Bull. Nov. Comp. Center, Comp.Science, 14 (2001), 1–6

Construction of models of computing structures with fine-grain parallelism in WinALT system*

D.T. Beletkov, I.V. Zhileev

Two-tier procedure of constructing compound model of computing structure with fine-grain parallelism are discussed in this paper. The description of tools of construction is presented. It was shown that the visual programming technology was utilized for their creation, and the care was taken to give to the user intuitively understandable and easy commands and to provide the reliability of textual and graphical parts design.

Introduction

The algorithm of a problem solution with fine-grain parallelism in a general case is a network of asynchronously communicating operators: more simple algorithms, each of which performs parallel in time and universal transformation of data arrays, placed in discrete (cellular) space.

Actually, formal representation of such algorithm is a description of computing structure that implements it with fine-grain parallelism. That is why the composition of models for such structures from the constantly extending set of modules for the verification and obtaining the complexity estimations seems to be vital in the WinALT [1].

The two-tier scheme of models composition is proposed in the paper: the level of design of a single module and the level of asynchronous composition of modules.

The composition of complex programs from the modules written in high level languages is well developed. And besides, the WinALT language also supports the modular construction of models with the help of functions and procedures. These language capabilities are fully utilized for the model construction. These auxiliary means of decomposition differ from the traditional ones. They describe interoperating fine-grain parallelism processes instead of communicating sequential ones which are widely accepted for the parallel programs. This feature of the composition requires the creation of composition tools which are involved at the both stages of model design and model simulation and debugging. At the stage of the design, the main at-

*Supported by the Russian Foundation for Basic Research under Grant 99-07-90422.

tention was drawn to implement the intuitively clear user's commands and the reliability of graphical and textual parts design.

1. The formal foundation of composition

The asynchronous composition of Parallel Substitution Algorithm (PSA) [2] serves the formal basement for the main composition. The PSA is a model for distributed computations. A parallel substitution, represented by a set of named cells is the main means for information transformation. W denotes the set of cells, M is a set of names being words from a finite alphabet A. A parallel substitution is formed by an expression $S_1 * S_2 \to S_3$, where S_i stands for configurations, S_i is the association that links each cell $m^0 \in M$ with a set of cells $S_i(m^0)$.

By using m^0 as an argument, the substitution generates a microcommand. The application of this microcommand to W has two steps: 1) search of the association of sets $S_1(m^0)$ and $S_2(m^0)$ in the set W; 2) if such association is found, in replacement of the set $S_1(m^0)$ on the set $S_3(m^0)$.

The Parallel Substitutions Algorihm Φ (PSA Φ) will be formed by a finite set of parallel substitutions. The PSA Φ carries out an iterative procedure of transformation of set of cells. At each step simultaneously and everywhere all microcommands, generated by substitutions from the PSA Φ, are applied.

In many real world applications (cellular automata, cellular-neural networks, associative structures etc.), a graphic image of cellular set is represented by a rectangle. This rectangle is made up from coloured cells. The left and right parts of microcommands also have graphic images.

Asynchronous composition at construction complex PSA Φ from certain PSA Φ_1, ..., PSA Φ_n consists in construction of control PSA Φ_c which provides switching on/off of algorithms PSA Φ_1, ..., PSA Φ_n by startup/finish commands. The construction of control PSA Φ_c at the first step is carried with the help of a Petri-net. At the second step, this net by formal rules is translated in to PSA Φ_c. The advantage of asynchronous composition is that it provides joint parallelization of calculations and control and preservation of parallelism of the initial problem.

2. The structure of a model

The computer representation of a model with fine-grain parallelism is determined by use of a composition, the presence of graphic images not only for data, but also for commands. A model is represented by a project [3]. The project consists of files of several types. One of them is the description of the project (.wap). This file contains structural representation of model. The project includes graphic objects containers (.3do) and source texts (.src) as

well. The .3do files contain visual images of data and commands. Source text files contain the descriptions of mechanisms of command application to a data. Data and command objects are located in .axt files. All files, included in the project, are stored in a separate directory on a disk. When opening a project all its files are opened, their contents becomes accessible for viewing and editing in separate windows. It is possible to add files (.3do, .axt, .src) created by the user, imported from other projects or libraries into the project.

3. The description of construction technology of model in WinALT system

The procedure of construction of compound model contains:

1) Model construction of modules in a project, import of modules from libraries and other projects;
2) Petri net construction for the control module and its inclusion into the the project, extension of module models by the operators that simulate of start/stop commands.

The usage of tools that support the two-tier scheme of composition by a user is depicted in Figure 1. The hatched arrows denote the actions performed by a user, the solid ones denote the actions of the system. There are four groups of tools.

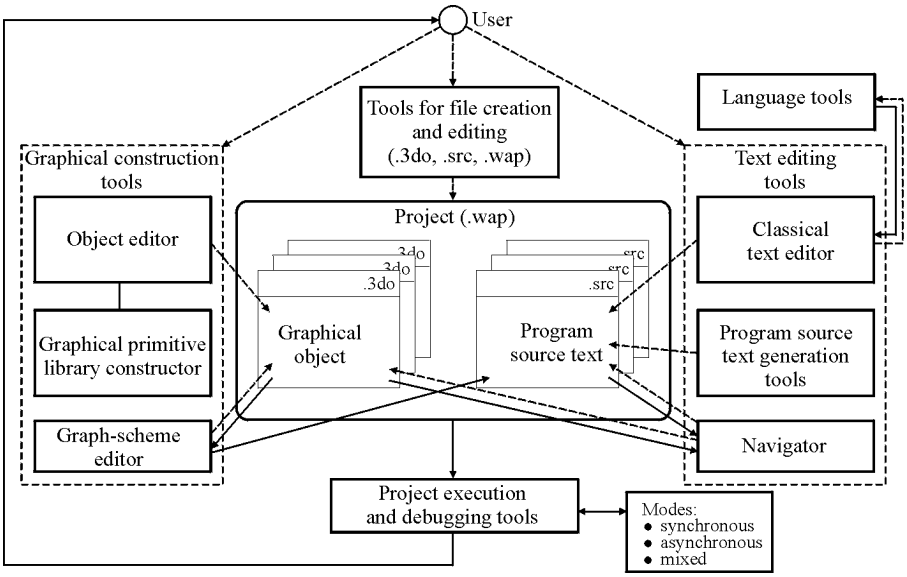

Figure 1. Model construction tools

Tools of project files creation and deletion. These tools aid a user to resolve a strategic task of project construction from a set of custom files, standard libraries and files from other projects.

Program text editing and visualization tools. These tool can be divided into two groups: a standard tools of text processing and the fine-grain parallelism specific ones.

The text editor implements all the typical operations of text processing. Their look and feel is the same as in a commonplace Windows application. The syntactical text highlighting is among the specific features of the editor. The lexems of the same type appear in the same color. The syntactical highlighting facilitates the program debugging and a comprehension of the program structure. The generation of the skeleton source text is implemented as well.

The specific fine-grain parallelism tools for model construction are represented by program block constructor, expression constructor, object explorer (Figure 2).

The first one is intended for the generation of the substitution operators and synchroblocks. Besides the constructor is utilized for generation of sequential loops and condition operators to support a user with an alternative way for substitution description. When creating a block a user selects a

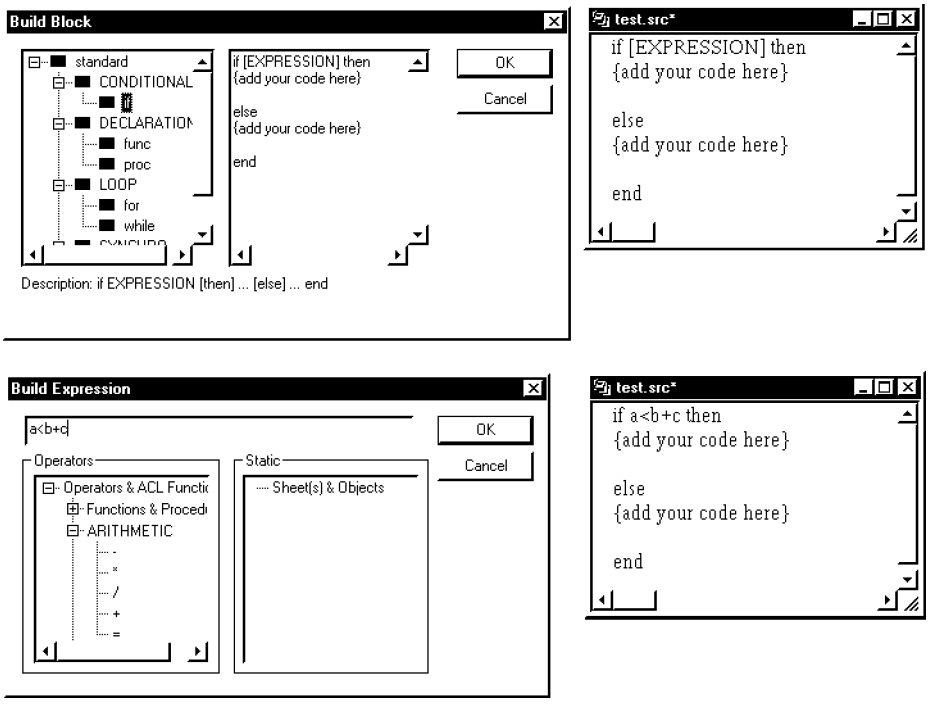

Figure 2. Block and expression constructors

particular operator from a list in a constructor dialog window. After a user fills out all the parameters the source text is generated.

The expression constructor automates the creation of complex functional transformations performed by cells of array and the description of substitution scope. It is intended for the creation of arithmetical and logical expressions. The appearance of expression constructor is similar to that of the block constructor.

Program part of a model manages objects by the names. The object navigator considerably improves the reliable transition of object names from the graphical part of the model into the textual one and the translation of a name into its respective object.

The tools of visualization and editing introduced above allow a user to create well structured descriptions of complex models with concurrent and sequential control by the means of WinALT procedures and functions.

The tools of graphical construction. The graphical editor is a part of these tools [3]. We shall only note that it is involved into each step of the construction of model's graphical part. The constructor of graphical primitive library is a new part of these tools. It facilitates the creation of graphical images of data and the reliability of their design because of its utilization for the construction of graphical primitive libraries. These primitives allow to set values of regions in cellular arrays instead of single cells. A library of boolean functions primitives serves a good sample of this type of libraries. Each function has a number of input and output links. The design of universal cellular structures graphical images is based on this library.

All the tools described above are used for the model construction. These models can serve as modules themselves at the second level of the construction. The principal tool at this level is the graph-scheme editor (Figure 3). A user can paint the control Petri net that has marked transitions related to

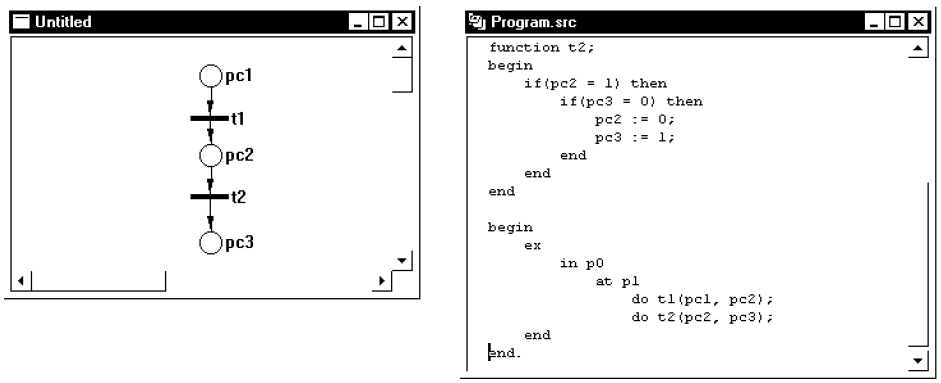

Figure 3. The graph-scheme editor

modules of which the model is formed of. The graph-scheme editor generates WinaLT source program and cellular arrays by a Petri net. At this moment, a user is capable of watching the transitions of the markers in the graphical image of the net. A program can be simulated in either asynchronous or mixed mode which were implemented in the system as the addition to the synchronous mode. In the asynchronous mode, only one applicable microcommand is executed at a single step of the simulation. In the mixed mode, more than one command can be applied at a time. Let us note as well that these modes of synchronization can be used for a wide set of other applications.

After a user have corrected the modules of a model with the help of graphical editor and model text visualizer so as to introduce cells and operators that set launch and termination commands, the project of the compound model is ready.

The tools of project simulation and debugging. These tools were introduced in [4]. In the context of these article, we mention them only to demonstrate the interaction of system with user in case of error appearance in model or finding the better model variant.

Conclusion

Further development of construction technology consists of increase of automation level by introducing:

1) tools supporting cellular object modification: their placement, sizes, orientation, both manually by a user and automatically from a model program in the process of simulation;
2) tools supporting automatic extension of models of modules by addition of operators and issuing the start/stop commands.

References

[1] Beletkov D., Ostapkevich M., Piskunov S., Zhileev I. WinALT, a software tool for fine-grain algorithms and structures synthesis and simulation // LNCS. – Springer, 1999. – № 1662. – P. 491–496.

[2] Achasova S.M., Bandman O.L., Markova V.P., Piskunov S.V. Parallel Substitution Algorithm. Theory and Application. – Singapore: World Scientific, 1994.

[3] Beletkov D.T. Graphical construction of computer models for 3D algorithms and model construction // Proceedings of Young Scientists Conf. – Novosibirsk: ICM&MG, 1998. – P. 3–13.

[4] Beletkov D.T. The tools of project debugging in WinALT simulation system // NCC Bulletin, Special Ser. – Novosibirsk: NCC Publ., 1999. – Issue 1. – P. 1–8.