma = \Sigma_i$$

The sequential digital computation, Σ_i , is the principle of organization: the specification of mathematical semantics by the enumeration of common, invariant operations (Principia Mathematica).

The operational perspective is definitive of the computational machine. It envelops the mechanical interface to mechanical spacetime. The operator at the intersection of the production and consumption of assembly language objects is a symbol assigned to a mechanical operation defined in hardware. The execution of the operation consumes time and energy to change the state of the machine.

The physical frame of reference includes information, organization, and spacetime:

$$
\begin{aligned}
\pi\,\{\, & o \leftarrow \tau, \\
& \nu \leftarrow \rho, \\
& \kappa \leftarrow \mu \\
\}\,. &
\end{aligned}
$$

The operational role of the machine in information spacetime, π, is observed in the physical reference frame composed from informational spacetime ($\kappa \leftarrow \mu$), informational association ($\nu \leftarrow \rho$), and informational structure ($o \leftarrow \tau$) [TMI/20190428/1].

This machine, Π, is observed in terms of physics and information in order to observe logic and mathematics in mechanical specification and operation, language and interaction. The reference frame has been conceived as the definition of the intrinsic principles of the aperture of objective. Physics and information relative to interactive computing machinery is complemented by the principles of organization that *describe* the interactive information space. Language and interaction map onto their semantics by definition.

The operative instruction,

symbol target source,

is the human computer interface to the specification of the logical semantics of information and processing with hardware. The mechanical interface in the physical reference frame is the symbolic logic offered by hardware execution. The instruction stream executed by hardware to implement processes is organized as a sequence of instructions. These sequences are organized into the pipelines and state machines that deliver economic utility and productivity to the users of infomechanical tools.

The organization of an interaction machine employs concepts that facilitate the scale of organization necessary to interaction as an economic objective. The organization of digital space is implemented using data types. The organization of digital work is implemented using code types. These mechanical objects are (partial and complete) structural definitions. Associations of identity are assigned to conventions of device with *textual* construction and employment from software languages.

The organization of this machine maps the frame principle of organization to the mechanical implementation of organization.

$$\Pi \leftarrow \pi$$

The software language is a partial system of organization and representation that maps onto processes of interaction.

$$\Pi = \pi + \lambda$$

A system of data types is implicit or explicit in a software language to structure a digital information space.

$$\Pi = \pi + \lambda + \tau$$

A system of code types is implicit in a software system to structure a mechanical instruction stream.

$$\Pi = \pi + \lambda + \tau + \kappa$$

The organized machine maps onto the organization of the interaction machine in code and data.

$$\Pi = \pi + \lambda + \tau + \kappa + \delta + \sigma$$

This association of conception to realization terminates the development of objective from frame to hardware and software.

$$\Pi =_{\lambda\tau\kappa} \Sigma$$
$$\Delta =_{\rho\tau} \Pi$$
$$T =_{\rho\lambda} \Sigma$$

We may consider additional and alternative attributes of conception, but the relationship between type and code is an interesting observable. The frame presented relates type to code by reference and language, which is the illumination afforded by the concepts offered.

Type

$$\forall \Sigma \; \exists T \vdash \forall \sigma \; \exists \tau$$

In assembly language code types, the data type determines operator and subroutine selection. The continuity of type over code is sustained while operations available over types are not missing by fault of the conception and implementation of a type system.

In assembly form,

symbol target source,

the continuity of syntax and semantics of the source and target terms varies with type. The symbol may determine the type of terms, or in another language the type of terms selects a subroutine from symbol. This semantic continuity over data and code type is reflected in syntactic continuity over syntactic types. Where syntactic expression types (σ) relate both data (τ) and code (κ) types, as in the language with inline expression, construction, and call expressions, the same continuity over types is sustained where type representation and relation is sustained.

$$\tau\ \kappa \leftharpoondown_{\lambda} \sigma$$

Syntactic types define data and code types in a human computer interface language. Because type continuity is sustained or sustainable, type expression is endowed with the natural continuity of the mechanical interface.

Future work

Observation of logic in the mechanical frame, including software and interaction.

Series

[TMI/20190511/1] Complexity [pdf]
[TMI/20190428/1] Spacetime [pdf]
[TMI/20190406/1] Radiant objects [pdf]
[TMI/20190404/1] Atomism [pdf]

Theory of mechanical information
John Pritchard, @syntelos

https://docs.google.com/document/d/1WMY2s8_TOSexakv3WRbQprqxmwHmbXYUxu7KzMoOCq4/edit?usp=drivesdk