

# Programming by steps

Raluca Oana Scarlatescu

The paper introduces a new method of software design and programming. The method is based on the old flow-charts, in order to obtain not only a logical representation of the problem to be solved, but also the real-time working of the software itself. What is a flow-chart? A sequence of steps, that could be classified like iteration, decision, jump, loop, etc. To “make” a step means to perform a certain function. Usually it is the job of the software engineer to put together these functions in a certain order and in this way a program is realised.

This paper presents a different idea to implement a flow-chart. On this is based the method of software design and programming in discussion. The steps of the flow-chart are a finite number of records in a database table, linked with their specific functions associated by name or number identification (id). To “make” a step supposes, like into a normally designed flow-chart, to perform the function and to pass to the next step, and so on, until the steps are finished.

These functions are macro-functions projected and stored in one or more separate programs. The macro-functions are characterised from some input and/or output parameters, that have to be transferred from one step to another, and realise a certain group of activities. The values of the parameters could be fixed or variable, dependent or independent of the execution of the other functions, a priori known or unknown. They could consist in different type of data (character, numerical, date, etc.). The macro-functions have a certain level of abstraction and standardisation, to be used in different situations.

A main software reads the information about each step, gathers the input values of the function associated with and interprets them, runs the function and scatters the output results in its parameters, manages the going over of all steps, and ensures the good execution of the entire system. There is a database management system, that stores the data about the steps and its precedence rules, functions and its parameters, dynamic values of the parameters, errors etc.

In view of all these facts, each program could be designed building the succession of steps with the necessary functions for the respective situations. In other words, the software engineer would fill some fields in the tables with the values necessary to complete the flow-chart and than to run the program. Or, in a future release, with an adapted graphic interface, a non-specialist user could be also the “writer” of the software.

The paper presents in details the principle of the Programming by steps, based on the scenario described above. It explains, also, which were the reasons that originally motivated the development of the method and defines the principal requisites to build an application system, the facility to develop other applications from the same family and the restrictions to be applied. The software package, that was designed and programmed with this method, is still working. It is oriented on the IVR applications, used to manage the phone calls with the computer.

There are enunciated the future aspects of the implementation and the advantages /disadvantages to design, implement and maintain the system.

The paper includes a comparative analysis of the Programming by steps and another two methods of software engineering: the Rapid Prototyping and the Component-based Design and Reuse. A demonstrative application is also enclosed. Integrative comments and conclusive remarks are provided in the conclusions of the paper.