

Jan van den Box, Marinus J. Plasmeijer, Pieter H. Hartel Input-Output Tools: A Language Facility for Interactive and Real-Time Systems IEEE Transactions on Software Engineering, May 1983

Interactive and real-time programs can usually be characterized as being input driven, that is, they are inactive until some input arrives, at which time they process the input, possibly generate some output and then return to the inactive state. Conventional programming languages do not provide adequate I/O facilities for these types of programs. This is the motivation given by Jan van den Bos et al for the development of the Input-Output Tools (IOT) language facility.

Some high level languages use a 'module' construct to package together a data structure with the procedures and functions that operate on that data structure. An IOT is a construct for packaging a set of procedures that act upon input data that is defined by an input rule. The power of the IOT construct lies in the concept of the input rule.

The input rule is an expression that specifies one or more input patterns that are to be matched. A set of operators is defined that allows the construction of regular expressions. Prefix and postfix functions can be specified with an expression to provide conditional matching for even more power. The operands of the expression are themselves IOTs. This implies a hierarchy of input-output tools. At the lowest level are the 'basic' IOTs which can be thought of as I/O primitives. The program itself is defined as the highest level IOT.

A number of examples are given in the article, showing the application of IOTs in several areas such as process control, computer graphics and terminal handling.

Although called input-output tools, the emphasis is definitely on the input -- which has traditionally been the more difficult of the two to generalize. The IOT model provides a structured way to deal with the parsing of stream input.

The article discusses 2 major implementations. Both of these are pre-processors for existing languages (C and Pascal/Modula-2).

The IOT language facility cannot be simply 'tacked' on to an existing programming language. The highly structured environment that it imposes on programs restricts the use of any programming language containing IOTs to a certain class of applications. This is consistent with the stated goals of the authors to design a language facility to support the special requirements of interactive and real-time programs.