

An introduction to Pasigraphy

What is Pasigraphy?

Pasigraphy is a pasigraphic programming language. A pasigraphic language is one that is universally and symbolically defined (see definition), rather than locally and explicitly, and so is understandable by all people.

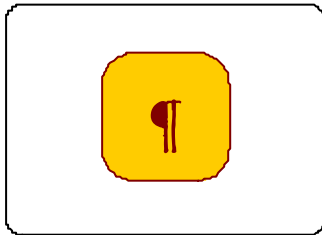
Pasigraphy is a programming language designed to be comprehensible by all people universally

pasigraphy

(pæsigræfi)

***n.* universal written language, especially using symbols for ideas rather than words. A system of universal writing, or a manner of writing that may be understood and used by all nations.**

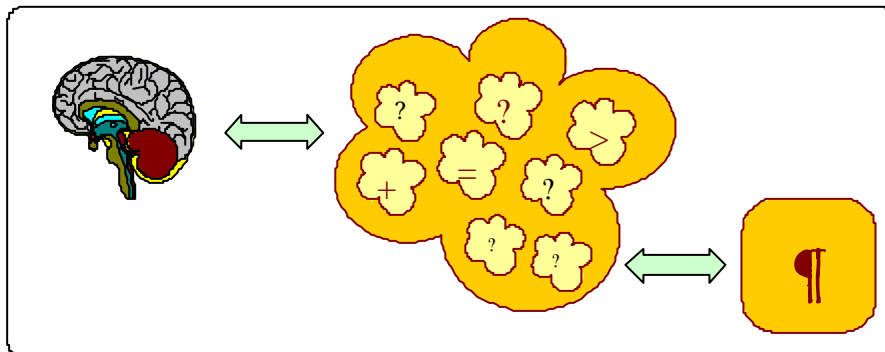
How is Pasigraphy pasigraphic?



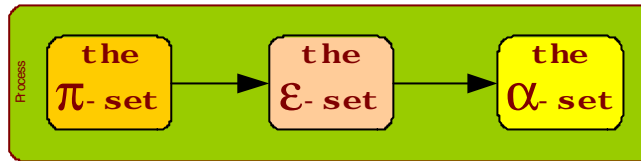
Pasigraphy is pasigraphic by being an *idealist* programming language: a central core of programming concepts is expressed as either universally comprehensible symbols or as locally comprehensible syllables. These are bound together with a grammar so they form naturalistic expressions to convey their meaning to programmer and machine.

So how is it different from other programming languages?

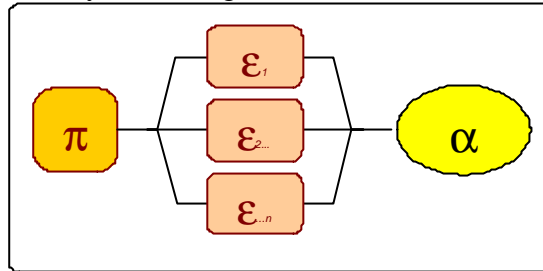
Pasigraphy is based on a systematic *a priori* analysis of the tasks required for computation. A hierarchical *symbolology* of these tasks is then created on the basis of this analysis, and it is with these symbols that the interpreter works. It is thus an *idealist* programming language.



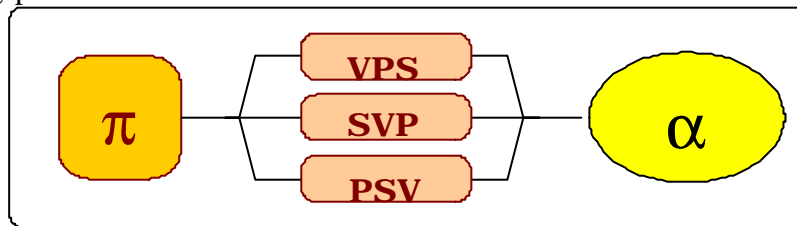
Pasigraphy works by having this core *symbol-set* of programming forms (the π -set), represented by a semasiographic *expression-set* (an ϵ -set) and interpreted by an interpreter as an *action-set* (the α -set).



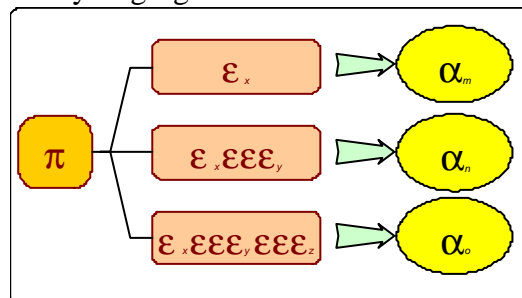
The π -set can be represented by an infinite number of ϵ -sets in any number of localized languages written, symbolic or gestural:



The core unit in the language is a *syllable* (an σ -token): there is a one-for-one equivalence of the concepts between the syllable set and the symbology, and it is these syllables that combine to make up the complex expressions that are needed for computation: these are the Pasigraphy *sentences*. These sentences are combined from syllables in a strictly determined order, and can take the form of verbs, nouns or denoting-phrases.

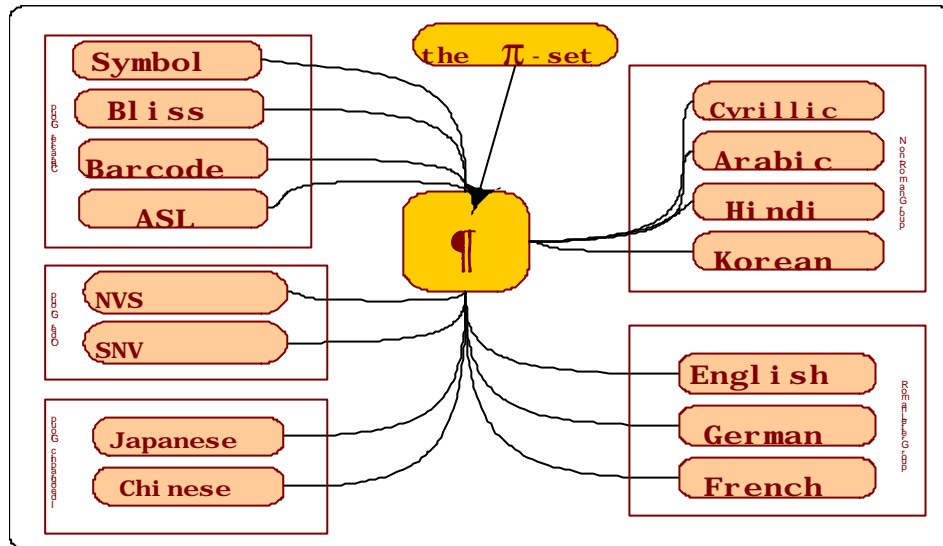


These syllables are agglutinative, and combine to make up compound expressions, which are then qualified by the parameters they delineate. The compound expressions are extensible by the subsequent addition of further syllables still. As the syllables combine to create complex expressions, so the addition of an extra qualifying syllable can alter the meaning of that sentence, but only in one direction – thus it behaves like ordinary language.



This makes possible modification to create modal expressions, temporal expressions, negation, conditional etc, and makes for the extraordinary power of expression found in other a priori programming languages like APL and LISP.

This goes against the tide of languages being chiefly defined in terms of Backus-Naur Form (Chomskyian level-3 context-free grammars). Rather, it takes what might be described as *Sammet-Panini Form*, following her work in language design in the 1960s, and his work on word construction in 3 millennia ago: expressions are intentional, they are drawn from the symbology, they are context sensitive and they form intelligible sentences.



There is an infinite number of possible representations of the core symbol set of Pasigraphy, each corresponding to a potential natural mode of expression: these are termed expression sets, and can be a natural language, an artificial language system, or even a non-linguistic expression that takes formal representation in non-textual form.

Currently there are several expression-sets, but there is a systematic research program afoot to develop the potential of Pasigraphy to embrace more language styles and groups. The desire is to enable people to program in what Pinker called “Mentalese”, and feel comfortable with the formation and grouping of concepts that make up programs and applications.

Those extant are the p-set, Protium (an English syllabic expression-sets), Peach (a Kanji expression-sets), Peony (a Chinese expression set), Pantagruel (a French expression set), Pontifex (a German expression set) and Panglossia (a purely graphical symbolic expression-sets). The Figure 1 above shows the direction of current research laid out in the various research groups: (Tasks updated January 2003)

- 1) Symbolic: here we are working on ASL and Semantographics in addition to the purely representation symbolic set, with a view to making non-verbal Mentalese expression possible for people who are fluent in these language forms. The use of ASL makes for potential gestured programming, which is a very exciting concept indeed. The project is making use of “Sign Language”, a Unicode Truetype font, to represent the gestures, and a data glove to capture movement.

Another interesting point of research is a bar-code expression-set, to permit automatic loading of code from a printed page.

- 2) Alternate word order: the contextualized binding order of the Pi-elements can logically be bound in any order – currently there are experimental NVS and SNV expression-sets being trialled.
- 3) Ideographic: in addition to Peach, we are converting Pasigraphy to Chinese as a matter of some priority, and the work should be completed by end 2001 (Now completed - Peony released March 2002)
- 4) Other Roman scripts languages: currently work is under way on French and German expression-sets, and these are expected to be released by February 2002. (Now completed)
- 5) Non-Roman character sets: longer-term research involves investigating non-Roman non-ideographic scripts, and there are plans for Arabic, Korean and Indian expression-sets to be written. Estimated completion time is end of 2nd Quarter 2002.

Of course, the number of potential expression-sets is limited, and so it would be possible for any individual or group to create a expression-sets to march there own language requirements. Indeed, it would be possible to create (for instance) an instantiation of Pasigraphy in the universal languages of Wilkinson or Delgano, or perhaps in Esperanto or Ido, or even ArtLangs such as Silmaril or Klingon, or sounded languages like Solresol. These are all equally expression-sets of Pasigraphy.

What does it look like?

Well, it doesn't look like anything, as it is idealistically defined. But as with all forms, the language can be reflected in any number of ways and then used to great advantage. We shall confine ourselves to Panglossia (the extended Hoare-algebra symbol expression-set), Protium (the English syllabic expression set) and Peach (the Japanese expression-set).

What are the data types in Pasigraphy?

There is a basic name space occupied by late-binding strings, and when the value is read, it is parsed as directed (by formula or attribute). Thus the same namespace can be treated as tree, set, string, triangular array etc.

Why should I care?

You should care if you are interested in a more naturalistic style of programming, if you are interested in simple but powerful, if you wish to work with colleagues in the context of their native cultural expression, if you want to take advantage of the power of aspect oriented programming, if you wish to take advantage of collections based programming: there are many more reasons besides.

What are the antecedents of Pasigraphy?

We're so glad you asked – there is a separate document on this very subject!

Why the paragraph symbol?

To get the flexibility we sought, we chose to embed Pasigraphy in a Macro-variant of XML. To this end, we needed a standard delimiter for the mark-up constructs. We chose the paragraph symbol not just because it was unique and expressible, but also because it was symbolic, and could be read by someone of any culture, and interpreted by their system to suit. This was to emphasise the symbolic, late-binding nature of the language.