COLOMBETTI84

Colombetti, Marco & Giovanni Guide, "Supporting Control Definition in Programming Languages", Computer Languages, Vol. 9, No. 1, 1984, pp. 1-23 - Control Structures. The recent development of programming languages has largely been influenced by software engineering methodologies, in particular structured programming, which is characterized by the use of abstraction reasoning in the design and construction of programs. Within sturctured programming methodologies, the concept of abstract data type has been thoroughly investigated, but the parallel concept of abstract control structures has not received much attention. The authors of this paper are of the view that the set of control structures available to the programmer strongly influences the programming style and discipline. They propose a meta-language, called DIL, to support user definition and implementation of control structures. The user applied DIL to a base language L to extend the control structures available, yielding an extended language L*. A control structure is defined by the authors as a family of partial flowcharts which share some common properties and represent a usual and repeating behaviour of control flow. The specification of a control structure requires a syntactic definition, which describes the form of the structure, and a semantic specification, which describes the function of the structure. The DIL definition of a control structure is hence given by the following BNF formula:

```
<definition> ::= DEFINE <identifier>
                    DECLARE <declaration>
                  PATTERN <pattern>
                    IMPLEMENTATION <implementation>
                 ENDDEF
```

The declaration, pattern and implementation consist of identifiers (of DIL and L variables, seperators, etc.), DIL constructs (iterator, enrichment, where-clause), and L statements (in the implementation only). The iterator allows denoting the repetition of pieces of text for a number of times unknown at the moment of definition, the enrichment allows a new construct to be used as an L* statement in a given scope, and the where clause restricts the nesting of control structures. The authors also describe the operation of the L* compiler, which parses the input L* text and generates the equivalent L program by expanding the text supplied in the implementation of the user-defined control structures. The development of DIL supports a new program construction technique in which the role of data types and control structures is considered in a balanced and unitary way. At each step of the refinement process, the appropriate data types are first chose and implemented, and the required processing algorithms invented; the adequate control structures are then introduced and programmed, which are needed for implementing the program in the most natural way. By using DIL as a programming tool, the user is allowed to design his own control mechanisms at a very high level, in addition to data types, thus defining the most appropriate programming environment required for a disciplined and effective programming activity.