a reflection, but as a materialization of the divine, and develop this thought from theory into ecstatic and political-theological practice. The step from writing to action is no longer metaphorical, as it would be with a semantic text such as a political speech or a manifesto. It is concrete and physical because the very code is thought to materially contain its own activation; as permutations, recursions or viral infections. It is not only words made flesh, but words being flesh.

Computation as a figure of thought

Lullist imagination. Explicit and hidden theological politics seem to run through the various poetics and cultural practices of formally executable code, yet contradictory tendencies exist as well:

(1) Totalism vs. Fragmentation
- (a) Totalism / Synthesis
  The employment of algorithmics, permutation, combination and recursion to exhaust all existing aspects of a matter: The exhaustion of topics in rhetoric, the exhaustion of theological truth in Llull's *ars*, the exhaustion of knowledge in Lullist encyclopedism, the exhaustion of wisdom in Kuhlmann's poem through reverse-engineering Solomon.
  Algorithmics functions as expansion of a small source code into a near-infinitude of material.
  This tendency culminates in Quirinus Kuhlmann's theoretical sketch of an *Ars magna librum scribendi*, a universal letter combination machine designed to write all existing and potential books in the world (see p. 61).
- (b) Fragmentation / Analysis
  Generative classification of knowledge in turn compartmentalizes it into specialist domains. Program execution can be seen as the handling of complexity, the possiblity of reducing, for example, one hundred pages of proteic verse permutations to one line of material and one paragraph of instruction; or the reduction of several thousands statements created by Llull to three algorithms and one table of terms.
  In other words, combinatory knowledge classification embodies the dialectic that not only an abundance of information can be generated from a minimal source code, but that vice versa an abundance of information can be analytically reduced to one algorithm.

(2) Rationalization vs. Occultism
   (a) Rationalism
      The indexing and classification of knowledge, the development of artificial formal languages as "user interfaces" for data and algorithms, computation as such.
   (b) Occultism
      Theurgy, totalism expanding into demiurgic creation, code execution translating into practical political theology.
      Kuhlmann's example shows that theurgy and demiurgy are not necessarily precursors of a scientific rationalization, but that scientific rationalization (like that of 17th century encyclopedic Lullism) itself can be used as a philosophical ground for occult theology, precisely as an attempt to exceed scientific Lullism in a physical and metapysical, microcosmic and macrocosmic grasp of totality. This further proves that the cultural history of executable code cannot simply be written as a linear Hegelian progress from magical to scientific practices.
(3) Hardware vs. Software
   (a) Hardware
      There is a tendency of building computational hardware for algorithms: the device described in the *Sefer Yetzirah*, Llull's and Giordano Bruno's combination wheels, Harsdörffer's *Denckring*. Kuhlmann, too, designs a *Wechselrad* (*permutation wheel*) to speed up the permutation of his sonnet.
   (b) Software
      The rhetorical tradition of mental computation and memorization, in the "inventio" and "memoria."
      Abstraction from mechanical devices through symbolic handles and denominators, for example in the replacement of Llull's mechanical "figurae" through abstract symbolic denominators in later symbolic logic and computer programming languages.
(4) Syntax vs. Semantics
   (a) Syntax
      Program code as purely formal and therefore syntactical abstraction from human language.
   (b) Semantics

The inscription of metaphors and semantic handles, from Llull's letters B-K up to statements like "for," "while," "if" in programming languages.

(5) Artificial vs. Natural Language
   (a) Artificial Language
   The idea that language can be computed only through a purely formal, syntactical metalanguage that exists separately from natural language.
   (b) Natural Language
   The idea that common human language itself is a product of computation, and can be described with algorithms, such as in Harsdörffer's *Denckring*, Kuhlmann's poetry or artificial intelligence research.