

## AUTOMATICALLY GENERATING A PROGRAM

### FIELD OF THE INVENTION

The present invention concerns the field of automatic program generation. The program is generated for a component of a system consisting of several components that communicate with each other. The particular field of the present invention is the automatic generation of a state-based program for a component on the basis of a given, interaction-based specification of the system. The invention is particularly intended to be used in connection with complex, interacting programs, for example for telecommunication applications.

### BACKGROUND OF THE INVENTION

Program development generally takes place today in a multi-stage process. Initially, the requirements for the system to be developed are collected and first ideas regarding the solution are drawn up. Later these first solution ideas are refined step by step until finally an executable implementation in the form of a program is obtained.

In particular, interaction-based formalisms are known for specifying a system of communicating components in the early stages of the development. A specification in such a formalism comprises descriptions of possible execution sequences of the system, the communication (interaction) between the components of the system being emphasized. Examples for interaction-based sequence descriptions are message sequence charts (MSCs) according to recommendation Z.120 of the ITU (International Telecommunication Union) as well as sequence diagrams in UML (Unified Modeling Language) of the OMG (Object Management Group).

A very high effort in terms of human working time is required if an executable program shall be developed without automatic support from this kind of interaction-based specification of the system. In this respect, problems arise both because of the ensuing costs and because of the long development time, which can not be shortened by more than a certain degree even if the number of co-workers is increased. Moreover, the development process is very prone to errors, especially in connection with complex, interacting applications, which are, for example, common in the field of telecommunications. As a rule, a major part of the overall costs must be allotted for the searching and correcting of errors.

EP 0 343 682 B1 and its German translation DE 689 23 126 T2 each describe a system for automated programming. EP 0 049 176 A1 describes a program composing method based on test data inputs.

### OBJECTS AND SUMMARY OF THE INVENTION

Accordingly, one object of the present invention is to avoid the above-mentioned problems fully or at least in part. Another object is to shorten the process of program development. Another object is to decrease the necessary effort for program development. Another object is to avoid possible sources of error. Another object of the invention is a full or partial automation of the stages of program development from the interaction-based system specification to the executable program. Another object is to provide this automation in the context of communicating and possibly distributed systems.

According to the invention, these and other objects are attained by a method, an apparatus and a computer program product for automatically generating a state-based program for a component of a system consisting of a plurality of components communicating with each other, wherein said program is generated from a specification of said system, said specification comprising interaction-based sequence descriptions of said system, and wherein the following steps are performed:

- a) determining all sequence descriptions of said component defined by said specification of said system,
- b) normalizing said sequence descriptions of said component such that a normalized sequence description comprises exactly one initial condition and exactly one final condition and, between said initial condition and said final condition, communication actions only,
- c) determining a state-based specification of said component by identifying all equal initial and final conditions of said normalized sequence descriptions of said component with a single state, and
- d) determining said state-based program for said component, wherein each sequence description contained in said state-based specification of said component is replaced by a sequence of said communication actions of this sequence description, separated by additionally inserted states.

The dependent claims concern preferred embodiments of the invention.

No fixed order shall be defined by the enumeration of steps in the claims. Rather, the individual steps and sub-steps can be executed in the order recited in the claims, or in another order or parallel or quasi parallel or interleaved with each other or interleaved with steps of other concurrent calculations. The use of the indefinite article in the expression “. . . automatically generating a state-based program . . .” is intended to comprise the situation that in a single execution of the steps of the invention several state-based programs are generated together, each such program for one component of the system.

The invention is based on the idea to automatically generate a state-based program from the specification of the system by means of a sequence of appropriate transformation steps. Thus the invention facilitates the development process considerably, since the costly, manual transformation is no longer necessary. The programming effort is shifted towards producing the system specification. In other words, the invention extends the functionality of known compilers since the automatic program generation no longer requires a source code in a programming language, but a mere system specification is sufficient.

The starting point of the invention is a specification of a system consisting of several components communicating with each other. In particular, the components may be data processing devices that are locally separated from each other, as this is often the case, for example, in the field of telecommunication applications. The components, however, may also be separated from one another from a merely logical point of view, for example, in order to structure the system or to shield the components from each other. In this case the components may be concurrent processes in a data processing system or even individual modules of a larger program. Intermediate forms between the two extremes of a strictly local and a strictly logical separation are contemplated as well.

The communication between the components is possible in all manners, for example via unidirectional or bidirectional communication channels or by means of synchroniz-